IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 28, 2018, accepted May 1, 2018, date of publication May 4, 2018, date of current version June 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2833153

Entity Linking on Chinese Microblogs via Deep Neural Network

WEIXIN ZENG[1], **JIUYANG TANG**[1,2], **AND XIANG ZHAO**[1,2]
[1]Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China
[2]Collaborative Innovation Center of Geospatial Technology, Wuhan 430079, China

Corresponding author: Xiang Zhao (xiangzhao@nudt.edu.cn)

This work was supported by NSFC under Grant 71690233 and Grant 71331008.

ABSTRACT Entity linking is the task of mapping mentions in text to target knowledge base, which is crucial to knowledge-base-related tasks such as knowledge fusion and knowledge base construction. Although English-oriented entity linking task has undergone continuing advancement, the entity linking systems targeted at Chinese language still suffer from lagged development. State-of-the-art Chinese entity linking systems devise multiple handcrafted features to measure similarity between mention and entity, whereas fail to mine semantic relations underneath the surface forms. In this paper, we propose to take the advantage of latent text features and generate representations of mention and entity via double-attention-based long short term memory network, which are further utilized to calculate mention-entity similarity. Furthermore, joint word and entity embedding training and well-designed candidate entities generation strategies are put forward to facilitate the implementation of neural network. The experimental results validate the superiority of our method Celan. Our proposal not only offers an improved deep neural network for generating mention and entity representation, but also enhances the performance of entity linking on Chinese microblogs.

INDEX TERMS Entity linking, named entity disambiguation, neural network, Chinese microblogs.

I. INTRODUCTION

With the explosive increase of unstructured data on the Internet, automatic extraction and formalization of valuable information becomes incrementally more crucial. Under this circumstance, knowledge base (KB), which can structure and organize emerging information and present them in an approachable way, was proposed and underwent continuing development. In the process of knowledge base construction and update, entity linking plays an indispensable role by connecting diverse texts with structured knowledge. Entity linking (EL) is the task of determining corresponding entities for mentions in texts. Entities are unique identifiers of objects, while mentions, which are surface forms of entities, can be ambiguous and misleading. The aim of entity linking task is to erase the ambiguity of mentions. Figure 1 presents an example of mapping mentions in a piece of weibo[1] (Chinese microblog) to entities in Chinese Wikipedia.

Despite the advancement of entity linking techniques, most of the works are focused on English corpus, which could not be directly applied to other languages due to two prominent factors, namely, various language features and uneven quality of target knowledge bases. A case in point is Chinese. Different from languages similar to English, Chinese words can be composed of multiple characters but with no space appearing between words [1], which is termed as Chinese word segmentation problem. As a result, Chinese entity linking techniques inevitably need to cope with word segmentation problem and minimize its error propagation. Furthermore, for the time being, there is no fully-accessible Chinese knowledge base with high quality, giving rise to the lagged development of Chinese entity linking systems.

State-of-the-art Chinese entity linking methods design effective handcrafted features to measure the similarity between mention and candidate entities, which are further combined and utilized as indicator for candidate entities ranking. Nevertheless, on the one hand, the significance of feature engineering is highlighted in current approaches, which is complicated and might not guarantee promising results. On the other, the frequently used bag-of-words (BOW) representations of mention and entity context cannot well capture the deep semantic meanings underneath text, which are crucial to enhancing the accuracy of mention-entity context similarity. Therefore, to improve the effectiveness of Chinese EL system, deep neural networks and word embedding

[1] https://weibo.com/

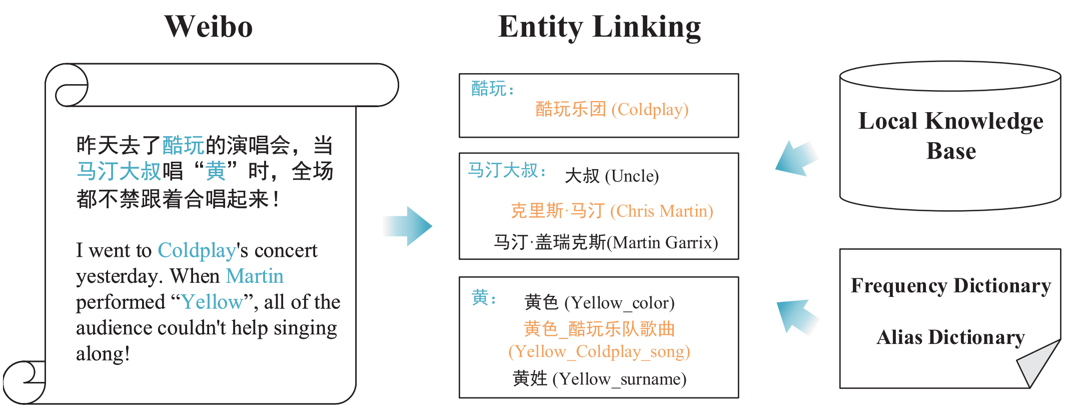

Weibo
Entity Linking
昨天去了酷玩的演唱会，当马汀大叔唱“黄”时，全场都不禁跟着合唱起来！
I went to Coldplay's concert yesterday. When Martin performed “Yellow”, all of the audience couldn't help singing along!
酷玩：
酷玩乐团 (Coldplay)
马汀大叔：
大叔 (Uncle)
克里斯·马汀 (Chris Martin)
马汀·盖瑞克斯(Martin Garrix)
黄：
黄色 (Yellow_color)
黄色_酷玩乐队歌曲 (Yellow_Coldplay_song)
黄姓 (Yellow_surname)
Local Knowledge Base
Frequency Dictionary
Alias Dictionary

FIGURE 1. **Linking a piece of weibo to local knowledge base.**

techniques could be harnessed to make up for the aforementioned deficiencies. Additionally, current Chinese entity linking datasets are constructed upon partial outdated knowledge bases, which could restrain the performance of EL. It is significant to re-annotate the datasets to a large-scale substitute and lay the foundation for future development. Example 1 elaborates part of the difficulties posed by Chinese EL and the limitations of current techniques.

Example 1: As shown in Figure 1, there is a piece of weibo with mentions 酷玩(Coldplay), 马汀大叔(Martin) and 黄(Yellow). The first step is to generate candidate entities for each mention. Both mention 酷玩(Coldplay) and 黄(Yellow) can find corresponding entities in Chinese Wikipedia through simply querying their surface forms. Nonetheless, there are no entities for mention 马汀大叔(Martin) since 马汀大叔(Martin) is a friendly way of calling a person named 马汀(Martin), which refers to entity 克里斯·马汀(Chris Martin). In this situation, by harnessing Chinese word segmentation techniques to split mention 马汀大叔(Martin) into 马汀(Martin) and 大叔(Uncle), the candidate entities for segments can be regarded as candidates for the original form. Eventually the correct entity 克里斯 · 马汀(Chris Martin), along with other noisy entities generated by the segments, form the candidates for mention 马汀大叔(Martin).

Then the crucial step is to determine the correct result from the set of candidate entities. Take mention 黄(Yellow) for instance, there are three candidate entities, namely, 黄色(Yellow-color), 黄色-酷玩乐队歌曲(Yellow-Coldplay-song), and 黄姓(Yellow-surname). One of the most important features contributing to the candidate ranking mechanism is the similarities between mentions and candidate entities' text representations. In this case the text for mention 黄(Yellow) is the weibo context, while for entities the texts are the first paragraph of the their Wikipedia pages. As can be seen from figure 1, the Weibo context is noisy and short, the semantic meaning of which might not be fully captured via current BOW representation, while well-trained neural networks can generate preciser representation by utilizing hidden text features.

In short, the shortcomings of existing Chinese EL solutions are two-fold:

- Mention and entity representations cannot capture deep semantic meanings of text and require complex feature engineering; and
- Results were achieved upon partial and outdated knowledge bases, restraining the effectiveness of newly proposed methods.

