Boosting Collective Entity Linking via Type-guided Semantic Embedding

Weiming Lu, Yangfan Zhou, Haijiao Lu, Pengkun Ma, Zhenyu Zhang and Baogang Wei

College of Computer Science and Technology, Zhejiang University

Abstract. Entity Linking (EL) is the task of mapping mentions in natural-language text to their corresponding entities in a knowledge base (KB). Type modeling for mention and entity could be beneficial for entity linking. In this paper, we propose a type-guided semantic embedding approach to boost collective entity linking. We use Bidirectional Long Short-Term Memory (BiLSTM) and dynamic convolutional neural network (DCNN) to model the mention and the entity respectively. Then, we build a graph with the semantic relatedness of mentions and entities for the collective entity linking. Finally, we evaluate our approach by comparing the state-of-the-art entity linking approaches over a wide range of very different data sets, such as TAC-KBP from 2009 to 2013, AIDA, DBPediaSpotlight, N3-Reuters-128, and N3-RSS-500. Besides, we also evaluate our approach with a Chinese Corpora. The experiments reveal that the modeling for entity type can be very beneficial to the entity linking.

1 Introduction

Entity Linking (EL) is the task of mapping mentions in natural-language text to their corresponding entities in a knowledge base (KB). One of the major challenges for EL is that mentions are often ambiguous, which can only be resolved with an appropriate context. For example, *Washington* in sentence *"In 1775, the Second Continental Congress commissioned **Washington** as commander-in-chief of the Continental Army in the American Revolution"* refers to *George Washington*, while *Washington* in sentence *"As of 2010, there were an estimated 81,734 immigrants living in **Washington**."* refers to *Washington,D.C.*

Several approaches have been proposed to improve the performance of EL, such as collective entity linking approaches [7, 11, 10, 12, 5, 13, 30], entity relatedness learning [21, 2, 13], and neural network approaches [9, 23, 27].

Obviously, the essential step of EL is to define a similarity measure between mention and entity. However, previous approaches usually used handcrafted features to measure the similarity, such as surface features, context features and special features [29]. Afterwards, generative models such as entity-mention model [6] and selective context model [15] were proposed for similarity measurement. Recently, deep learning approaches are becoming increasingly popular for the EL task. For example, a neural network was proposed to model context, mention and entity for entity disambiguation [23]. Words and entities are mapped into the same continuous vector space for named entity disambiguation by extending the skip-gram model [27].

However, there are still some places to boost entity linking. First, the context of mention should be modeled as a sequence to capture the semantics, while the context were both modeled as a bag of words in [23, 27]. Second, the types of mention and entity would be the hints for entity linking. In the above examples, the mention following the verb *commission* may be a person, while the mention following *living in* may be a location. In addition, we found that the words in categories or tags indicate both the semantic and type of entity. For example, entity *George Washington* has categories such as *American surveyors*, *commanders in chief*, and *Presidents of the United States*. While, entity *Washington,D.C.* has categories such as *Capital districts and territories*, *Washington metropolitan area*, *Planned cities in the United States*. Obviously, the categories are exactly consistent with the semantic and type of mentions in context.

Based on the above observations, we propose a type-guided semantic embedding approach to boost collective entity linking. We use bidirectional Long Short-Term Memory (BiLSTM) to model the context, and use dynamic convolutional neural network (DCNN) [14] to model the categories. Then, we build a graph with the semantic relatedness of mentions and entities based on the semantic embedding for collective entity linking. We evaluate our approach by comparing the state-of-the-art entity linking approaches over a wide range of very different data sets, such as TAC-KBP from 2009 to 2013, AIDA, DBPediaSpotlight, N3-Reuters-128, and N3-RSS-500. Besides, we also evaluate our approach with a Chinese Corpora.

The rest of the paper is structured as follows: We describe the type-guided semantic embedding for entity linking in Section 2, and present our experimental results in Section 3. Section 4 reviews the related works and Section 5 gives the conclusion.

