Hindawi
Complexity
Volume 2021, Article ID 8895742, 9 pages
https://doi.org/10.1155/2021/8895742

Research Article

Entity Linking Based on Sentence Representation

Bingjing Jia,[1,2] **Zhongli Wu**,[2] **Pengpeng Zhou**,[1] **and Bin Wu**[1]

[1]*Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Anhui Science and Technology University, Bengbu 233000, China*

Correspondence should be addressed to Bin Wu; wubin@bupt.edu.cn

Received 12 September 2020; Revised 26 October 2020; Accepted 8 January 2021; Published 19 January 2021

Academic Editor: Wei Wang

Entity linking involves mapping ambiguous mentions in documents to the correct entities in a given knowledge base. Most existing methods failed to link when a mention appears multiple times in a document, since the conflict of its contexts in different locations may lead to difficult linking. Sentence representation, which has been studied based on deep learning approaches recently, can be used to resolve the above issue. In this paper, an effective entity linking model is proposed to capture the semantic meaning of the sentences and reduce the noise introduced by different contexts of the same mention in a document. This model first uses the symmetry of the Siamese network to learn the sentence similarity. Then, the attention mechanism is added to improve the interaction between input sentences. To show the effectiveness of our sentence representation model combined with attention mechanism, named ELSR, extensive experiments are conducted on two public datasets. Results illustrate that our model outperforms the baselines and achieves the superior performance.

1. Introduction

With the development of big data, a large number of unstructured texts have appeared on the Internet, and almost all of the mentions in the texts are ambiguous. For example, Figure 1 gives a snippet "Spain and U.S. teams are very competitive. This year's Fed Cup finalists—defending champion Spain and the United States—will hit the road to open the 1997 women's international team competition," in which the mention "Spain" may refer to Spain, the Spain national football team, the Spanish Empire, or the Spain Fed Cup team. When the context of the mention is different, it may refer to different entities. There are thousands of entities in a knowledge base (KB), which are the basis of many research studies [1]. Therefore, KB is regarded as surrogates for real-world entities. The main purpose of entity linking focuses on linking mentions in text to corresponding entities in a KB such as Freebase. Entity linking is beneficial to fully understanding these texts [2–4], and it also can boost the development of question answering, machine reading, and knowledge base population.

The task of entity linking is challenging because of inherent ambiguity of natural language. Previous studies commonly rank entities based on a measure of similarity between the mention and the entity to determine the best candidate. Various types of features have been designed, including entity popularity, entity type, and entity co-occurrence. For entity popularity, Milne and Witten [5] achieved 90% accuracy on the Wikipedia test articles using the prior probability. For entity type, Nie et al. [6] used type information to improve the co-attention effect. Chen et al. and Gupta et al. [7, 8] captured latent entity type information through pretrained entity embedding. For entity co-occurrence, graph-based methods are explored to improve the impact of coherence. Guo and Barbosa [9] obtained the indirect connections of entities through random walks on the disambiguation graph. There is some useful information in these statistics and features, which makes the entity linking problem easier. In most cases, these statistics and features are provided by KB. The separate entity records only have description information when the incomplete KB is sparse and simple [10]. Therefore, the above methods may not perform well. It will be expensive and time consuming to inject structural features and statistics into entities manually. So, only the entity description, which is the most common information in KB, is considered in this paper. In addition,

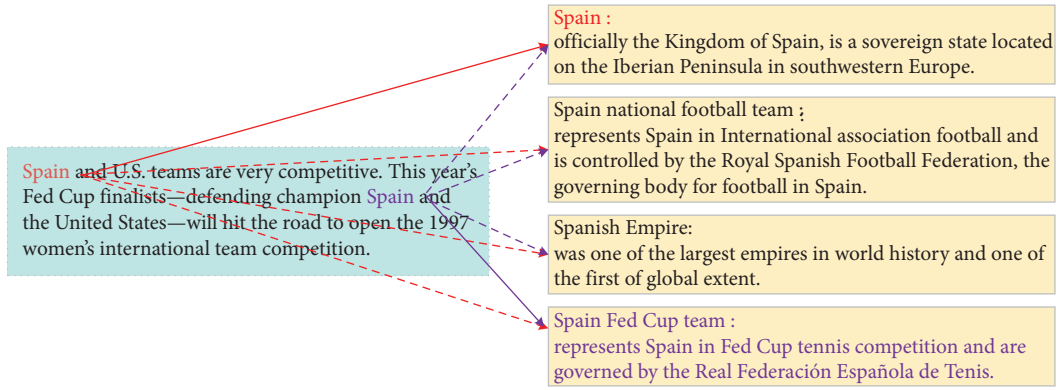

Figure 1: Illustration of mentions in the snippet and their candidate entities in the knowledge base. Solid lines point to the correct target entities corresponding to the mentions and to the descriptions of these correct target entities.

when a mention appears multiple times in a document, the conflict of its contexts in different locations may lead to difficult linking. As shown in Figure 1, the first and second occurrences of mention "Spain" may link to an entity of "Spain" and an entity of "Spain Fed Cup team," respectively. It is necessary to take information from a sentence where a mention appears.

Therefore, ELSR is proposed at a sentence level. This model uses the symmetry of Siamese network to derive the dependencies between the mention context and entity description, which also alleviates the issue caused by the incomplete knowledge base. The main contributions of this paper are listed as follows:

(i) A novel symmetrical neural model is proposed for entity linking, which not only captures the semantic meaning of the sentences but also reduces the noise introduced by different contexts of the same mention in a document.