In this work, we put forward an efficient Chinese entity linking system leveraging attention-based deep neural network(DNN), namely, Celan, to overcome the aforementioned shortcomings. Specifically, Celan comprises three steps: (1) Candidate entities generation. By taking advantage of four efficient strategies, this step generates candidate entities for mentions and enhances upper bound of linking accuracy. (2) Joint word and entity embeddings training. This step maps word and entity into the same high-dimensional space to better represent their semantic meanings, which serve as inputs for the neural network. (3) Entity Disambiguation via DNN with double-attention mechanism. We harness LSTM to generate representations of mentions and entities, which are further utilized to model semantic similarity between mentions and entities. To balance the contributions from different words to the abstract representations of mention context and entity description, we propose a double-attention mechanism, which also renders the whole neural network more effective.

Furthermore, considering the lack of large-scale and fully-accessible Chinese knowledge base, we resort to Chinese Wikipedia, which can be regarded the best KB satisfying the two criteria mentioned before. Meanwhile, we transfer the annotations of current Chinese entity linking datasets to Chinese Wikipedia and make the updated datasets public.[2] The experimental results on these datasets validate the effectiveness of Celan, and the in-depth analysis shows that compared with state-of-the-art Chinese entity linking methods, Celan achieves better performance by offering a more accurate semantic matching solution.

[2] https://github.com/DexterZeng/chineseEL-datasets

TABLE 1. Mention regularization.

Original Mentions	Processed Mentions	Type
《乱世佳人》(Gone with the Wind)	乱世佳人(Gone with the Wind)	Removing useless punctuations
史蒂文．福斯特(Stephen Foster)	史蒂文 · 福斯特(Stephen Foster)	Regularizing foreign names
强尼戴普(Johnny Depp)	强尼 · 戴普(Johnny Depp)	Regularizing foreign names
皇帝詹姆斯(James the “emperor")	皇帝(emperor)	Splitting compounds
	詹姆斯(LeBron James)	Splitting compounds

A. CONTRIBUTIONS

The main contributions of this article can be summarized into three ingredients:

- We propose an improved LSTM framework with double attention mechanism to model abstract representations of mentions and entities.
- Current Chinese EL datasets are revised and publicised in accordance to a fully-accessible and large-scale knowledge base (Chinese Wikipedia).
- To the best of our knowledge, we are the first to utilize attention-based DNN in Chinese entity linking task, and experimental results on the revised datasets validate the strengths of our method.

B. ORGANIZATION

This paper is organized as follows. Section 2 introduces the methodology, including candidate entities generation, joint embeddings training and entity disambiguation via DNN. Experimental settings and results are detailed in Section 3. Section 4 summarizes related work, followed by conclusion in Section 5.

II. METHODOLOGY

The work flow of our linking system initiates from candidate entities generation. This step requires the aid from local knowledge base, frequency dictionary and alias dictionary. The former two tools, along with the raw inputs for embedding training, are derived from Chinese Wikipedia dump on 01-Dec-2017.[3] Then the well-trained LSTM with double attention mechanism converts the mentions and entities into abstract representations, and outputs the similarity scores which are utilized to rank the candidate entities and generate final results. The specific approaches are detailed as follows.

A. CANDIDATE ENTITIES GENERATION

The initial input for EL framework is a set of raw documents, which could be long passages such as news reports, or short texts in the form of twitters. Additionally, the inputs might come with specified mentions to be disambiguated or without. Under the circumstance where mentions are not pointed out, the Named Entity Recognition (NER) techniques would be utilized to detect mentions in the texts. Current Chinese EL datasets are constructed upon Chinese weibo, which is akin to twitter and contains noisy short texts. The mentions in the datasets have already been extracted, thereby excluding NER from our entity linking framework. Figure 1 shows a piece of the input.

Then the candidate entities could be retrieved by querying the mentions in the local knowledge base. For English EL tasks, the local knowledge base could be derived from some downloadable KBs such as YAGO[4] and DBpedia,[5] or even English Wikipedia. On the contrary, there are few large-scale Chinese KBs. CN-DBpedia [2], which is the largest and state-of-the-art Chinese KB, only offers a set of APIs for free use. Other KBs such as Zhishi.me[6] and XLore,[7] are either not frequently updated or in a relatively small scale. Nonetheless, the Chinese Wikipedia, is not only revised and updated on a daily basis, but also fully accessible through dumps. Consequently, in order to obtain a up-to-date local Chinese KB, we downloaded and parsed the Chinese Wikipedia dump and imported it into a MySQL database, which was utilized as the target local KB in this work. The constructed local KB can not be made publicly available due to its huge size, whereas the construction process is elaborated in Section 3 for reproduction.

The simplest approach to generate candidate entities is to directly query mention name without modification, which in many cases would get no results, since mentions can appear in diverse forms, especially in Chinese. In the interest of improving the recall of entities generation, we devise following four strategies, the effectiveness of which are embodied in the experiment section.

1) STRATEGY 1: MENTION REGULARIZATION

First and foremost, the mentions should be regularized before candidates generation. Retrieved from Weibo, where the texts are short and noisy, mentions are accordingly in irregular forms. As is depicted in Table 1, some mentions contain useless punctuations, while foreign names appear in non-uniform ways. We unify the various mention representations and display some typical examples in Table 1.

Besides, we also notice that some mentions are compound words coined by Internet users, which do not have direct counterparts in KB. Thereafter, a Chinese segmentation tool is harnessed to split those compounds and in addition to querying compound mentions, the split words are also queried.

[3] https://dumps.wikimedia.org/zhwiki/20171201/

[4] https://old.datahub.io/dataset/yago

[5] http://wiki.dbpedia.org/develop/datasets

[6] http://zhishi.me/

[7] http://xlore.org/

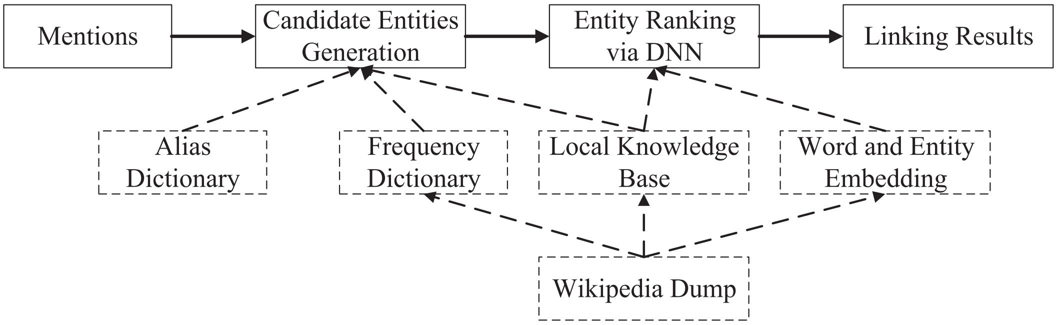


FIGURE 2. Work flow of Celan.

TABLE 2. Frequency dictionary.

Mentions	**Possible Entities**	Frequencies
李娜**(Li Na)**	李娜-网球运动员(Li Na -Tennis-player)	48
	李娜-击剑运动员(Li Na -Fencer)	8
	李娜-歌手(Li Na -Singer)	3
	...	...

TABLE 3. Alias dictionary.

Formal entity name	**Possible aliases.**
沙奎尔 · 奥尼尔(Shaquille O’Neal)	大鲨鱼(Giant “Shark”), 大柴油机(Giant “Diesel”) ...
克里斯 · 马汀(Chris Martin)	马山芋(Martin) ...
梁启超(Liang Qichao)	卓如(Zhuoru); 任公(Rengong) ...
石田彰(Ishida Akira)	石头(“Stone”) ...
...	...

2) STRATEGY 2: FREQUENCY DICTIONARY

Similar to state-of-the-art entity linking methods [3]–[5], we construct a name dictionary for formalizing irregular forms of mentions. Specifically, the elements in the dictionary are obtained from Wikipedia pages. Aside from normal words, Wikipedia pages also contain anchor texts, which are attached with links directing at other pages. Since each Wikipedia page represents a unique entity, the anchor text accordingly could be regarded as surface form of the entity it points to. As thus, we can attain a dictionary consisting of surface forms and their possible referent entities.

Furthermore, the frequencies of referent entities given a surface form are also recorded, since they serve as indicators of the most possible entities for a mention. In all, a frequency dictionary is derived from mining anchor texts of Wikipedia pages, which not only help generate candidate entities for mentions, but also contribute to the calculation of prior probability in the following ranking process. Part of the frequency dictionary is presented in Table 2.