2 Model

In this section, we first describe our type-guided semantic embedding, and then we boost the collective entity linking through the embedding.

2.1 Type-guided Semantic Embedding

The typed-guided semantic embedding for mention and entity is illustrated as Figure 1.

As mentioned above, we use BiLSTM and DCNN to learn the representations of mention and entity respectively. Their resulting vector representations x_m and x_e can be used to compute a mention-entity similarity score through a similarity matrix M. Then, the join layer concatenates x_m, x_e and the similarity score into a single vector, which is then passed through three fully connected hidden layers. Finally, the output of the hidden layers is further fed to the softmax classification layer, which will generate a initial mention-entity linking probability. The details will be elaborated in the following sections.

BiLSTM based Mention Embedding In order to capture the type and semantic of a mention, we apply BiLSTM to model the context $c_m = [..., x_{m-2}, x_{m-1}, x_m, x_{m+1}, x_{m+2}, ...]$ of the mention x_m, as shown in Figure 2.

Since the word itself and its POS type are both important for mention embedding, so the vector of each context word x_m is made up of two parts: a word embedding and a type embedding. The word embedding is represented by a W_x-dimensional vector, and initialized by word2vec with the skip-gram model, while the type embedding is

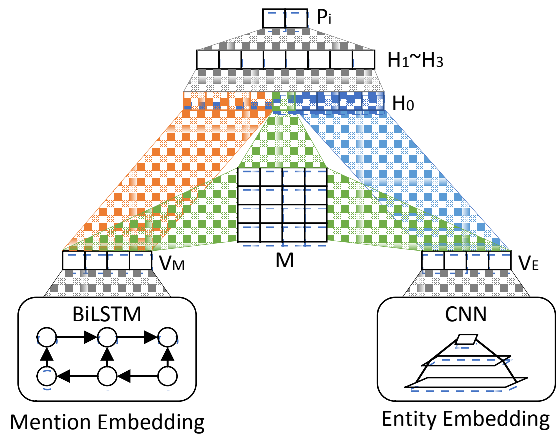

Fig. 1. Typed-guided Semantic Embedding

represented by a W_t-dimensional vector, and initialized by a one-hot vector. Finally, the outputs of the forward and backward of BiLSTM are concatenated as the mention embedding v_m for x_m. The window size of the context is S.

DCNN based Entity Embedding Since the words in categories or tags indicate both the semantic and type of entity, and they are orderless, so we use DCNN [14], which is a convolution neural network with dynamic k-max pooling, to model the category of entity.

The DCNN network has three convolutional layers with four feature maps each, and same padding is used in each layer. There is a k-max pooling layer after each convolutional layer, and a folding layer is applied between the last convolutional layer and the k-max pooling layer. Finally, the output of the last k-max pooling layer is flattened as a vector v_e to be the entity embedding. The detailed DCNN network is shown in Figure 3, where each solid circle represents one feature map of the CNN network, and the input of the DCNN network is the category of each entity, which contains a bag of words $x_1, x_2, ..., x_L$.

In the figure, each word in the category is represented by a W_x-dimensional word embedding from word2vec, so the input of the DCNN network is a $L \times W_x$ vector. The pooling parameter k of three k-max pooling layers is 2, 3 and $max(L/6, 1)$. The details of the DCNN can be referred to [14].

Embedding Training To train the embedding, we concatenate the mention embedding v_m, the entity embedding v_{e_i} and their similarity score $s(v_m, v_{e_i}) = v_m^T \cdot M \cdot v_{e_i}$ as a vector $H_0 = [v_m^T, s(v_m, v_{e_i}), v_{e_i}^T]$, and then pass H_0 through three hidden fully connected layers as shown in Figure 1. That is, $H_1 = \sigma(W_0 \cdot H_0 + b_0)$, $H_2 = \sigma(W_1 \cdot H_1 + b_1)$, and $H_3 = \sigma(W_2 \cdot H_2 + b_2)$. Finally, H_3 is further passed to a softmax layer to obtain a initial mention-entity linking probability p_i between mention m and entity e_i.