(ii) The key information extracted only from the mention context and entity description are suitable for the incomplete knowledge base, which can make up the sparseness and the simplicity. Besides, the attention mechanism can improve the interactive effects between two sentences.

(iii) To validate the performance of the model, experiments are conducted over two corpora. The results illustrate that our model is powerful in most cases.

The structure of the rest of this paper is organized as follows. Section 2 presents related work. The entity linking model is detailed in Section 3, and experimental results are described in Section 4. Section 5 gives conclusions and suggestions for future work.

2. Related Work

Entity linking is a well-studied task in KB population. It is also a key step for the question answering, content analysis, and information retrieval. The early works applied extensive hand-designed features and require intensive manual efforts, including prior popularity, name string similarity, Wikipedia's category information, and so on. A representative work attempted to develop sophisticated features to rank the entities [11]. The results showed that the context features and the special features perform well. Then, the hidden information generated by the entity-topic model was used to enhance the context feature [12]. Besides, link information in KB was leveraged to construct the graph, which was combined with four confidence scores [13]. Lexical and statistical features were added into the unified semantic representation for documents and entities to solve the all-against-all matching problem [9].

To reduce time and alleviate manual work, recent neural network-based models have been investigated to capture semantic features of mentions and entities, which achieved the state-of-the-art performance. In [14], a denoising autoencoder was first applied to encode documents and entities. Francislandau et al. [15] combined sparse feature with multiple granularity information learned by convolutional neural networks (CNNs). Sun et al. [16] utilized memory network to automatically find key information for a mention from surrounding contexts to facilitate entity linking. Ganea and Hofmann [17] combined entity embeddings with the contextual attention for local linking. Then, the loopy belief propagation was used for global inference and achieved competitive results. RLEL [18] regarded entity linking as a sequence decision problem and gave the result based on a reinforcement learning model. Unfortunately, their model only used the previously referred entities and failed to find the consistency of subsequent entities. Because of the flexibility and efficiency of the graph neural networks, they can capture better graph representations and pay attention to find the relatedness between entities. Cao et al. [19] utilized graph convolutional network to encode entity graphs. However, this model only focused on target entities, which generated lots of noisy data. Wu et al. [20] added some local and global features into an entity dependency graph and utilized the graph convolutional networks to capture the structural information among entities. Fang et al. [21] generated high recall candidate sets and introduced a sequential graph attention network to obtain the topical coherence of mentions. However, the construction of graphs is very time consuming, and the complexity of these models is relatively high.

Following the burgeoning popularity of embedding methods on knowledge graphs, some studies tried to apply embedding algorithms on entity linking. Yamada et al. [22] easily measured the similarities between mentions and entities by constructing a novel embedding, in which words and entities were placed into the same vector space. When jointly learning representations of texts and entities, NTEE [23] could easily predict entities for the text in the KB. However, the above two studies neglected the relatedness among word senses. FCSE [24] proposed multiprototype word embedding model by extending senses of a word based on the global and local information. Results showed that FCSE outperformed other embedding models. SA-ESF [25] combined the prior probability and the similarities between mentions and entities to rank the entities. Besides, it utilized structural features, context features, and entity ID feature to represent entity embeddings. The average F1 score was up to 86.6%, which showed that the symmetrical Bi-LSTM neural network is effective. The emergence of bidirectional encoder representation from transformers (BERT) has attracted the attention of researchers [26]. PEL-BERT created a protocol knowledge base and fine-tuned BERT on the protocol corpus to link entities [27]. This model achieved the highest accuracy of 72.9% and provided a guideline for domain-specific entity linking. The BERT-based entity similarity score was integrated into the local context model to find latent entity type information, which improved the F1 score by 1.32% on AIDA-CoNLL test set when compared with baselines [28].

However, most models incorporate multiple features, which are extracted from KB. When facing closed domains, KB is too simple to get statistics and structural features. Few models perform well on the incomplete knowledge base. Besides, when a mention appears multiple times in a document, the conflict of its contexts in different locations may lead to difficult linking. It is necessary to link entity at a sentence level. Therefore, sentence similarity can be used to predict the relationships between mentions and entities. Most existing approaches are based on the Siamese network, which has two branches [29]. Each branch shares the same set of weights and the same architecture. The advantage of the Siamese network is that it is easy to train. Despite great success in these methods, the interaction between the two sentences is often ignored and has not been fully utilized. In this paper, a general entity linking model is proposed. This model is symmetrical, which only considers the mention context and entity description. Meanwhile, the Siamese network is used to represent sentences and the attention mechanism can improve the interaction between input sentences.

3. The Proposed Model

The Siamese network can encode the two sentences into the embedding vectors in the same space and make the model smaller and easier to train by sharing parameters. Meanwhile, adding attention mechanisms for Siamese network not only capture vital information but improve the interaction between input sentences. According to the above advantages, our proposed model consists of four main components, including embedding, fine-tuning BERT, interaction, and prediction. The overall architecture of the model is illustrated in Figure 2. The embedding layer can convert tokens into word embeddings. The fine-tuned BERT layer generates sentence embeddings. The interaction layer attempts to improve sentence representation with consideration of the interactive effects from the other sentence. The final layer gives the label prediction by a two-layer multilayer perceptron. As shown in Figure 2, the only input of our model is mention context and entity description. The mention context is the sentence where the mention appears, and the entity description is the first sentence selected from its corresponding Freebase article. Firstly, the mention text and entity description are converted into word embeddings through the embedding mechanism proposed in BERT. After fine-tuning, the sentence representations will be improved. Secondly, attention mechanism is used to obtain the rich interactions both for the mention context and the entity description. Finally, two vectors are compared through elementwise operations and aggregated into a fixed-length vector. And a two-layer multilayer perceptron is used to estimate similarity score between the mention and the entity. The model is symmetric. In the following sections, we describe each of these modules in detail.