3) STRATEGY 3: WIKIPEDIA FUNCTIONAL PAGES

Wikipedia contains rich semantic structures, such as disambiguation pages (polysemy), redirect pages (synonym) and hyperlinks between Wikipedia [6], which are of great benefit to retrieving entities for ambiguous mentions. We extend Example 1 to further elaborate this strategy.

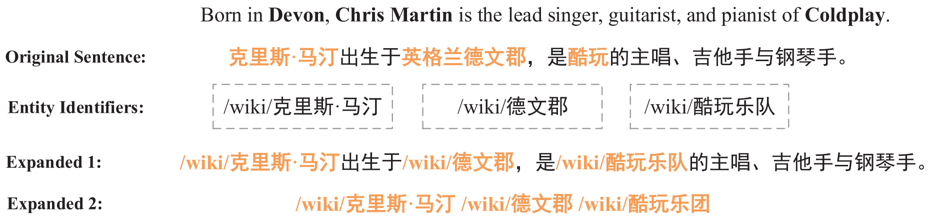
Example 2: When given mention 酷玩, there is no direct Wikipedia page corresponding to this surface form. Nevertheless, there is a redirect page in Wikipedia indicating that 酷玩 is an abbreviation referring to entity 酷玩乐团(Coldplay), thus generating entity 酷玩乐团(Coldplay) for mention 酷玩.

In addition, directly querying mention 黄(Yellow) would result in entity 黄色(Yellow-color) standing for the color. However, 黄(Yellow) also have other meanings, which are recorded in Wikipedia disambiguation pages. In the disambiguation page featuring 黄(Yellow), there are entity 黄色(Yellow-color), entity 黄色-酷玩乐队歌曲(Yellow-Coldplay-song), entity 黄姓(Yellow-surname) and some more. We take advantage of disambiguation pages to improve the coverage of candidate entities.

4) STRATEGY 4: ALIAS DICTIONARY

With the aid of frequency dictionary and Wikipedia functional pages, the candidate entities for most of the mentions would be generated. Nevertheless, some mentions are aliases of their target entities, and simply utilizing the aforementioned strategies could not retrieve the true entity. As a consequence, we construct an alias dictionary to achieve this task.

The alias dictionary is composed of formal entity names and their potential aliases. The information is mined from the infobox of Wikipedia and Baidu Baike, which is the largest



Born in **Devon**, **Chris Martin** is the lead singer, guitarist, and pianist of **Coldplay**.

Original Sentence: 克里斯·马汀出生于英格兰德文郡，是酷玩的主唱、吉他手与钢琴手。

Entity Identifiers: /wiki/克里斯·马汀 /wiki/德文郡 /wiki/酷玩乐队

Expanded 1: /wiki/克里斯·马汀出生于/wiki/德文郡，是/wiki/酷玩乐队的主唱、吉他手与钢琴手。

Expanded 2: /wiki/克里斯·马汀 /wiki/德文郡 /wiki/酷玩乐团

FIGURE 3. Corpus expansion.

Chinese encyclopaedia. Part of the alias dictionary is shown in Table 3.

B. JOINT WORDS AND ENTITIES EMBEDDINGS LEARNING

After the generation of candidate entities, the next step is to rank the candidates and determine the most possible one via deep neural network. As inputs of DNN, the embeddings of word and entities are crucial to its final performance. Embeddings are n-dimensional vectors of concepts which describe the similarities between these concepts using cosine similarity [7]. It is assumed that the concepts are similar if they frequently co-occur with the same other concepts. This has already been well researched for words [8] and documents [9] in literature. Considering the fact that each entity has its specific surface form composed of words, an intuitive idea of representing entity would be the addition of word embeddings. However, it is evident that entities have their unique features and individually representing them would be more reasonable.

In this work, in line with recent entity linking systems [3], [10], we propose an embedding method that jointly maps words and entities into the same continuous vector space, where similar words and entities are placed close to one another.

The joint embedding method is derived from conventional skip-gram model [8] for learning word embedding. The training objective of skip-gram model is to generate word representations that can help predict context words given a specific word. Formally, let $O = w_1, w_2, \ldots w_N$ be a sequence of words, the model aims to maximize the following average log probability:

$$\Theta_w = \frac{1}{N}\sum_{i=1}^{N}\sum_{-c\leq j\leq c, j\neq 0}\log P(w_{i+j}|w_i). \tag{1}$$

where c is the size of context window, w_i represents the target word and w_{i+j} is a context word. The conditional probability is defined by the following softmax function:

$$P(w_{t+j}|w_t) = \frac{\exp(\mathrm{V}_{w_t}{}^{\top}\mathrm{U}_{w_{t+j}})}{\sum_{w\in W}\exp(\mathrm{V}_{w_t}{}^{\top}\mathrm{U}_{w_{t+j}})}. \tag{2}$$

where W represents the set of all words in the vocabulary, V_w and U_w denote the vectors of word w in matrices V and U. After training, the matrix V is used for word embedding.

Then the conventional model is extended to joint embedding model. First, we introduce how to create the corpus for joint embedding training. Specifically, the texts in Wikipedia pages are comprised of words and anchor texts. Utilizing the link associated with each anchor text, the corresponding entity identifier for the anchor text could be attained. As is displayed in Figure 3, substituting anchor texts with entity identifiers, the expanded sentences (Expanded 1) for joint embeddings can thus be generated. Plus, we also extract entity identifiers from the original sentences and form new inputs to better capture the relations among entities (Expanded 2).

In the joint training corpus, entity name is treated as a special form of word, and accordingly equations 1 and 2 can be modified as follows to integrate entities:

$$\Theta_a = \frac{1}{N}\sum_{i=1}^{N}\sum_{-c\leq j\leq c, j\neq 0}\log P(\tau_{i+j}|\tau_i). \tag{3}$$

$$P(\tau_{t+j}|\tau_t) = \frac{\exp(\mathrm{V}_{\tau_t}{}^{\top}\mathrm{U}_{\tau_{t+j}})}{\sum_{\tau\in\Gamma}\exp(\mathrm{V}_{\tau_t}{}^{\top}\mathrm{U}_{\tau_{t+j}})}. \tag{4}$$

where $\Lambda = \tau_1, \tau_2, \ldots \tau_N$ is a sequence of tokens (words or entity names), τ_i represents the target token and τ_{i+j} is a context token. Γ represents the set of all tokens in the corpus, V_τ and U_τ denote the vectors of token τ in matrices V and U. After training, the matrix V is used for joint word and entity embedding.

The merits of jointly representation learning are threefold: (1) The final word embeddings are conceptually more accurate as the mentions in their context, which might appear in various surface forms, are replaced by constant entity identifiers; (2) Compared with the relatively small corpus derived from knowledge bases, the resulting entity embeddings are learned over a large text corpus and have higher frequencies during training process; (3) Since the representations of words and entities are learned in the same space, the measurement of similarities between words, entities, a word and an entity can be easily achieved by cosine similarities.

C. ENTITY DISAMBIGUATION VIA DNN

Traditional methods utilize various discrete and hand-designed features/heuristics to measure similarities between mentions and entities for candidate entities ranking. Nevertheless, they might suffer from the data sparseness issue caused by unseen words or features, the difficulty of calibrating, and the incapability to generate underlying similarity

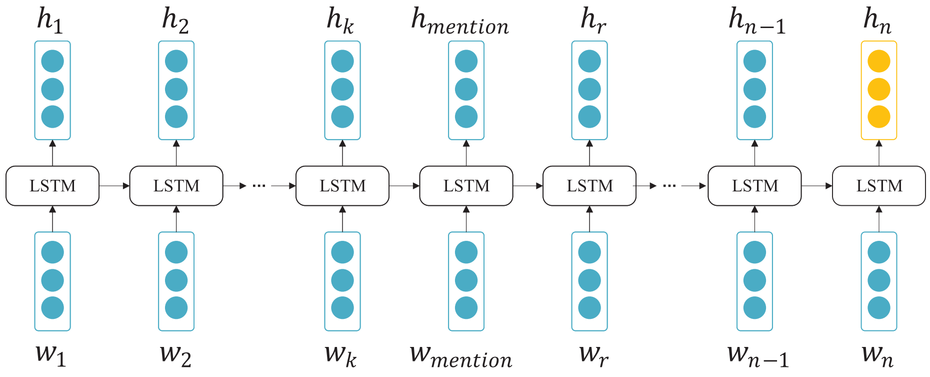


FIGURE 4. Basic LSTM structure.

structures at high levels of abstraction for EL due to the reliance on hand-designed coarse features [11].