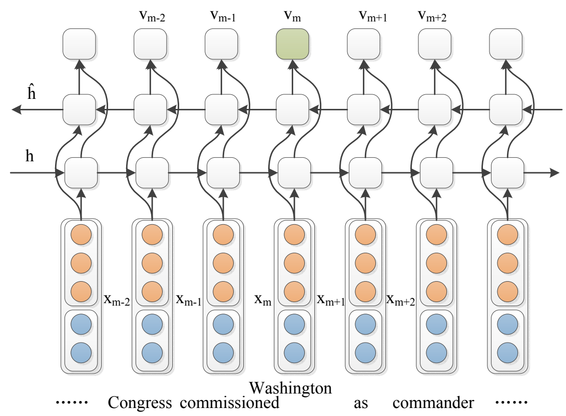

Fig. 2. Mention embedding by BiLSTM

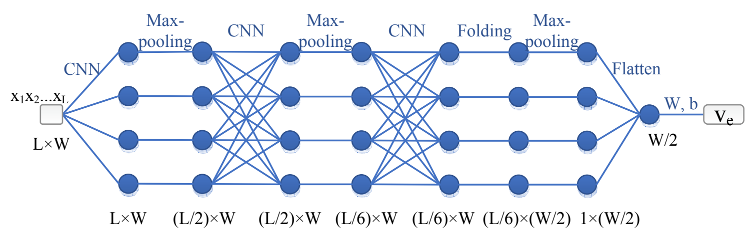

Fig. 3. Category embedding by DCNN

Given a training set $\mathcal{T} = \{t_i = \{(m_i, c_i, E_i, e_{i_j})|e_{i_j} \in E_i\}, i = 1, 2, ..., |\mathcal{T}|\}$, for each training sample t_i, m_i is the mention, c_i is the context of m_i, E_i is the entity candidate set for m_i, and e_{i_j} is the target entity. So the vector $y_i = [y_{i_1}, y_{i_2}, ..., y_{i_{|E_i|}}]$ can be represented the linking information for the entity candidate set for mention m_i, where $y_{i_j} = 1$ and $y_{i_k} = 0$ for $k \neq j$.

Finally, the embeddings are learned by minimizing the cross-entropy cost function: $J(\theta) = -\sum_{t_i \in \mathcal{T}} \sum_{j=1}^{|E_i|} y_{i_j} \log p_{i_j} + \lambda||\theta||$, where θ are all parameters needed in the type-guided semantic embedding network. Here, we use RMSprop to optimize the parameters θ by using TensorFlow.

2.2 Collective Entity Linking

Since all mentions in the same sentence or paragraph are encouraged to resolve to entities that are related to each other, so we integrate the type-guided semantic embedding into the collective entity linking framework for the better linking performance.

We first construct a mention-entity graph $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges. $\mathcal{V}$ includes three types of nodes: mentions $M = \{m_1, m_2, ..., m_N\}$, their candidate entities $E = \{E_1, E_2, ..., E_N\}$ and some unambiguous entities $E' = \{e'_1, e'_2, ..., e'_{|E'|}\}$. Taking Figure 4 as an example, there are two mentions m_1 and m_2, and each of them has several candidate entities. For instance, m_1 has its candidate entities e_{1_1} and e_{1_2}, and m_2 has its candidate entities e_{2_1} and e_{2_2}. In addition, there are two unambiguous entities e'_1 and e'_2.

