information

MDPI

Article

Attention-Based Joint Entity Linking with Entity Embedding

Chen Liu [1,2,3], Feng Li [1,2,*], Xian Sun [1,2,3] and Hongzhe Han [2,4]

1 Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China; 15611529726@163.com (C.L.); sunxian@mail.ie.ac.cn (X.S.)
2 Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China; hanhz@mail.ie.ac.cn
3 School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100190, China
4 Cloud Computing Center, Chinese Academy of Sciences, Dongguan 523808, China
* Correspondence: lifeng@mail.ie.ac.cn; Tel.: +86-10-5888-7208

Received: 4 December 2018; Accepted: 28 January 2019; Published: 1 February 2019

Abstract: Entity linking (also called entity disambiguation) aims to map the mentions in a given document to their corresponding entities in a target knowledge base. In order to build a high-quality entity linking system, efforts are made in three parts: Encoding of the entity, encoding of the mention context, and modeling the coherence among mentions. For the encoding of entity, we use long short term memory (LSTM) and a convolutional neural network (CNN) to encode the entity context and entity description, respectively. Then, we design a function to combine all the different entity information aspects, in order to generate unified, dense entity embeddings. For the encoding of mention context, unlike standard attention mechanisms which can only capture important individual words, we introduce a novel, attention mechanism-based LSTM model, which can effectively capture the important text spans around a given mention with a conditional random field (CRF) layer. In addition, we take the coherence among mentions into consideration with a Forward-Backward Algorithm, which is less time-consuming than previous methods. Our experimental results show that our model obtains a competitive, or even better, performance than state-of-the-art models across different datasets.

Keywords: entity linking; LSTM; CNN; CRF; Forward-Backward Algorithm

1. Introduction

Given a query consisting of mentions (name strings) and their background document, entity linking involves linking the mentions to their corresponding entities from a reference knowledge base, such as Wikipedia [1]. It is a fundamental task in the field of natural language processing, which can facilitate many other tasks, such as semantic search, question answering, relation extraction, and text understanding [2].

The challenge of entity linking comes from polysemous mentions and various forms of entities. For example: The mention "**Jordan**" could refer to the basketball player "**Michael Jordan**" or the actor "**Michael B Jordan**", and the entity "**Michael Jordan**" could be named as "**His Airness**" or "**MJ**". All the existing state-of-the-art methods regard entity linking as a ranking problem. They try to find the most semantically matched entity for every given mention as its predicted entity [1,3–5]. The key difference among these methods is how to encode the entity and mention contexts.

In a knowledge base (e.g., Wikipedia), one can obtain certain kinds of information about an entity, such as the entity context and entity description, as shown in Figure 1. Generally, these information

reflect different aspects of entity, and our experimental results show that the entity linking system could obtain better performance with an entity embedding framework which can capture a larger range of different entity information aspects.

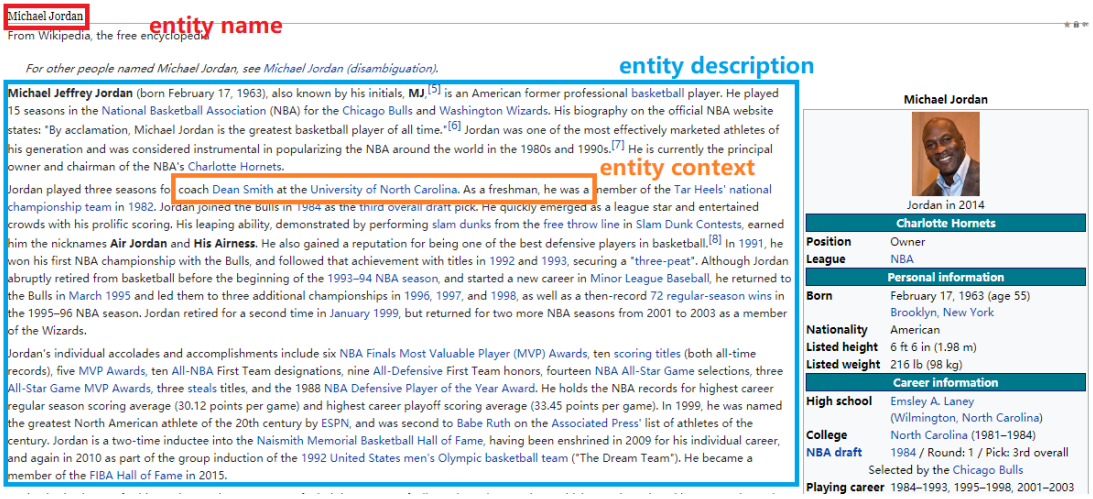

Figure 1. In Wikipedia, each entity has a canonical page and this is the canonical page for the entity "**Michael Jordan**". The description for the entity "**Michael Jordan**" consists of the words in the blue box. There are many hyperlinks in the page and the words around the hyperlink form the entity context for the corresponding entity. For example, the context for the entity "**North Carolina Tar Heels men's basketball**" consists of the words in the orange box.