Recently, with the proliferation of deep neural networks, many EL systems [11]–[13] leverage DNNs to automatically extract hidden features in texts to enable accurate similarity computation and candidate entities ranking. Nevertheless, the existing solutions do not make full use of the information embedded in texts. On one level, the mention's position is ignored in previous models. As a result, if the context consists of two or more mentions, all mentions are viewed as identical, which is clearly impractical. On the other, existing approaches neglect the word order in contexts, which has proved to be critical for natural language understanding [14].

The attention-based LSTM model proposed in [3] overcomes aforementioned shortcomings and achieves state-of-the-art performance. In this work, we put forward Celan, which outperforms previous model by integrating double attention mechanism and optimizing the inputs of LSTM.

1) LONG SHORT-TERM MEMORY (LSTM)

Recurrent Neural Network(RNN) is an extension of conventional feed-forward neural network, whereas it has gradient vanishing or exploding problems. In order to overcome these issues, Long Short-term Memory network (LSTM) is developed, which has achieved superior performance, especially in modelling variable-length sequence. In the LSTM architecture, there are three gates and a cell memory state. Figure 4 illustrates the architecture of a standard LSTM.

Specifically, given input sequence $S = w_1, w_2, \ldots, w_n$, where w_t is a word embedding to be passed as input at time t, the LSTM unit will output h_t for each time step t. The hidden vector h_t is computed as follows:

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \tag{5}$$

$$f_t = \sigma(W_f \cdot X + b_f) \tag{6}$$

$$i_t = \sigma(W_i \cdot X + b_i) \tag{7}$$

$$o_t = \sigma(W_o \cdot X + b_o) \tag{8}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c \cdot X + b_c) \tag{9}$$

$$h_t = o_t \odot \tanh(c_t) \tag{10}$$

where i, f, o, C are input gate, forget gate, output gate, and cell memory respectively. σ denotes sigmoid function, while $\odot$ stands for element-wise multiplication. W_i, W_f, W_o are weighted matrices and b_i, b_f, b_o represent the biases of the LSTM network, which are parameters to be learned during training. Standard LSTM network regards the last hidden vector h_n as the representation of text, as is highlighted by different color in Figure 4. Nevertheless, we follow previous EL work [3] and use the max-pooling result of all hidden states over time steps i.e, max-pooling$(h_1, \ldots, h_n)$ to produce a fixed-length final representation.

2) THE FRAMEWORK OF DOUBLE ATTENTION BASED LSTM

We present the framework of double attention based LSTM in Figure 5. The framework consists of three basic LSTM units, which were utilized to model the left context of mention, the right context of mention, and entity profile, respectively. Then the max-pooling result of the two LSTMs for mention are concatenated as the final representation of mention, while the final representation of entity is made up of the max-pooling result of entity description LSTM and the entity embedding. The similarity score is generated by forwarding the concatenation of mention and entity representations to two fully connected layers. A double-attention mechanism is embedded in the framework to alleviate the negative effect imposed by noisy information and improve model performance. The components are detailed as follows:

a: MENTION REPRESENTATION

Considering that mention context is of diverse length, we use the words within a window size c on both sides of mention as its contexts, which are denoted as $w_1, w_2, \ldots, w_c, w_{mention}$ and $w_{mention}, w_{c+1}, w_{c+2}, \ldots, w_{2c}$ for left and right context respectively. Note that although the mention is represented as a single token $w_{mention}$ here, it can be composed of more than one word. Two separate LSTMs are utilized to model left context and right context, and the right context sequence is reversed as $w_{2c}, w_{2c-1}, , \ldots, w_{c+1}, w_{mention}$ on account of two reasons. On the one hand, putting mention string as the last unit could better utilize the semantics of mention [15].

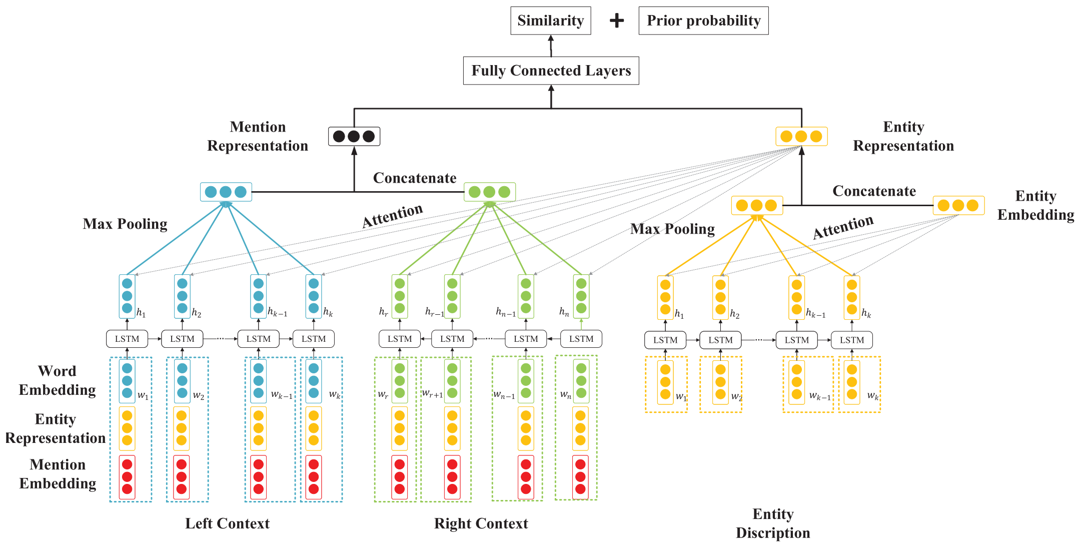


FIGURE 5. The framework of LSTM with double attention mechanism.

On the other, LSTMs can be aware of the mention position by aligning mention at the end of sequence. We use Example 3 to further explain this intuition.

Different from [3], we do not directly use word embeddings of context as inputs for LSTMs since it does not capture the interactions among mention, entity, and contexts. Instead, each word embedding is concatenated with mention embedding and entity representation before passing to the neural network, which can explicitly utilizes the connections between entity representation, mention embedding and each context word when composing the representation of a mention.

Then we use the max-pooling result of all hidden states over time steps as the representations of left context and right context. The representations of context on the two sides are concatenated to generate the fixed-length mention representation.

Example 3: Given the sentence "唱歌的李娜和打网球的李娜不是同一个人(Li Na who sings and Li Na who plays tennis are not the same person)" with two different mentions sharing the same name 李娜(Li Na), if the position of mention is not specified, the two mentions would have the same context and be linked to the same entity. Nonetheless, adopting our strategy to individually model the left and right context will lead to different inputs for the two mentions 李娜(Li Na), and accordingly individual linking results.

b: ENTITY REPRESENTATION

Entity representation is the combination of entity embedding and entity description representation. The entity embedding is obtained from the results of joint word and entity embedding training. Since entities are regarded similar to words in training, an entity embedding encodes both syntactic and semantic information about how the associated entity is mentioned. The entity embedding itself cannot fully reflect the information of an entity, thereby requiring more contents to enrich entity representation. Specifically, we set a window size of p to extract entity description from the first paragraph of its Wikipedia page. After word segmentation and embedding, the inputs are forwarded to another LSTM with max-pooling mechanism to generate entity description representation. Eventually the concatenation of entity embedding and entity description embedding is considered as the entity representation.

c: DOUBLE ATTENTION MECHANISM

The standard LSTM cannot detect important components in the inputs, giving rise to the noisy mention/entity representations. In order to address this issue, we propose to design a double attention mechanism that can capture the key parts of inputs in response to a given entity embedding/entity representation. Concretely, in Celan, the entity embedding is used as attention vector to highlight the informative parts in entity description, resulting to a preciser representation of entity description. Additionally, we harness the entity representation as the second attention vector to extract significant parts in mention context, so as to get rid of irrelevant information in the mention representation. The visual explanation is presented in Figure 5.