3.1. Embedding. BERT facilitates pretraining deep bidirectional representations on unlabeled text by fusing the left and the right context in all layers, including two steps: pretraining and fine-tuning [26]. During pretraining, the input sequence will be tokenized when they are fed directly into ELSR. Special token [CLS] indicates the start of every input sequence, and [SEP] is inserted after each sentence. Then, each example of sentence pairs can be represented as a triple (C, D, l), where $C = (c_1, c_2, \ldots, c_N)$ is a sentence containing the mention, $D = (d_1, d_2, \ldots, d_M)$ is another sentence containing the corresponding entity to be link, and $y \in Y$ is the label indicating whether the entity is correct. Here, $Y = \{0, 1\}$, where $y = 1$ means the correct entity for a mention and $y = 0$ otherwise. N and M represent the length of the pair sentences. Therefore, as shown in Figure 3, the input representation is constructed by joining the corresponding token, segment, and position embeddings.

3.2. Fine-Tuning BERT. The word embeddings for mention contexts and entity descriptions are denoted as $E^c = (E_1^c, E_2^c, \ldots E_{N-1}^c, E_N^c,)$ and $E\,d = (E_1^d, E_2^d, \ldots E_{M-1}^d, E_N^d,)$ respectively, where E_i^c or E_j^d is a k-dimensional vector. When feeding them into BERT, the important features are learned parallelly through several transformer blocks. Then, the output of the i-th word is converted to H_i^c over the mention contexts embeddings E^c. The generation method of H_j^d is similar to H_i^c.

$$H_i^c = \mathrm{BERT}\left(E_i^c\right), \quad \forall i \in [1, 2, \ldots, N-1, N], \tag{1}$$

$$H_i^d = \mathrm{BERT}\left(E_j^d\right), \quad \forall i \in [1, 2, \ldots, M-1, M], \tag{2}$$

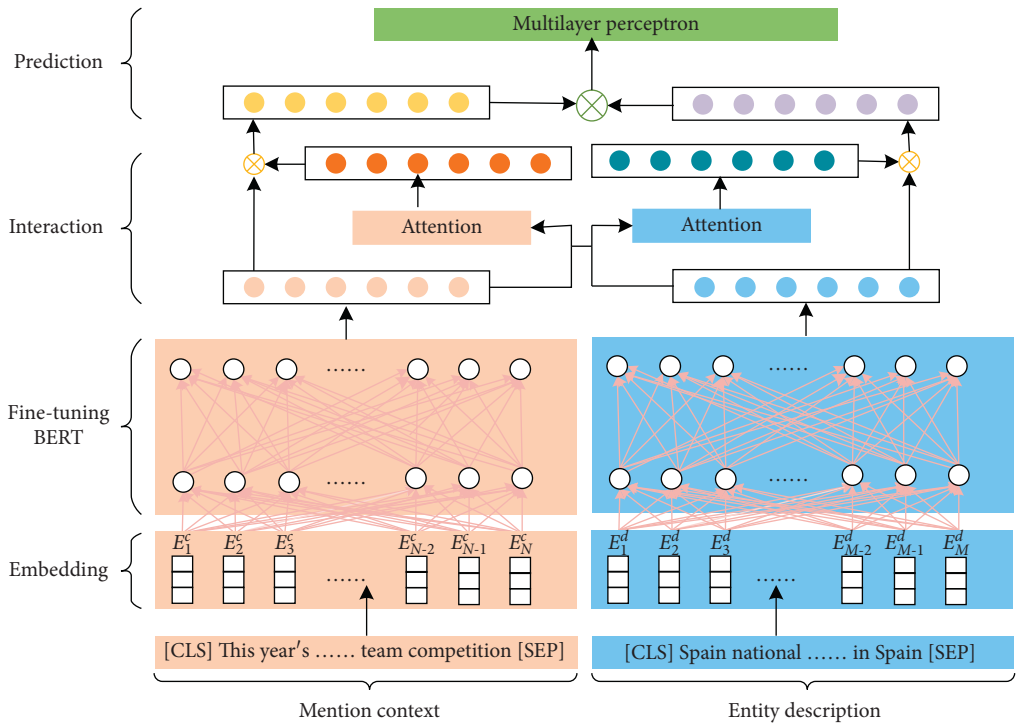

Figure 2: Architecture for entity linking based on sentence representation, where and denote multiple operations between two vectors.

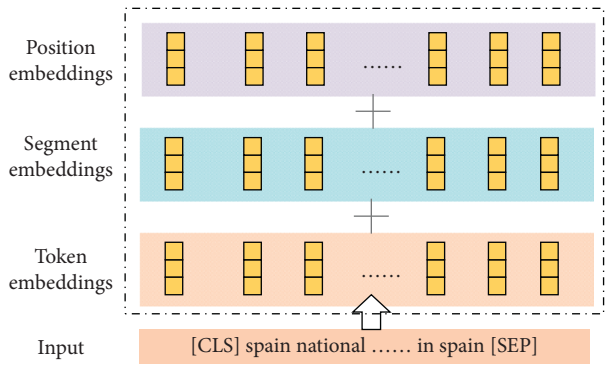

Figure 3: The input representation.