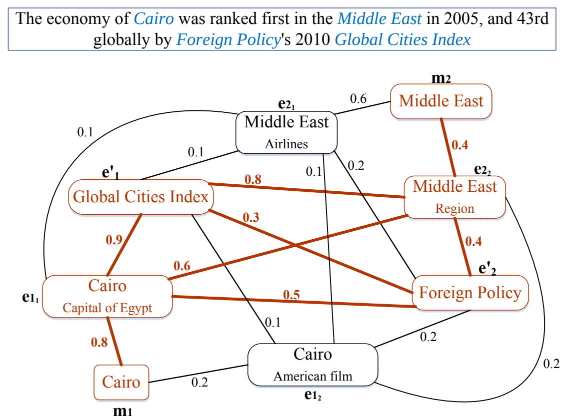

Fig. 4. Collective Entity Linking

There are two types of weighted edges in $\mathcal{G}$: entity-entity edge and mention-entity edge. Let W be the weight matrix of the graph $\mathcal{G}$. Then the weight of edge between mention m_i and its candidate entity e_{i_j} is computed as $W_{ij} = p_{i_j}$, which is calculated through the type-guided semantic embedding network. In order to compute the weight of an entity-entity edge, we learn the *Paragraph Vector* [16] for each entity from its content of the corresponding wikipedia article, and then compute the weight of edge between two entities e_i and e_j as $W_{ij} = \cos(v_{e_i}, v_{e_j})$, where v_{e_i} is the paragraph vector for entity e_i. The weight matrix W could be normalized for each row.

After the construction of the mention-entity graph, we utilize Random Walk with Restart [24] to calculate the score of each candidate entity $e_{i_j} \in E_i$ for mention m_i. Then, mention m_i could be linked to the entity with the highest score.

Formally, we initialize a $|\mathcal{V}|$-dimensional vector s by $s[i] = 1/|M|$ when $\mathcal{V}_i$ is a mention node. Then the formula $r = (1-\lambda) \cdot W \cdot r + \lambda \cdot s$ is computed iteratively until convergence, where $r \in R^{|\mathcal{V}| \times 1}$ and can be initialized as s. Here, $r[i_j]$ can be considered as the score of its corresponding entity e_{i_j}. Finally, the entity linked to mention m_i should be $e(m_i) = \arg\max_{e_{i_j} \in E_i} r[i_j]$.

3 Experiments

3.1 Experimental Settings

Since our approach is language independent, we conduct the experiments on both Chinese corpora and English corpora.

For Chinese corpora, we collected articles in Baidu Baike[1], which is a large-scale collaborative Chinese encyclopedias like Wikipedia, and built a Chinese knowledge base (CKB) like DBPedia. Each entity in the CKB has its properties, categories, description and so on. Similarly, we used Wikipedia as the knowledge base for entity linking on English corpora.

For the English corpora, we evaluated our approach over a wide range of different data sets, such as TAC-KBP, AIDA/CoNLL-Complete, DBpediaSpotlight, N3-Reuters-128, and N3-RSS-500.

In order to train the type-guided semantic embedding network, a large training set should be provided, so we created the training set automatically based on the inner-links in each article. Specifically, we assume that the inner-links in Wikipedia and Baidu Baike are all correct at first, therefore an sample (m_i, c_i, E_i, e_{i_j}) can be added to the training set if and only if (i) the anchor text of the inner-link is m_i, and it links to an entity e_{i_j}. (ii) more than two entities can be found according to the surface form or the synonyms of m_i, then these entities can be served as the entity candidate set E_i. The testing set for Chinese corpora is also generated similarly. Both the training set and testing set are generated randomly. Finally, we generated 743,978 samples for training and 55,716 samples for testing on Chinese corpora.

In the evaluation, we computed both micro (aggregates over all mentions) and macro (aggregates over all documents) precision scores for the entity linking.

3.2 Results on Chinese Corpora

When evaluating our approach on Chinese corpora, we set $W_x = 400$, $W_t = 148$, $S = 40$ in the semantic embedding and $\lambda = 0.5$ in the collective entity linking framework through cross validation.

In addition to baseline *PriorProb* which links to surface forms to the entities with the highest prior probability, we also compared our approach with some state-of-the-art entity disambiguation approaches such as DSRM[13] and LIEL[22]. In *PriorProb+CEL*, we assigned the weight of the edge between entity and mention with the prior probability during collective entity linking. The experimental results are shown in Table 1.