Suppose $H \in R^{d\times N}$ is a matrix comprising hidden vectors $(h_1, \ldots, h_N)$ produced by LSTMs (left context, right context or entity description), where d is the size of hidden layers and N is the length of sequence. The attention mechanism will produce the weighted hidden representation $\bar{H}$.

$$Z = \tanh \begin{bmatrix} W_h H \\ W_v v_a \otimes e_N \end{bmatrix} \tag{11}$$

$$\alpha = soft\max(w^{\mathrm{T}} Z) \tag{12}$$

$$\bar{H} = H\alpha^{\mathrm{T}} \tag{13}$$

where v_a represents the attention vector (entity embedding or entity representation), e_N is the vector of 1s, α represents weight vector of attention. W_h and W_v are parameters to be learned. Then the re-weighted $\bar{H}$ is forwarded to max-pooling process. Note that $v_a \otimes e_N = [v_a; v_a; \ldots; v_A]$, which means $\otimes$ repeatedly concatenates the vector on the other side of the operator. In a nutshell, we utilize a unified expression of equations to illustrate how double attention mechanism works. The idea of exerting attention on the context can eliminate noise and optimize mention/entity representation, hence improving the accuracy of similarity computation.

d: FULLY CONNECTED LAYERS

The mention and entity representations are concatenated and then forwarded to two fully connected layers. The output of the second fully connected layer is a single node, denoting the similarity score after processed by a sigmoid function. Suppose s is the final similarity score and g represents whether the entity is the true entity (ground truth). The training objective is to minimize the following loss value:

$$L(s, g) = g \log(s) + (1 - g) \log(1 - s) \tag{14}$$

e: CANDIDATE ENTITIES RANKING

The candidate entities are not ranked solely based on context similarity. Instead, the final score of each candidate entity is the combination of similarity score and prior probability $p(e|m)$ of an entity e, which denotes the possibility that the entity e is the true one given a specific mention m. The specific values of prior probabilities are derived from frequency dictionary, while entities not in the frequency dictionary are assigned with frequency value of 0. Formally, the ranking score of mention-entity pair (m, e) is:

$$r(m, e) = \theta sim(m, e) + \eta p(e|m) \tag{15}$$

where θ and η are coefficients balancing the weights of similarity and prior probability.

III. EXPERIMENTS AND RESULTS

In this section, we first give an overview of experimental settings, which include how to parse Wikipedia dump to construct local knowledge base, create frequency dictionary, generate inputs for embedding training, the construction of alias dictionary, as well as specific settings in Celan for reproduction. Then the original datasets and revisions, along with methods in comparison, are introduced, followed by results and detailed analysis.

A. EXPERIMENTAL SETTINGS

One of the main drawbacks existing among all previous Chinese entity linking works is the fuzzy description of the local knowledge base construction and specific steps for raw data treatment, rendering it hard to reproduce. Therefore, in this work, we detailed the pre-process and model parameters as follows:

1) LOCAL KNOWLEDGE BASE CONSTRUCTION

The local knowledge base was derived from Chinese Wikipedia dump on 01-Dec-2017. Specifically, we utilized MySQL to store the knowledge due to its popularity and simplicity, as well as limited demand for relation information in our work. Other choices might include Neo4j, MongoDB and other NoSQL databases. The original Wikipedia was in the form of XML, and we utilized mwdumper.jar[8] to convert it to the SQL file. The next step was to create tables in MySQL to store Wikipedia information.[9] Finally the generated SQL file containing Wikipedia information was inserted to the database. However, this database cannot be directly put into use since there was much noise inside. We conducted basic SQL operations to keep the minimum information needed for this task, which consists of entity ID, entity name, entity description and functional pages. Meanwhile, considering that the content of Chinese Wikipedia is mixed with traditional Chinese and simplified Chinese, we adopted simplified Chinese and performed the transformation with hanziconv.[10] In all, there were 2,488,144 items in our local knowledge base, including entities, redirect pages and disambiguation pages.

2) FREQUENCY DICTIONARY

The frequency dictionary was also derived from the Wikipedia dump, and Wikipedia Extractor[11] was leveraged to extract useful information from the XML file. The obtained text was then unified and cleaned with traditional-simplified Chinese transformation and irregular punctuation removal, thereafter generating the raw text for frequency dictionary generation and joint embedding training. With regard to the former task, we obtained all the anchor texts along with their links, and replace the links with corresponding entity names, thus creating the frequency dictionary. Note that the process of constructing alias dictionary is described in Section 2.

3) JOINT EMBEDDING TRAINING

As is detailed in Section 2, the corpus for embedding training is composed of the original form and two expanded forms of each sentence, which should be segmented before forwarded to training. Jieba[12] was used as the Chinese segmentation tool in our work due to it popularity and support for user-defined dictionary. We considered the Wikipedia page (entity) names as entries in user-defined dictionary since they can largely improve the performance of segmentation tool, which is exemplified in below. Note the entity identifiers should also be added to the dictionary to avoid segmentation.

Example 4: Without user-defined dictionary, 伊利集团 (Yili Group) would be segmented as 伊利(Yili) and 集团(Group). However, 伊利集团(Yili Group) is a page name

[8] http://dumps.wikimedia.org/tools/

[9] https://phabricator.wikimedia.org/source/mediawiki/browse/master/maintenance/tables.sql

[10] https://pypi.python.org/pypi/hanziconv/0.2.1

[11] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

[12] https://pypi.python.org/pypi/jieba/

in Wikipedia referring to a company, and splitting it would make the word lose its original meaning. By adding all the entity names to the user-defined dictionary, words like this would be kept in its original form.

After word segmentation and removal of punctuations, the corpus was ready for training. Specifically, we used Gensim for joint entity and word embedding training. The embedding dimension was set to 100, window size was 5, and iteration was set to 3. We finally attained 6,363,417,735 effective items with specific embedding values.

4) SETTINGS FOR NEURAL NETWORK

The specific values of parameters in the network are presented in Table 4. Note that the values are determined based on empirical practises, which might not be optimal and we leave the detailed parameter analysis for future study.

TABLE 4. Neural Network parameter settings.

Parameters	Values
mention context window size c	20
entity description window size p	100
LSTM hidden state size for mention context	288
LSTM hidden state size for entity description	96
output size for first fully connected layer	400
activation function for fully connected layer	$\tanh$
number of epochs for training	30
batch size for training	128
optimizer for training	Adam

We set the window size for left/right mention context and entity description at 20 and 100 respectively. Noteworthy is that this denotes the number of segmented tokens instead of characters in the sentence. If the number of tokens was below the value, zero paddling was utilized. The tokens were transformed into embeddings before being forwarded to the neural network and stayed fixed during the training process.

As far as the training corpus for neural networks is concerned, it is also derived from Wikipedia pages. The anchor texts in Wikipedia pages were considered as mentions, while the entity names corresponding to the links attached to them could be regarded as true entities. We prepared the mention contexts and entity descriptions as described before. Due to limitation of computational resources, we restrained the target entities to the aggregated candidate entities over the following two datasets and collect 100 mention contexts for each entity. Furthermore, negative sampling was utilized to accelerate the training process. Specifically, for each true sample, 5 negative samples were created by replacing the correct entity with another randomly selected entity from the mention's candidate entities. In a nutshell, the network was trained over 456,406 samples.

B. DATASETS

We validate the effectiveness of Celan on the datasets created for Chinese Entity Linking tasks in NLPCC 2013[13] and NLPCC 2014[14]. Both datasets were created over weibo, a short and noisy text source, to increase task difficulty. Noticeably, both datasets annotated the data to its own knowledge base. The local knowledge base utilized in 2013 was part of Baidu Baike and merely took advantage of the information in InfoBoxes. As for the 2014 task, the reference knowledge base included about 400,000 entities based on InfoBoxes from a 2013 dump of Chinese Wikipedia. It has also been emphasized that both knowledge bases contained much noise and might need cleaning and regularization. Considering the incompleteness and noise in those knowledges, we manually annotated them to our local knowledge base derived from Chinese Wikipedia dump on 01-Dec-2017 and made it public. Our constructed local knowledge base could not be publicized due to its huge size but we have elaborated the specific steps as above.