Different semantic information is contained in different layers of a neural network. To adapt BERT to the entity linking task, the most effective layer should be set. Intuitively, different learning rates can be used to fine-tune the layer of the BERT to generate more general information. In addition, since longer sequences are disproportionately expensive, it is necessary to select an appropriate length to balance the effect and the speed.

3.3. Interaction. This layer allows for rich interactions between the mention contexts and entity descriptions, which only contain a part of useful information for EL. Therefore, attention mechanism is used to select the important words and reduce the noise. Concretely, there are two steps for enhancing the mutual influence between two sentences. First, the soft attention alignment [30] is leveraged to connect the two sentences based on the relevant information. Let S_{ij} denote the attention weight, $\widehat{H}_i^c$ represent a weighted summation of $\{H_j^d\}_{j=1}^M$, and $\widehat{H}_j^d$ stand for a weighted summation of $\{H_i^c\}_{i=1}^N$, respectively, as shown in equations (3)–(5).

$$S_{ij} = \left(H_i^c\right) \cdot H_j^d, \tag{3}$$

$$\widetilde{H}_i^c = \sum_{j=1}^{M} \frac{\exp\left(S_{ij}\right)}{\sum_{k=1}^{M} \exp\left(S_{ik}\right)} H_j^d, \quad \forall i \in [1, \ldots, N], \tag{4}$$

$$\widetilde{H}_j^d = \sum_{i=1}^{M} \frac{\exp\left(S_{ij}\right)}{\sum_{k=1}^{N} \exp\left(S_{kj}\right)} H_i^c, \quad \forall j \in [1, \ldots, M]. \tag{5}$$

The above formulas mean that the content in $\{H_j^d\}_{j=1}^M$, which is closely related to H_i^c, is selected to represent $\widetilde{H}_i^c$.

Similarly, we choose the content related to H_j^d from $\{H_i^c\}_{i=1}^N$ to create $\tilde{H}_j^d$. There are multiple matching operations between two sentence vectors to measure the "similarity" or "closeness." The operations mainly include the elementwise difference and product of the tuple $\langle H^c, \tilde{H}^c \rangle$ and $\langle H^d, \tilde{H}^d \rangle$, which can be used to capture the interactive semantic information. Finally, those fine-grained features are concatenated to enhance the sentence representation, as given in equations (6) and (7).

$$O^c = \left[H^c - \tilde{H}^c; H^c \odot \tilde{H}^c; H^c; \tilde{H}^c\right], \tag{6}$$

$$O^d = \left[H^d - \tilde{H}^d; H^d \odot \tilde{H}^d; H^d; \tilde{H}^d\right]. \tag{7}$$

3.4. Prediction. The information contained in the context representation O_c and the description representation O_d should be combined based on a heuristic matching trick, which was proposed by Mou et al. [31]. represents the matching operations, including elementwise difference, elementwise product, and concatenation. Concretely, elementwise difference and product can measure the "similarity" and "closeness" between two sentences. Concatenation represents the standard characteristics of the "Siamese" network. The complete matching procedure is given in the following equation:

$$k = \left[O^c; O^d; O^c - O^d; O^c \circ O^d\right], \tag{8}$$

where "O" represents elementwise product and k denotes the matching features stacked together. Finally, k will be put into a two-layer multilayer perceptron to predict the probabilities of each label. The probability indicates the similarity between mention contexts and entity descriptions. The closest one is selected as the final result.

During training, the network aims to minimize the cross-entropy loss illustrated in equation (9). The idea behind this is that the output probability of a correct entity would be larger than the probability of the corrupted entity by a margin of 1 [32].

$$L = \sum_{(m,e\in T)} \max\left(0, 1 - \psi\left(m, e^+\right) + \psi\left(m, e^-\right)\right), \tag{9}$$

where T denotes the dataset of sentence pair, m is the sentence which represents the mention to be linked, e^+ is the gold standard entity, e^- is a corrupted entity, and $\psi(m, e)$ represents the probability, which indicates the comprehensive correlation between m and e.

4. Experiment

This section details the implementation of ELSR, including experiment, experimental results, and model analysis. First, general setting of ELSR is introduced in Section 4.1. Then, Section 4.2 not only explores the properties of ELSR but also compares it with the state-of-the-art methods. Finally, the model analysis is conducted in Section 4.3. Our model is implemented in the Tensorflow framework on the hardware with Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10 GHz and GeForce GTX 1080 Ti.

4.1. Experiment

4.1.1. Dataset and Metric. ELSR is evaluated on two standard datasets: AIDA and KBP. Moreover, KB is important for the entity linking task. Most existing methods are based on Wikipedia, which is the most popular encyclopedia in the world. However, data in Wikipedia are not structured. Freebase can solve this problem. Freebase is a large collaborative knowledge base, which has more than 43 million entities and 2.4 billion related facts about entities. Therefore, Freebase is taken as our reference knowledge base. Concretely, AIDA dataset [33] has been manually annotated and split into AIDA-Train, AIDA-A, and AIDA-B. We train our model on AIDA-Train, valid it on AIDA-A, and report performance on AIDA-B. KBP 2016 and 2017 is from Text Analysis Conference-Knowledge Base Population (TAC-KBP) [34], in which each document only contains one mention and extracted from Newswire (NW) or Discussion Forum (DF). We train and valid our model on the 2016 English dataset and test the performance on 2017 English dataset. Owing to previous work [35], all the mentions have been annotated in the documents. Table 1 gives the descriptions of the two datasets. Here, we extract more than ten thousand entities and their descriptions for the task of entity linking.