Table 1. Results on Chinese corpora

Method	Micro Prec.	Marco Prec.
PriorProb	0.6983	0.6844
LIEL	0.7063	0.7189
DSRM	0.7434	0.7296
PriorProb+CEL	0.7191	0.7123
Our Approach	**0.8107**	**0.8211**

From the table, we can see that our approach can obtain the best performance. Besides, in order to prove the effectiveness of the type-guided semantic embedding, we replaced the prior popularity $p(e_i|m)$ with our learned similarity score $s(v_m, v_{e_i})$ in

[1] http://baike.baidu.com/

DSRM, and improved DSRM by 3.42 and 5.25 percentage points on micro and macro precisions.

3.3 Results on English Corpora

In order to compare our approach with other entity disambiguation frameworks on publicly available data sets on English corpora, we used GERBIL (General Entity Annotation Benchmark Framework)[2] to evaluate the approaches on the D2KB task, whose goal is to map a set of given entities mentions to entities from a given knowledge base or to NIL. GERBIL is an evaluation framework for semantic entity annotation, and has provided several annotators (i.e. Babelfy[18] and DBPedia Spotlight[17]) and datasets (i.e. DBpediaSpotlight, N3-Reuters-128, N3-RSS-500).

The first step of EL is to generate the possible entity candidates for each mention m. We resorted to the Wikipedia search engine to retrieve all related entities as the candidates. However, the target entities are NIL in some cases in the TAC-KBP datasets, but we can still find the target entity for the mention according to the search results. For example, in a document fragment from TAC-KBP 2009 dataset: ”*two years later when the Canton Bulldogs beat the Chicago Cardinals*”. The mention ”*Canton Bulldogs*” is linked to NIL according to the dataset, but the Wikipedia entry for ”*Canton Bulldogs*” (American football team) is available at present which is created in 2015. The reason is that some Wikipedia pages are created after the dataset construction, so our approach would return wrong entity according to the ground truth. Thus, we removed those cases from the TAC-KBP datasets, and only retained English cases. Finally, we formed two types of datasets for each TAC-KBP dataset from 2009 to 2013:

1. TAC-KBP(sub): the target entity is among the candidates, whose size is at least 2.
2. TAC-KBP(full): TAC-KBP(sub) + the cases with its’ target may be NIL.

We carried out the experiments with the TAC-KBP (2009-2013) datasets on the GERBIL framework, and the results are shown in Table 2.

From the table, we find that our approach significantly outperforms the other approaches on the TAC-KBP datasets except TAC-KBP 2012. The main reason for the degradation in TAC-KBP 2012 is: many mentions in TAC-KBP 2012 refer to locations and places, such as *Bristol*, *Porto* and *Lyon*. However, the corresponding candidate entities for these mentions all have the same type (eg. *Location* and *Place*), which makes our approach unsuitable a little.

We also carried out the experiments on the GERBIL framework with its provided annotators and datasets, including AIDA[11], Babelfy[18], FREME NER[3], Kea[26], WAT[19], DBPedia Spotlight[17], Dexter[1], euNER[4], xLisa [28], and NERD-ML [25]. The results are shown in Table 3.

Overall, our approach reaches the best averaged F1 of all approaches. In detail, our approach significantly outperforms all other approaches on the N3-Reuters-128 and N3-RSS-500 data sets, but performs comparatively poor on the AIDA/CoNLL-Complete and DBpediaSpotlight data sets. There are two reasons for the performance degradation. Firstly, As in TAC-KBP 2012 dataset, many mentions in AIDA/CoNLL-Complete and DBpediaSpotlight also refer to the candidate entities with the same type (eg. Location,

[2] http://aksw.org/Projects/GERBIL.html
[3] https://github.com/freme-project/freme-ner