C. RESULTS AND ANALYSES

Similar to most recent studies [3], [10], [16], [17], we did not address NIL situations in this study. In other words, a given mention is guaranteed to have corresponding entities in the local knowledge base. As a consequence, we merely kept the in-KB samples in both datasets and also made the record of modification accessible on-line. Eventually, there were 411 hand labelled mentions in NLPCC 2013 and 264 hand labelled mentions in NLPCC 2014 in need of disambiguation and linking.

Recently, there were some research works devoted to Chinese entity linking and all of them merely published the result on one of the aforementioned datasets. Therefore, we separately selected the systems with best performances, re-implemented their methods, and reported their results for each dataset. Concretely, for NLPCC 2013 dataset, we selected Mao *et al.* 2017 [18] and Xiang *et al.* 2016 [19], which reported the best results (better than the winner of the competition). As to NLPCC 2014 dataset, the competitors are Wanli *et al.* 2015 [20] and Chong *et al.*2016 [21], which also presented the best results. Note that since the source codes of the competitors are not publicized and difficult to reproduce, we directly adopt the results reported on the original papers.

1) EFFECTIVENESS OF CANDIDATE GENERATION STRATEGIES

We first report the results of *Recall* in Table 5 with resepect to different candidate generation strategies. Chances are that in the candidate generation step, the candidate entities generated for some mentions do not contain the true entity, thus leading to wrong linking result in spite of following steps. As a consequence, it is vital to improve the *Recall* index since this value represents the upper bound of final linking accuracy. The definition of *Recall* is detailed in Equation 16.

$$Recall = \frac{\#mentions\ (candidates\ containing\ true\ entity)}{\#all\ mentions} \tag{16}$$

[13] http://tcci.ccf.org.cn/conference/2013/pages/page04_eva.html

[14] http://tcci.ccf.org.cn/conference/2014/pages/page04_eva.html

TABLE 5. ***Recall* with respect to different strategies.**

	Original	+S1	+S1+S2	+S1+S2+S3	+S1+S2+S3+S4
Recall (2013)	0.423	0.569	0.820	0.886	0.966
Recall (2014)	0.367	0.564	0.799	0.879	0.977

In table 5, S is the abbreviation of Strategy while Original denotes directly querying the mention name. Evidently, all strategies contribute to the final recall value. The second strategy, frequency dictionary, largely improves the recall and the increments are 0.251 and 0.235 respectively, which can be attributed to large coverage of Wikipedia anchor texts. The final recall values are close to 1 and merely 14 and 6 mentions in each dataset are not generated with true candidate.

TABLE 6. **Entity linking *Accuracy* of all systems on different datasets. - denotes that the results are not available and the original codes are also unable to reproduce.**

Systems	NLPCC 2013	NLPCC 2014
Celan	**0.925**	**0.875**
Mao et al, 2017	0.905	-
Xiang et al, 2016	0.898	-
Chen et al, 2015	-	0.830
Feng et al, 2016	-	0.837

2) COMPARISONS WITH OTHER SYSTEMS

Then we compare Celan with other linking systems regarding the entity linking performances. Note that the coefficients θ and η for context similarity and prior probability are set at 0.5 and 0.5 respectively. The evaluation metric is *Accuracy*, which is also adopted by recent studies [11], [13], [22]. The definition of *Accuracy* can be found in Equation 17. We display the final results in Table 6.

$$Accuracy = \frac{\#mentions\ linked\ to\ the\ right\ entity}{\#all\ mentions} \quad (17)$$

Our proposed method Celan achieves the best results on both datasets. It outperforms the second runner by 2.3% on NLPCC 2013 and by 3.8% on NLPCC 2014. For the two competitors on NLPCC 2013 dataset, although Xiang *et al.* 2016 claim a better performance compared to the winner of the competition by devising a set of hand-generated features to capture similarities between mention and entity, they still require complex feature engineering and fail to mine the deep semantics of text. Mao *et al.* 2017 take a step further and utilize word vector techniques in their system, whereas the overall framework still lingers on the surface. Similarly, both Chen *et al.* 2015 and Feng *et al.* 2016 design effective features to improve the results on NLPCC 2014 dataset, but still neglect the latent features existing underneath the texts. We are the first to utilize deep neural networks for capturing deep text semantics in Chinese entity linking, and the results validate the effectiveness of this approach.

3) COMPARISONS WITH VARIANTS OF Celan

We further compare Celan with its variants to show the contribution made by each component in the ranking process. **prior** represents ranking candidate entities solely based on prior probability $p(e|m)$. **embedding** denotes calculating the similarity between embeddings for mention name and entity name and ranking entities based on embedding similarity. **prior + embedding** stands for combining prior probability and embedding similarity as ranking indicator. **prior + LSTM + double − attention** represents Celan and **prior + LSTM** is Celan without attention mechanism.

TABLE 7. **Entity linking *Accuracy* of variants of Celan.**

Variants	NLPCC 2013	NLPCC 2014
prior	0.791	0.799
embedding	0.713	0.769
prior + embedding	0.793	0.807
prior + LSTM	0.880	0.848
prior + LSTM + double attention	**0.925**	**0.875**

Table 7 shows the results achieved by different variants. Both **prior** and **embedding** achieves linking accuracy over 0.7, which not only proves the effectiveness of these two features, but also should be attributed to the fact that some mentions only have one candidate entity (which is the right one). Nevertheless, when combining them together, the improvements are quite negligible, with 0.2% and 0.8% respectively when compared with **prior**. Therefore, to minimize hand-designed features, we do not integrate **embedding** in Celan. Adding **LSTM** to **prior** shows evident performance gains, at 8.9% and 4.9% for individual dataset, which demonstrates that neural networks can capture text similarities between mentions and entities, thus improving the overall performance. Introducing **double attention** mechanism to the **prior + LSTM** framework also boosts the performance on both datasets, validating the merits of selecting useful information existing in text sources.

4) ERROR ANALYSIS

We conduct the following error analysis to show cases which Celan is incompetent to deal with and possibly suggest future research directions. Overall speaking, about a third of the erroneous cases is caused by the candidate generation process. In other words, the candidate entities for those mentions do not contain true entity in the first place, which can not be compensated by next steps. The lack of prior probability for candidate entities generated by strategy 3 and strategy 4 is responsible for 30% of the erroneous cases. The prior probability is a significant element of the final ranking score and provided that the true entity is generated by the last two strategies, the lack of prior probability will lower its ranking score, thereby giving rise to the false linking result. The rest false cases can be roughly attributed to the noisy context of weibo, from which the neural networks might fail to extract latent features.

IV. RELATED WORK

Early works on entity linking tend to design a set of useful features to capture similarities between mentions and entities, as well as coherences among entities. We generalize those studies as entity linking via feature engineering, which can be further divided into independent and collective methods.

In the former approach, mentions are disambiguated merely in accordance to the context similarity between mentions and entities, which transforms the problem into candidate entities ranking based on mention-entity similarity so as to obtain the most possible result. The similarity is attained by combining hand-designed features such as name string similarity, prior probability, context bag-of-words similarity and so forth. The candidate entities are ranked either by unsupervised methods [23], which calculate cosine similarities of feature vectors and output the results, or supervised approaches [24], [25], which build classifiers trained on annotated dataset. Despite the fact that methods of this kind can achieve good results, they neglect the coherence among entities, which theoretically can improve overall performances.

As for collective linking methods, the most popular approach is to construct a entity graph to capture relations among mentions and entities and then utilize various algorithms to obtain a sub-graph containing the final results. The intuition behind this line of work is that since mentions in the same document are semantically coherent, in the entity graph, the true entities should have stronger connections and form a dense sub-graph. Based on this, several works [26]–[30] proposed and applied modified graph algorithm on the entity graph to attain final results, which improve the disambiguation accuracy to a certain degree. Overall, the collective linking methods generally perform better than the independent counterparts in the traditional entity linking with complex feature engineering.