According to the experiment protocols proposed in [6], only non-NIL mentions which have target entities in KB are considered. Furthermore, $precision = recall = F_1 = accuracy$. Therefore, accuracy is selected as our evaluation metric, which can be calculated as the number of correctly linked mentions divided by the total number of all mentions. As shown in equation (10), accuracy also indicates whether a top-ranked entity is the ground truth or not:

$$\text{accuracy} = \frac{|\{\text{correctly linked mentions}\}|}{|\text{total number of all mentions}|}. \tag{10}$$

4.1.2. Parameter Settings. Following the existing work [35], the mentions are annotated in the same way, and candidate generation becomes the first step. Elasticsearch is utilized to construct index on Freebase. Given a mention with type, hundreds of candidate entities may be generated. To improve the efficiency of EL, the number of candidate entities is reduced to 15. As descripted in Section 3, the only input of ELSR is mention context and entity description. The mention context is the sentence where the mention appears rather than a fixed slice around the mention. Since the important information is concentrated at beginning, the entity description is the first sentence selected from its corresponding Freebase article. ELSR is first initialized based on the $\text{BERT}_{\text{BASE}}$, which has 768 hidden units, 12 transformer blocks, and 12 self-attention heads [26, 36]. Then, we apply the learning rate of $2e{-}5$ and the batch size of 16 to

Table 1: Dataset statistics.

Dataset	Doc	Men	Men/Doc
AIDA-Train	946	18448	19.5
AIDA-A	216	4791	22.1
AIDA-B	231	4485	19.4
KBP2016	168	5618	33.4
KBP2017	167	3683	22.0

Doc, Men, and Men/Doc denote number of documents, number of mentions, and average number of mentions per document, respectively.

fine-tune the model. The number of MLP layers is 2. When the epoch is set as 2, better performances can be obtained.

Since ELSR focuses on the sentence-level EK task, the length of the sentence will affect the results (too long will introduce noise, and too short will lose key information). Therefore, the effect of different max sentence length is explored on KBP2017 dataset. As given in Figure 4, accuracy reaches its peak when the max sentence length is 105. Besides, each layer of BERT provides different features. Commonly, the lower layers capture surface information, the middle layers give syntactic information, and the higher layers contain semantic information. Then, ELSR is fine-tuned with different layers. As given in Table 2, the feature from the eleventh layer of BERT obtains the best performance. Correspondingly, the mean of multiple layers, such as the first 4 layers, the last 4 layers, and all layers, provide poor results. Therefore, the following experiments are based on the eleventh layer.

4.2. Experimental Results. This section gives the empirical results of ELSR as well as baselines on two datasets from AIDA-B and KBP2017. Section 3.4 introduces three matching operations, including elementwise difference, elementwise product, and concatenation. Suppose "cat" represents concatenation and "$\circ$" and "$-$" refer to elementwise difference and product, respectively. In addition to the combining method described in equation (8), there are six other ways to combine two vectors. Therefore, it will generate some variants of ELSR. The first block of Table 3 gives the comparison of those baselines, which are on the basis of different matching operations. Here, all the baselines capture meaningful information over the Siamese network and BERT, abbreviated to SIABERT. When SIABERT is combined with separate matching operation, elementwise product obtains the highest accuracy of 90.33% on AIDA-B and elementwise difference yields the best performance with 81.40% accuracy on KBP2017. When the matching operations are combined in pairs, combining elementwise difference and elementwise product improves the accuracy to 91.29% and 83.62%.

Besides, in the second block of Table 3, ELSR is first compared with other Siamese frameworks, including Siamese-LSTM and Siamese-CNN. Two sentences can be encoded into sentence vectors to compute cosine similarity through both of the above models. We can see that Siamese-LSTM is superior to the Siamese-CNN, which implies that LSTM is more effective for encoding the sentence. In addition,

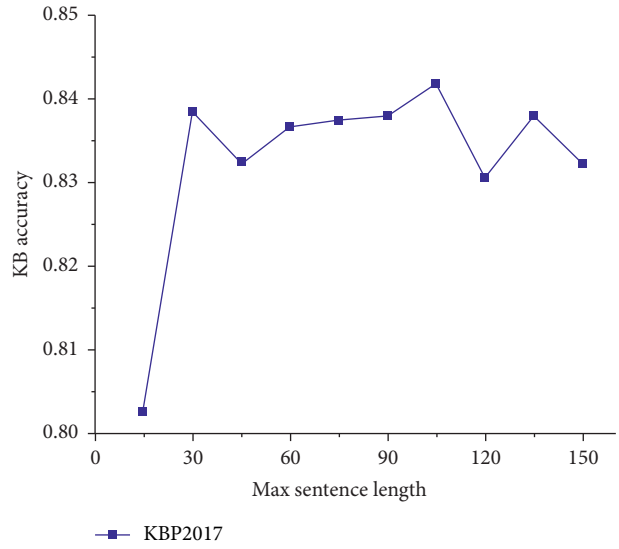

Figure 4: Results with different max sentence length on KBP2017 dataset.