Table 2. Results on TAC-KBP datasets

Annotator	micro F1	macro F1	micro F1	macro F1
	TAC-KBP 2009(sub)		TAC-KBP 2009(full)	
WAT	0.717	0.681	0.5849	0.5134
DBPedia Spotlight	0.6063	0.5878	0.5449	0.4906
AIDA	0.5644	0.4552	0.531	0.4318
Babelfy	0.5606	0.5556	0.4919	0.4078
Kea	0.4782	0.3728	0.3416	0.236
FREME_NER	0.445	0.3262	0.3148	0.3148
Our Approach	**0.8467**	**0.8315**	**0.7061**	**0.6985**
	TAC-KBP 2010(sub)		TAC-KBP 2010(full)	
WAT	0.745	0.7333	0.5624	0.449
DBPedia Spotlight	0.588	0.5765	0.5007	0.4034
AIDA	0.5501	0.5059	0.4166	0.4166
Babelfy	0.6267	0.6157	0.5332	0.4382
Kea	0.5726	0.549	0.4868	0.3866
FREME_NER	0.4835	0.3451	0.3476	0.2437
Our Approach	**0.8634**	**0.8549**	**0.7948**	**0.7743**
	TAC-KBP 2011(sub)		TAC-KBP 2011(full)	
WAT	0.6998	0.6777	0.4635	0.4
DBPedia Spotlight	0.6542	0.6379	0.4633	0.3727
AIDA	0.4335	0.3953	0.3406	0.2739
Babelfy	0.5896	0.5847	0.4228	0.325
Kea	0.4179	0.3887	0.3179	0.2693
FREME_NER	0.358	0.2492	0.317	0.317
Our Approach	**0.7483**	**0.7309**	**0.6632**	**0.6568**
	TAC-KBP 2012(sub)		TAC-KBP 2012(full)	
WAT	0.3382	0.3257	0.2955	0.2469
DBPedia Spotlight	**0.513**	**0.5038**	**0.5239**	**0.5051**
AIDA	0.2692	0.2545	0.245	0.1773
Babelfy	0.4849	0.4707	0.3193	0.2884
Kea	0.2721	0.2697	0.2803	0.2738
FREME_NER	0.1845	0.1272	0.1181	0.0797
Our Approach	0.3732	0.369	0.3043	0.284
	TAC-KBP 2013(sub)		TAC-KBP 2013(full)	
WAT	0.6598	0.6531	0.5784	0.461
DBPedia Spotlight	0.6875	0.6735	0.5794	0.5032
AIDA	0.6108	0.5204	0.5286	0.4269
Babelfy	0.7158	0.6939	0.6249	0.5341
Kea	0.6108	0.5204	0.5286	0.4269
FREME_NER	0.2923	0.1939	0.2739	0.1769
Our Approach	**0.7755**	**0.7755**	**0.7126**	**0.6964**

Table 3. The comparison of macro-averaged F1 for different approaches through GERBIL

Annotator	AIDA/CoNLL-Complete	DBpediaSpotlight	N3-Reuters-128	N3-RSS-500	Average
WAT	**0.6708**	0.6778	0.4286	0.364	0.5353
DBPedia Spotlight	0.4897	0.6863	0.265	0.161	0.4005
AIDA	0.4942	0.1648	0.317	0.374	0.3375
Babelfy	0.5993	0.5115	0.3877	0.381	0.4699
Kea	0.5834	0.7247	0.4502	0.389	0.5368
FREME_NER	0.5901	**0.8202**	0.4709	0.379	0.5651
Dexter	0.4704	0.2506	0.3037	0.293	0.3294
euNER	0.4735	0.1938	0.3394	0.32	0.3317
xLisa	0.3616	0.5724	0.2879	0.368	0.3975
NERD-ML	0.1164	0.5282	0.3418	0.3013	0.3219
Our Approach	0.61	0.6131	**0.5538**	**0.621**	**0.5995**

Place and Person). Secondly, the datasets contain some structured data such as tables and lists in Web pages, which can reduce the performance of BiLSTM for mention embedding, since they are not in sequence, and then further reduce the performance of our approach.