Recent years have also witnessed EL tasks in new forms, such as List-only EL task [31], [32], Target Entity Disambiguation problem [33], [34] and Named Entity Disambiguation with Linkless KBs [35]. The disparity mainly lies in the knowledge base side. Most of the solutions still utilize abundant features to achieve state-of-the-art entity linking results.

A. ENTITY LINKING VIA NEURAL NETWORKS

Over recent years, with the emergence of neural networks, many natural language processing (NLP) studies have integrated DNNs in their models to extract latent features and avoid complex feature engineering tasks. He *et al.* [36] are the first to introduce neural networks into entity linking framework by utilizing Denoising Auto-encoder (DA) to model mention and entity representation. Aimed at learning entity and context representation to calculate similarity score, Zwicklbauer *et al.* [7] extend word2vec and doc2vec and train the models on corpus created from Wikipedia and Goggle links. Several works [10], [22] argue that words and entities should be mapped to the same space to enable accurate similarity computation and propose joint word and embedding training.

In addition to extending word2vec in representation learning, convolutional neural network (CNN) [4], [11]–[13], [37], recurrent neural network (RNN) [3], [4] and attention mechanism [3], [5] are also proposed to extract more effective features to model mention/entity representation. Among all works, NeuPL [3] yields the most robust and superior results over multiple datasets. The neural network in NeuPL is an attention-based LSTM, which is simiar to our work. Nevertheless, the main differences consist in two aspects. First, we do not simply forward word embedding to the LSTMs on the mention side. Instead, we concatenate word embedding with mention and entity embedding to capture the interactions among them, which has been proved effective in sentiment classification task [14], [15]. Furthermore, we reckon that the words in entity description does not contribute equally to the final entity representation and introduce attention mechanism on the entity side to filter less valuable information.

B. CHINESE ENTITY LINKING

Compared with continuing advance in English-oriented entity linking task, Chinese entity linking is still in its infancy. This can be mainly ascribed to three aspects: (1) Lack of knowledge bases. As pointed out before, currently there is no Chinese knowledge base satisfying the criteria of being both up-to-date and fully accessible, which restrains the development of Chinese EL. (2) Shortage of evaluation datasets. For the time being, to the best of our knowledge, there are merely four published datasets designed particularly for Chinese EL, including CLP-2012 (not accessible), NLPCC 2013, NLPCC 2014, NLPCC 2015. All of the NLPCC datasets are built upon Weibo, a short and noisy form of text, and NLPCC 2015 requires named entity recognition as an indispensable step. Nevertheless, longer and cleaner text forms such as news, have not been used to construct Chinese EL dataset yet, whereas they are the main forms of datasets in English entity linking. (3) Difficulties posed by Chinese language processing. Text processing, as the basic foundation for entity linking, faces huge challenges exerted by Chinese. For instance, Chinese word segmentation is one of the prominent tasks that still need on-going progress, and it also affects the development of Chinese EL.

Therefore, there are only a few high-quality studies devoted to Chinese EL. Most of them [19]–[21] devise hand-generated features to obtain mention-entity similarity for candidate entities ranking. Liu *et al.* [38] propose graph-based collective entity linking and achieve competitive results. Mao *et al.* 2017 [18] is the first to incorporate embedding techniques in Chinese EL and their system outperforms previous methods. We further bridge the gap between Chinese and English entity linking by proposing Celan, an entity linking framework incorporating improved DNN, to realize robust and effective entity linking process.

V. CONCLUSION

Chinese entity linking suffers from delayed development due to lack of appropriate knowledge base and difficulties posed by Chinese text processing. In this work, we propose Celan, a robust Chinese entity linking system based on LSTM with double attention mechanism, to achieve state-of-the-art entity linking performance by modelling accurate mention and entity representations. Four strategies are introduced to improve recall of candidate entities generation and joint word and entity embedding training is implemented to generate input for neural networks. Moreover, we elaborate the process of constructing a up-to-date local knowledge base derived from Wikipedia dump, and revise the current Chinese EL datasets accordingly. The experimental results on the revised dataset validate the effectiveness of Celan and we believe this work can narrow the gap between Chinese and English entity linking.

For future work, we plan to investigate two aspects. One is to devise a Chinese entity linking dataset derived from news to fill in the blank in Chinese EL. Current datasets are solely based on short context and a new dataset based on long text such as news can further examine the robustness of entity linking systems. Another is to incorporate collective entity linking techniques into Celan to boost its stability over long text.

REFERENCES

[1] X. Chen, Z. Shi, X. Qiu, and X. Huang, "Adversarial multi-criteria learning for chinese word segmentation," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, vol. 1. Vancouver, Canada, Jul./Aug. 2017, pp. 1193–1203. [Online]. Available: https://doi.org/10.18653/v1/P17-1110

[2] B. Xu *et al.*, "CN-DBpedia: A never-ending Chinese knowledge extraction system," in *Proc. 30th Int. Conf. Ind. Eng. Other Appl. Appl. Intell. Syst. (IEA/AIE)*, Arras, France, Jun. 2017, pp. 428–438. [Online]. Available: https://doi.org/10.1007/978-3-319-60045-1_44

[3] M. C. Phan, A. Sun, Y. Tay, J. Han, and C. Li, "NeuPL: Attention-based semantic matching and pair-linking for entity disambiguation," in *Proc. ACM Conf. Inf. Knowl. Manage. (CIKM)*, Singapore, Nov. 2017, pp. 1667–1676. [Online]. Available: http://doi.acm.org/10.1145/3132847.3132963

[4] N. Gupta, S. Singh, and D. Roth, "Entity linking via joint encoding of types, descriptions, and context," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Copenhagen, Denmark, Sep. 2017, pp. 2681–2690. [Online]. Available: https://aclanthology.info/papers/D17-1284/d17-1284

[5] O. Ganea and T. Hofmann, "Deep joint entity disambiguation with local neural attention," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Copenhagen, Denmark, Sep. 2017, pp. 2619–2629. [Online]. Available: https://aclanthology.info/papers/D17-1277/d17-1277

[6] X. Han and J. Zhao, "Named entity disambiguation by leveraging wikipedia semantic knowledge," in *Proc. 18th ACM Conf. Inf. Knowl. Manage. (CIKM)*, Hong Kong, Nov. 2009, pp. 215–224. [Online]. Available: http://doi.acm.org/10.1145/1645953.1645983

[7] S. Zwicklbauer, C. Seifert, and M. Granitzer, "Robust and collective entity disambiguation through semantic embeddings," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Pisa, Italy, Jul. 2016, pp. 425–434. [Online]. Available: http://doi.acm.org/10.1145/2911451.2911535

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2013, pp. 3111–3119.

[9] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, Beijing, China, Jun. 2014, pp. 1188–1196. [Online]. Available: http://jmlr.org/proceedings/papers/v32/le14.html

[10] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji, "Joint learning of the embedding of words and entities for named entity disambiguation," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn. (CoNLL)*, Berlin, Germany, Aug. 2016, pp. 250–259. [Online]. Available: http://aclweb.org/anthology/K/K16/K16-1025.pdf

[11] T. H. Nguyen, N. Fauceglia, M. Rodriguez-Muro, O. Hassanzadeh, A. M. Gliozzo, and M. Sadoghi, "Joint learning of local and global features for entity linking via neural networks," in *Proc. 26th Int. Conf. Comput. Linguist. (COLING)*, Osaka, Japan, Dec. 2016, pp. 2310–2320. [Online]. Available: http://aclweb.org/anthology/C/C16/C16-1218.pdf

[12] Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, and X. Wang, "Modeling mention, context and entity with neural networks for entity disambiguation," in *Proc. 24th Int. Joint Conf. Artif. Intell. (IJCAI)*, Buenos Aires, Argentina, Jul. 2015, pp. 1333–1339. [Online]. Available: http://ijcai.org/Abstract/15/192