Table 2: Fine-tuning BERT with different layers on KBP2017 dataset.

Layer	Accuracy (%)
Layer 1	77.21
Layer 2	79.00
Layer 3	77.98
Layer 4	81.53
Layer 5	82.08
Layer 6	79.12
Layer 7	76.18
Layer 8	80.93
Layer 9	81.59
Layer 10	82.67
Layer 11	**84.17**
Layer 12	77.65
First 4 layers	79.22
Last 4 layers	81.37
All layers	82.40

Best results are highlighted in bold.

Table 3: The accuracy results of all methods on the 2 public datasets.

Model	Accuracy (%)	
	AIDA-B	KBP2017
SIABERT + cat	87.50	78.52
SIABERT + $-$	90.26	81.40
SIABERT + $\circ$	90.33	78.00
SIABERT + cat, $\circ$	89.84	81.86
SIABERT + cat, $-$	89.86	79.66
SIA BERT + $-$, $\circ$	91.29	83.62
Siamese-BiLSTM [37]	78.10	68.48
Siamese-CNN [38]	77.75	65.14
BiMPM [20]	78.32	69.10
SSAMN [16]	88.20	83.70
TypeCoAtt + sparse (pretrain) [5]	89.80	—
Encoder + co-attention + decoder [21]	82.30	—
ELSR	**92.09**	**84.17**

Best results are highlighted in bold.

Table 4: The accuracy results based on different sentence representation strategies.

Strategy	Accuracy (%)	
	AIDA-B	KBP2017
Mean pooling	83.79	77.14
CLS	85.98	80.64
Interaction	**92.09**	**84.17**

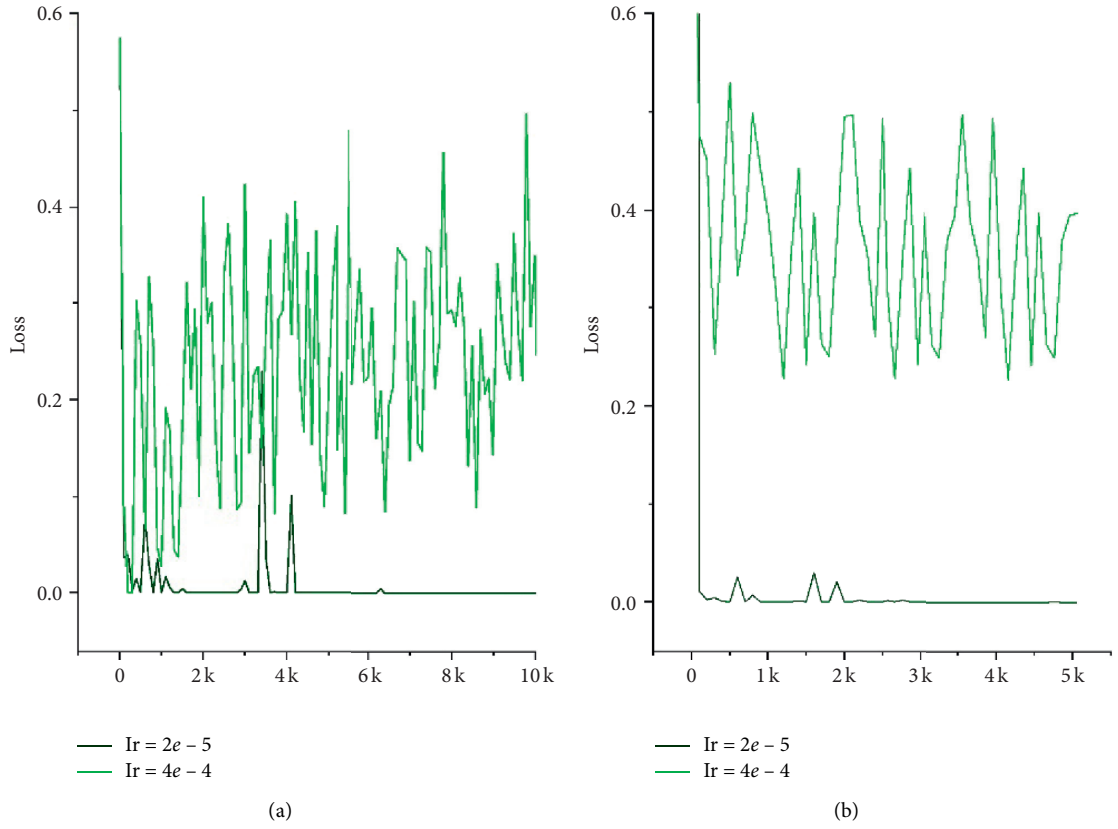

Figure 5: Test loss of ELSR using two different datasets. (a) Size of AIDA-B. (b) Size of KBP2017.

BiMPM is also a typical sentence similarity method [39]. BiMPM uses multiple perspectives to match sentences from two directions, and then the matching results are aggregated into a vector to make a decision. The above three approaches work worse than all the variants of ELSR, which indicates SIABERT is effective. SSAMN uses self-attention and memory network to link entity [35]. Both TypeCoAtt + sparse and encoder + co-attention + decoder employ co-attention mechanism to find important information from noisy text [6, 40]. Our ELSR not only outperforms TypeCoAtt + sparse and encoder + co-attention + decoder but also achieves slightly better performance than SSAMN on the base of the Siamese network, which indicates that our proposed model can improve the interaction between sentences from multiple levels of granularity. The final accuracy of ELSR on AIDA-B and KBP2017 reaches 92.09% and 84.17%, respectively. Therefore, ELSR is superior to all other methods.