4 Related Work

Entity linking is beneficial to annotate text by linking mentions appearing in text with their corresponding entities in the knowledge bases, so it has been widely studied in the last decade, and there is a comprehensive survey [20] for entity linking recently.

Traditional approaches [3, 8] addressed the entity linking problem by comparing the similarity between context information of a mention and the corresponding candidate entities in KB. Nowadays, several approaches have been proposed to improve the performance of EL, such as collective entity linking approaches [7, 11, 10, 12, 5, 13, 30], entity relatedness learning [21, 2, 13], and neural network approaches [9, 23, 27].

Collective entity linking approaches assume that entities occurred in the same document would have a high global coherence. [7] proposed a graph-based collective entity linking method, which can jointly infer the referent entities of all mentions in document by exploiting both the global interdependence between different EL decisions and the local mention-to-entity compatibility. [10] proposed a stacking based collective entity linking method, which stacks a global predictor on top of a local predictor to collect coherence information from neighboring decisions. [12] proposed a semi-supervised graph regularization model for entity linking in tweets by incorporating both local and global evidences from multiple tweets. [5] proposed a coherence model with an attention mechanism, where the score for each candidate only depends on a small subset of mentions, since an entity may only have relations to a small subset of other entities.

In addition, entity relatedness learning, which learns the semantic similarity between entities for coherence modeling, can also boost collective entity linking approaches. For example, [21] measured the semantic similarity between Wikipedia concepts based on the taxonomy of the knowledge base. [2] discovered suitable entity relatedness functions that can better support the entity linking task. [13] presented a semantic relatedness model (DSRM) based on deep neural networks (DNN) and semantic knowledge graphs (KGs) to measure entity semantic relatedness. [30] also measured the relatedness between entities based on semantic embeddings that capture entity and document contexts.

More recently, neural networks are widely used to address the EL task. For example, [9] proposed a deep learning approach with stacked denoising auto-encoders and supervised fine-tuning to learn context-entity similarity measure for entity disambiguation. [23] encoded mention, context and entity with a tensor neural networks for entity linking. [27] proposed a joint learning method to map words and entities into the same continuous vector space for entity linking.

5 Conclusion

In this paper, we propose a type-guided semantic embedding approach to boost collective entity linking. We used the Bidirectional Long Short-Term Memory (BiLSTM) to model the context of a mention, and dynamic convolutional neural network (DCNN) to model the type of an entity. Then, we built a graph with the semantic relatedness of mentions and entities for the collective entity linking. Finally, we evaluated our approach by comparing the state-of-the-art entity linking approaches over a wide range of very different data sets, such as TAC-KBP from 2009 to 2013, AIDA, DBPediaSpotlight, N3-Reuters-128, and N3-RSS-500. Besides, we also evaluated our approach with a Chinese Corpora. The experiments reveal that the modeling for entity type can be very beneficial to the entity linking.

Acknowledgements

This work is supported by the Zhejiang Provincial Natural Science Foundation of China (No. LY17F020015), the Chinese Knowledge Center of Engineering Science and Technology (CKCEST), and the Fundamental Research Funds for the Central Universities (No. 2017FZA5016).

References

1. Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R., Trani, S.: Dexter: An open source framework for entity linking. In: Proceedings of the Sixth International Workshop on Exploiting Semantic Annotations in Information Retrieval. pp. 17–20. ESAIR '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2513204.2513212
2. Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R., Trani, S.: Learning relatedness measures for entity linking. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 139–148. ACM (2013)
3. Cucerzan, S.: Large-scale named entity disambiguation based on wikipedia data. In: EMNLP-CoNLL (2007)
4. Dojchinovski, M., Kliegr, T.: Entityclassifier.eu: Real-time classification of entities in text with wikipedia. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III. pp. 654–658 (2013)
5. Globerson, A., Lazic, N., Chakrabarti, S., Subramanya, A., Ringaard, M., Pereira, F.: Collective entity resolution with multi-focal attention. In: ACL (2016)
6. Han, X., Sun, L.: A generative entity-mention model for linking entities with knowledge base. In: ACL (2011)