[13] M. Francis-Landau, G. Durrett, and D. Klein, "Capturing semantic similarity for entity linking with convolutional neural networks," in *Proc. Conf. North Amer. Chapt. Assoc. Comput. Linguist., Hum. Lang. Technol. (NAACL HLT)*, San Diego CA, USA, Jun. 2016, pp. 1256–1261. [Online]. Available: http://aclweb.org/anthology/N/N16/N16-1150.pdf

[14] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Austin, TX, USA, Nov. 2016, pp. 606–615. [Online]. Available: http://aclweb.org/anthology/D/D16/D16-1058.pdf

[15] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective LSTMs for target-dependent sentiment classification," in *Proc. 26th Int. Conf. Comput. Linguist. (COLING)*, Osaka, Japan, Dec. 2016, pp. 3298–3307. [Online]. Available: http://aclweb.org/anthology/C/C16/C16-1311.pdf

[16] A. Pappu, R. Blanco, Y. Mehdad, A. Stent, and K. Thadani, "Lightweight multilingual entity extraction and linking," in *Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM)*, Cambridge, U.K., Feb. 2017, pp. 365–374. [Online]. Available: http://dl.acm.org/citation.cfm?id=3018724

[17] O.-E. Ganea, M. Ganea, A. Lucchi, C. Eickhoff, and T. Hofmann, "Probabilistic bag-of-hyperlinks model for entity linking," in *Proc. 25th Int. Conf. World Wide Web (WWW)*, Montreal, QC, Canada, Apr. 2016, pp. 927–938. [Online]. Available: http://doi.acm.org/10.1145/2872427.2882988

[18] E. Mao, B. Wang, Y. Tang, and D. Liang, "Entity linking method of chinese micro-blog based on word vector," (in Chinese), *Comput. Appl. Softw.*, vol. 34, no. 4, pp. 11–15, 2017.

[19] Y. Xiang, Y. Guo, X. Xu, W. Zeng, and L. Li, "Entity words disambiguation and entity linking with multi-strategy in chinese microblogs," (in chinese), *Comput. Appl. Softw.*, vol. 33, no. 8, pp. 12–17, 2016.

[20] C. Wanli, Z. Hongying, and W. Yonggang, "Chinese micro-blog named entity linking based on multisource knowledge," (in chinese), *J. Chin. Inf. Process.*, vol. 29, no. 5, pp. 117–124, 2015.

[21] F. Chong, S. Ge, G. Yu-Hang, G. Jing, and H. He-Yan, "An entity linking method for microblog based on semantic categorization by word embeddings," (in Chinese), *Acta Autom. Sinica*, vol. 42, no. 6, pp. 915–922, May 2016.

[22] W. Fang, J. Zhang, D. Wang, Z. Chen, and M. Li, "Entity disambiguation by knowledge and text jointly embedding," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn. (CoNLL)*, Berlin, Germany, Aug. 2016, pp. 260–269. [Online]. Available: http://aclweb.org/anthology/K/K16/K16-1026.pdf

[23] R. C. Bunescu and M. Paşca, "Using encyclopedic knowledge for named entity disambiguation," in *Proc. 11st Conf. Eur. Assoc. Comput. Linguist. (EACL)*, Trento, Italy, Apr. 2006, pp. 1–8. [Online]. Available: http://aclweb.org/anthology/E/E06/E06-1002.pdf

[24] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin, "Entity disambiguation for knowledge base population," in *Proc. 23rd Int. Conf. Comput. Linguist. (COLING)*, Beijing, China, Aug. 2010, pp. 277–285. [Online]. Available: http://aclweb.org/anthology/C10-1032

[25] R. Mihalcea and A. Csomai, "Wikify!: Linking documents to encyclopedic knowledge," in *Proc. 16th ACM Conf. Inf. Knowl. Manage. (CIKM)*, Lisbon, Portugal, Nov. 2007, pp. 233–242. [Online]. Available: http://doi.acm.org/10.1145/1321440.1321475

[26] J. Hoffart *et al.*, "Robust disambiguation of named entities in text," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Jul. 2011, pp. 782–792. [Online]. Available: http://www.aclweb.org/anthology/D11-1072

[27] A. Alhelbawy and R. J. Gaizauskas, "Graph ranking for collective named entity disambiguation," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguist. (ACL)*, vol. 2. Baltimore, MD, USA, Jun. 2014, pp. 75–80. [Online]. Available: http://aclweb.org/anthology/P/P14/P14-2013.pdf

[28] Z. Guo and D. Barbosa, "Robust entity linking via random walks," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, Shanghai, China, Nov. 2014, pp. 499–508. [Online]. Available: http://doi.acm.org/10.1145/2661829.2661887

[29] M. Pershina, Y. He, and R. Grishman, "Personalized page rank for named entity disambiguation," in *Proc. Conf. North Amer. Assoc. Comput. Linguist. (NAACL HLT)*, Denver, CO, USA, May/Jun. 2015, pp. 238–243. [Online]. Available: http://aclweb.org/anthology/N/N15/N15-1026.pdf

[30] X. Han, L. Sun, and J. Zhao, "Collective entity linking in Web text: A graph-based method," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Beijing, China, Jul. 2011, pp. 765–774. [Online]. Available: http://doi.acm.org/10.1145/2009916.2010019

[31] Y. Lin, C. Lin, and H. Ji, "List-only entity linking," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist. (ACL)*, vol. 2. Vancouver, BC, Canada, Jul./Aug. 2017, pp. 536–541. [Online]. Available: https://doi.org/10.18653/v1/P17-2085

[32] W. Zeng, X. Zhao, J. Tang, and H. Shang, "Collective list-only entity linking: A graph-based approach," *IEEE Access*, vol. 6, pp. 16035–16045, 2018.

[33] C. Wang, K. Chakrabarti, T. Cheng, and S. Chaudhuri, "Targeted disambiguation of ad-hoc, homogeneous sets of named entities," in *Proc. 21st World Wide Web Conf. (WWW)*, Lyon, France, Apr. 2012, pp. 719–728. [Online]. Available: http://doi.acm.org/10.1145/2187836.2187934

[34] Y. Cao, J. Li, X. Guo, S. Bai, H. Ji, and J. Tang, "Name list only? Target entity disambiguation in short texts," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Lisbon, Portugal, Sep. 2015, pp. 654–664. [Online]. Available: http://aclweb.org/anthology/D/D15/D15-1077.pdf

[35] Y. Li, S. Tan, H. Sun, J. Han, D. Roth, and X. Yan, "Entity disambiguation with linkless knowledge bases," in *Proc. 25th Int. Conf. World Wide Web (WWW)*, Montreal, QC, Canada, Apr. 2016, pp. 1261–1270. [Online]. Available: http://doi.acm.org/10.1145/2872427.2883068

[36] Z. He, S. Liu, M. Li, M. Zhou, L. Zhang, and H. Wang, "Learning entity representation for entity disambiguation," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguist. (ACL)*, vol. 2. Sofia, Bulgaria, Aug. 2013, pp. 30–34. [Online]. Available: http://aclweb.org/anthology/P/P13/P13-2006.pdf

[37] W. Xu and J. Yu, "A novel approach to information fusion in multi-source datasets: A granular computing viewpoint," *Inf. Sci.*, vol. 378, pp. 410–423, Feb. 2017. [Online]. Available: https://doi.org/10.1016/j.ins.2016.04.009

[38] Q. Liu, Y. Zhong, Y. Li, Y. Liu, and Z. Qin, "Graph-based collective Chinese entity linking algorithm," *J. Comput. Res. Develop.*, vol. 53, no. 2, pp. 270–283, 2016.

WEIXIN ZENG received the bachelor's degree from the National University of Defense Technology, China, in 2017, where he is currently pursuing the master's degree. His research interests include knowledge graph and entity linking.

JIUYANG TANG received the Ph.D. degree from the National University of Defense Technology (NUDT), China, in 2006. He is currently a Professor with NUDT. His research interests include knowledge graph and text analytics.

XIANG ZHAO was born in 1986. He received the Ph.D. degree from the University of New South Wales, Australia, in 2014. He is currently an Assistant Professor with the National University of Defense Technology, China. He has published several refereed papers in international conferences and journals, such as SIGMOD, SIGIR, VLDB Journal, and TKDE. His research interests include graph data management and knowledge graph construction.