4.3. Model Analysis

4.3.1. Analysis of Interaction. This part will explore the influence of interaction layer in our model. The interaction layer uses attention mechanism to learn some vital information and capture interactive information from the other sentence. To investigate effect of the interaction layer, the hidden state CLS and mean pooling of the eleventh layer are used to represent the sentence, respectively. As shown in Table 4, using mean pooling strategy hurts the performance for about 8% and 7% and using CLS hurts the performance for about 6% and 4%, respectively. Obviously, interaction layer can improve the sentence representation.

4.3.2. Analysis of Test Loss. The stochastic gradient descent is used to optimize the loss of ELSR with projection over sampled pairs (e^+, e^-). ELSR is fine-tuned using different learning rates. As observed from Figure 5, when the learning rate is small, the model begins to converge with the increase of datasets. However, for the learning rate of $4e{-}4$, the model fails to converge. Therefore, the learning rate is set to $2e{-}5$.

4.3.3. Error Analysis. This part gives the error analysis made by ELSR on the KBP2017 dataset. On the one hand, the mention which is expressed by its abbreviations or part of

full names is more likely to refer to the wrong entity. For example, if the context is "Which is why I think the switch to Intel has a little more to do with Steve Jobs himself than IBM's roadmap," the mention "IBM" refers to the entity "IBM" rather than the entity "IBM Global Services." The incomplete expression of the mention may weaken the contextual information and lead to an incorrect prediction. On the other hand, the misleading or missing context can also cause the wrong reference, e.g., a sentence heavily discussing about football will favor resolving the mention "Japan" to the entity "Japan national football team" instead of the gold entity "Japan."

5. Conclusions

When a mention appears multiple times in a document, the conflict of its contexts in different locations may lead to difficult linking. In this paper, ELSR is proposed to capture the semantic meaning of the sentences and reduce the noise introduced by different contexts of the same mention in a document. Compared to the traditional models, ELSR, which can generate the fixed-size vectors for input sentences, is efficient for measuring sentence similarity. Besides, adding attention mechanisms for the Siamese network not only captures vital information but also improves the interaction between input sentences. The difference and similarity between two sentences can be better captured. Therefore, ELSR outperforms in learning sentence similarity, and extensive experiments demonstrate our competitiveness based on the traditional Siamese network.

Although some key problems have been studied in this paper, there are still many limitations to overcome. The relations between textual mentions are ignored. In the future, we would like to combine our method with collective approach to explore the relatedness between entities. Meanwhile, mention recognition and entity generation leave some improvement space, which can be incorporated into the first step of entity linking. Due to the convincing performance of graph neural networks, we will also explore how to improve the entity linking based on them.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Bingjing Jia and Zhongli Wu contributed equally to this study.

Acknowledgments

This study was supported by the National Key R&D Program of China (no. 2018YFC0831500), the National Natural Science Foundation of China (no. 61972047), the Key Project of Natural Science Research of Universities in Anhui (nos. KJ2020A0062 and KJ2016A176), and the Key Project of University Outstanding Young Talents Project in Anhui (no. gxyqZD2018069).

References

[1] J. Ma, Y. Qiao, and G. Hu, "ELPKG: a high-accuracy link prediction approach for knowledge graph completion," *Symmetry*, vol. 11, no. 9, 2019.

[2] L. Ratinov, D. Roth, and D. Downey, "Local and global algorithms for disambiguation to Wikipedia," in *Proceedings of the Meeting of the Association for Computational Linguistics*, pp. 1375–1384, August 2011, Bangkok, Thailand.

[3] P. Sen, "Collective context-aware topic models for entity disambiguation," in *Proceedings of the The Web Conference*, pp. 729–738, April 2012, Lyon, France.

[4] W. Shen, J. Wang, and J. Han, "Entity linking with a knowledge base: issues, techniques, and solutions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 2, pp. 443–460, 2015.

[5] D. N. Milne and I. H. Witten, "Learning to link with wikipedia," in *Proceedings of the Conference on Information and Knowledge Management*, pp. 509–518, October 2008, Napa Valley, CA, USA.

[6] F. Nie, Y. Cao, and J. Wang, *Mention and Entity Description Co-Attention for Entity Disambiguation*, Association for the Advancement of Artificial Intelligence, Menlo Park, CA, USA, 2018.

[7] S. Chen, J. Wang, and F. Jiang, "Improving entity linking by modeling latent entity type information," in *Proceedings of the Conference on Artificial Intelligence*, February 2020, New York, NY, USA.

[8] N. Gupta, S. Singh, and D. Roth, "Entity linking via joint encoding of types, descriptions, and context," in *Proceedings of the Empirical Methods in Natural Language Processing*, pp. 2681–2690, September 2017, Copenhagen, Denmark.

[9] Z. Guo and D. Barbosa, "Robust named entity disambiguation with random walks," *Sprachwissenschaft*, vol. 9, no. 4, pp. 459–479, 2017.

[10] Y. Li, S. Tan, and H. Sun, "Entity disambiguation with linkless knowledge bases," in *Proceedings of the The Web Conference*, pp. 1261–1270, April 2016, Montréal, Québec, Canada.