7. Han, X., Sun, L., Zhao, J.: Collective entity linking in web text: A graph-based method. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 765–774. SIGIR '11, ACM, New York, NY, USA (2011), `http://doi.acm.org/10.1145/2009916.2010019`
8. Han, X., Zhao, J.: Nlpr_kbp in tac 2009 kbp track: A two-stage method to entity linking. In: TAC (2009)
9. He, Z., Liu, S., Li, M., Zhou, M., Zhang, L., Wang, H.: Learning entity representation for entity disambiguation. In: ACL (2013)
10. He, Z., Liu, S., Song, Y., Li, M., Zhou, M., Wang, H.: Efficient collective entity linking with stacking. In: EMNLP. pp. 426–435 (2013)
11. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: 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)
12. Huang, H., Cao, Y., Huang, X., Ji, H., Lin, C.Y.: Collective tweet wikification based on semi-supervised graph regularization. In: ACL (2014)
13. Huang, H., Heck, L., Ji, H.: Leveraging deep neural networks and knowledge graphs for entity disambiguation. CoRR abs/1504.07678 (2015)
14. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014)
15. Lazic, N., Subramanya, A., Ringgaard, M., Pereira, F.: Plato: A selective context model for entity resolution. TACL 3, 503–515 (2015)
16. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: ICML. vol. 14, pp. 1188–1196 (2014)
17. Mendes, P.N., Jakob, M., Garcła-silva, A., Bizer, C.: Dbpedia spotlight: Shedding light on the web of documents. In: In Proceedings of the 7th International Conference on Semantic Systems (I-Semantics (2011)
18. Moro, A., Raganato, A., Navigli, R.: Entity linking meets word sense disambiguation: a unified approach. Transactions of the Association for Computational Linguistics 2, 231–244 (2014)
19. Piccinno, F., Ferragina, P.: From tagme to wat: a new entity annotator. In: Proceedings of the first international workshop on Entity recognition & disambiguation. pp. 55–62. ACM (2014)
20. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: Issues, techniques, and solutions. IEEE Transactions on Knowledge and Data Engineering 27, 443–460 (2015)
21. Shen, W., Wang, J., Luo, P., Wang, M.: Linden: linking named entities with knowledge base via semantic knowledge. In: WWW (2012)
22. Sil, A., Florian, R.: One for all: Towards language independent named entity linking. In: ACL (2016)
23. Sun, Y., Lin, L., Tang, D., Yang, N., Ji, Z., Wang, X.: Modeling mention, context and entity with neural networks for entity disambiguation. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI). pp. 1333–1339 (2015)
24. Tong, H., Faloutsos, C., Pan, J.Y.: Random walk with restart: fast solutions and applications. Knowl. Inf. Syst. 14, 327–346 (2008)
25. Van Erp, M., Rizzo, G., Troncy, R.: Learning with the web: Spotting named entities on the intersection of nerd and machine learning. In: # MSM. pp. 27–30 (2013)
26. Waitelonis, J., Sack, H.: Named entity linking in# tweets with kea (2016)
27. Yamada, I., Shindo, H., Takeda, H., Takefuji, Y.: Joint learning of the embedding of words and entities for named entity disambiguation. In: CoNLL (2016)
28. Zhang, L., Rettinger, A.: X-lisa: Cross-lingual semantic annotation. Proc. VLDB Endow. 7(13), 1693–1696 (Aug 2014), `http://dx.doi.org/10.14778/2733004.2733063`

29. Zheng, Z., Li, F., Huang, M., Zhu, X.: Learning to link entities with knowledge base. In: HLT-NAACL (2010)
30. Zwicklbauer, S., Seifert, C., Granitzer, M.: Robust and collective entity disambiguation through semantic embeddings. In: SIGIR (2016)