[11] Z. Zheng, F. Li, and M. Huang, "Learning to link entities with knowledge base, human language technologies," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, June 2010, Los Angeles, CA, USA.

[12] X. Han and L. Sun, "An entity-topic model for entity linking," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, July 2012, Los Angeles, CA, USA.

[13] A. Alhelbawy and R. Gaizauskas, "Graph ranking for collective named entity disambiguation," in *Proceedings of the Meeting of the Association for Computational Linguistics*, June 2014, Baltimore, MA, USA.

[14] Z. He, S. Liu, and M. Li, "Learning entity representation for entity disambiguation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, vol. 2, pp. 30–34, June 2013, Sofia, Bulgaria.

[15] M. Francislandau, G. Durrett, and K. Dan, "Capturing semantic similarity for entity linking with convolutional bneural networks," in *Proceedings of the 2016 Conference of the North*

American Chapter of the Association for Computational Linguistics, pp. 1256–1261, Human Language Technologies, June 2016, San Diego, CA, USA.

[16] Y. Sun, Z. Ji, L. Lin, X. Wang, and D. Tang, "Entity disambiguation with memory network," *Neurocomputing*, vol. 275, pp. 2367–2373, 2018.

[17] O. E. Ganea and T. Hofmann, "Deep joint entity disambiguation with local neural attention," 2017, https://arxiv.org/abs/1704.04920.

[18] Z. Fang, Y. Cao, and Q. Li, "Joint entity linking with deep reinforcement learning," in *Proceedings of the World Wide Web Conference*, pp. 438–447, May 2019, San Francisco, CA, USA.

[19] Y. Cao, L. Hou, and J. Li, "Neural collective entity linking," 2018, https://arxiv.org/abs/1811.08603.

[20] J. Wu, R. Zhang, and Y. Mao, "Dynamic graph convolutional networks for entity linking,," in *Proceedings of The Web Conference*, pp. 1149–1159, April 2020, Ljubljana, Slovenia.

[21] Z. Fang, Y. Cao, and R. Li, "High quality candidate generation and sequential graph attention network for entity linking," in *Proceedings of The Web Conference*, pp. 640–650, April 2020, Ljubljana, Slovenia.

[22] I. Yamada, H. Shindo, and H. Takeda, "Joint learning of the embedding of words and entities for named entity disambiguation," 2016, https://arxiv.org/abs/1601.01343.

[23] I. Yamada, H. Shindo, and H. Takeda, "Learning distributed representations of texts and knowledge base," 2017, https://arxiv.org/abs/1705.02494.

[24] Y. Cao, J. Shi, and J. Li, "On modeling sense relatedness in multi-prototype word embedding," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, vol. 1, pp. 233–242, November 2017, Taipei, Taiwan.

[25] S. Hu, Z. Tan, and W. Zeng, "Entity linking via symmetrical attention-based neural network and entity structural features," *Symmetry*, vol. 11, no. 4, 2019.

[26] J. Devlin, M. W. Chang, K. Lee, and Bert, "Pre-training of deep bidirectional transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.

[27] S. Li, W. Cui, and Y. Liu, "PEL-BERT: a Joint model for protocol entity linking," 2020, https://arxiv.org/pdf/2002.00744.

[28] S. Chen, J. Wang, and F. Jiang, "Improving entity linking by modeling latent entity type information," 2020, https://arxiv.org/abs/2007.13778.

[29] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4353–4361, June 2015, Seattle, WA, USA.

[30] Y. Liu, M. Gardner, and M. Lapata, "Structured alignment networks for matching sentences," in *Proceedings of the Empirical Methods in Natural Language Processing*, pp. 1554–1564, November 2018, Brussels, Belgium.

[31] L. Mou, R. Men, and G. Li, "Natural language inference by tree-based convolution and heuristic matching," 2015, https://arxiv.org/abs/1512.08422.

[32] Y. Sun, L. Lin, and D. Tang, "Modeling mention, context and entity with neural networks for entity disambiguation," in *Proceedings of the International Conference on Artificial Intelligence*, pp. 1333–1339, June 2015, San Diego, CA, USA.

[33] J. Hoffart, M. A. Yosef, and I. Bordino, "Robust disambiguation of named entities in text," in *Proceedings of the Empirical Methods in Natural Language Processing*, pp. 782–792, November 2011, Hong Kong, China.

[34] H. Ji and J. Nothman, "Overview of TAC-kbp2016 tri-lingual EDL and its impact on end-to-end KBP," *Theory and Applications of Categories*, 2016.

[35] Y. Hu, "Research and implementation of English entity discovery and linking system based on freebase," Master's Thesis, Beijing University of Posts and Telecommunications, Beijing, China, 2019.

[36] A. Vaswani, N. Shazeer, and N. Parmar, "Attention is all you need," in *Proceedings of the Neural Information Processing Systems*, pp. 5998–6008, December 2017, Long Beach, CA, USA.

[37] B. Jin and Z. Jin, "Car FAQ assistant based on BILSTM-siamese network," *OALib-Open Access Library Journal*, vol. 06, no. 10, pp. 1–7, 2019.

[38] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, AZ USA, February 2016.

[39] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 4144–4150, August 2017, Melbourne, Australia.

[40] S. Zhang, J. Lou, X. Zhou, and W. Jia, "Entity linking facing incomplete knowledge base," in *Proceedings of the Web Information Systems Engineering - WISE 2018*, pp. 325–334, January 2018, Hong Kong, China.