Some earlier methods [2,6] only take the entity description into consideration, with heuristic methods like BOW or TF-IDF. However, Sun [1] argues that these methods are insufficient to disentangle the underlying explanatory factors of the data and proposes a method which employs a convolutional neural network (CNN) to encode the entity description. Some other methods [3,7–9] try to encode the entity, based on the idea of word embedding, which only takes entity context into consideration. However, all the methods above fail to capture different information aspects of entity, which could result in a loss of information. Inspired by Nitish Gupta's work [4], we design an entity embedding framework which can capture different information aspects of entity. Other entity information aspects may be lacking, like entity type, while the entity description and entity context are common for most knowledge bases. So, in our work, we mainly investigate how to effectively encode and combine entity description and entity context, to generate dense unified entity embeddings. It is also worth noting that our entity embedding framework could be easily extended to the case in which there are more entity aspects. We use long short term memory (LSTM) [10] and CNN [11] to encode the entity context and entity description, respectively, and afterwards we design a function to encourage the entity embedding's similarity to all of the encoded representations (e.g., representation of entity context and representation of entity description). Compared to previous methods, our approach can capture different aspects of entity information, and our experimental results show that our global model could obtain a better performance, with entity embeddings which can capture richer entity information aspects.

The semantics of mention mainly come from the mention context and other mentions in the given document. For mention context, most previous methods assume that all the words in the context have the same importance [1,4,6,8,12]. Obviously, this should be investigated carefully. Some works [8,13] try to introduce the standard attention mechanism to fix this problem. However, the standard attention mechanism can be viewed as a process of performing soft selections of individual words independently; it ignores the dependencies between words, and may make mistakes when complex expressions are involved. Inspired by Wang's work [14], we propose a novel conditional random field (CRF)-based attention mechanism, which can effectively capture the important text spans for the mention with a

CRF layer. The coherence among mentions could also be helpful in entity linking. For example, when the mention "**Jordan**" and the mention "**Bulls**" both appear in a document. We can infer that the mention "**Jordan**" refers to the entity "**Michael Jordan**" and the mention "**Bulls**" refers to the entity "**Chicago Bulls**" with a high probability. In this paper, we use the Forward-Backward Algorithm to calculate the coherence among mentions, which is less time-consuming than the graph-based methods and probabilistic methods used in earlier entity linking systems [2,3].

The main contributions of this work are the following:

1. We present an entity embedding framework, which can effectively capture different information aspects.
2. We are the first ones who use a CRF-based attention mechanism to capture the important text spans in the mention context, to improve the performance of our linking system.
3. We take the coherence among mentions into consideration with the Forward-Backward algorithm, which is less time consuming than those graph-based models used in previous work.
4. Based on the above three contributions, we build our global model. Our experimental results show that our model can achieve a competitive, or better, performance than state-of-the-art models.

2. Related Work

There have been a lot of studies on entity linking but, in this section, we only focus on those prior works which relate to our main contributions.

2.1. Encoding of Entity

Similar to the idea of word embedding [15], the main goal of entity embedding is to compress the relevant information of an entity into a vector. In entity linking systems without entity embedding, such as the method proposed in [2], the coherence between two entities is often calculated by counting the number of the page they share, which is called a co-occurrence entity pair. However, this method can suffer from sparsity issues and large memory footprints. In addition, there are often some new entities, which need to be added into a knowledge base in practice, while all the co-occurrence entity pairs need to be recounted under this framework when new entities are added. With the help of entity embedding, new entities could be added in an incremental manner, just as with word embedding [15], which is necessary in practice. To the best of our knowledge, Stefan Zwicklbauer [3] was the first one to extend word embedding to entity embedding. Later, some works [8,13] followed Stefan Zwicklbauer's work. The framework proposed by Zwicklbauer only takes entity context into consideration. It ignores the fact that the entity description is also an important semantic aspect, which may result in loss of information. These days, some works [1,16] have been carried out to address the problem. However, they fail to obtain unified entity embeddings, like in the framework proposed by Matthew [16], which uses the CNN to encode the entity title and entity description, respectively, but both of the encoded representations are independent when linking mentions to their corresponding entities. What is special, in our model, is that we design a function to encourage the entity embedding to be similar to all the encoded representations. Our model can capture different information aspects about entities and generate unified dense entity embeddings. More importantly, entities could be added in an incremental way in our method, which was often ignored in previous methods.

2.2. Encoding of Mention Context

There have been many approaches to encoding the mention context in the literature. Z. Chen et al. [6,12] used some heuristic methods, such as bag of words (BOW) or TF-IDF , to encode mention context. However, those heuristic methods capture the semantics of mention context in a coarser way than deep learning methods. Yaming Sun et al. [1] used CNN and stack denoising auto-encoders to encode the mention context. Some other deep learning models were also used in entity linking to encode the mention context, such as LSTM and Doc2vec [15]. But, all the methods above made the same assumption: That all the words in the context have the same

importance, which should be investigated carefully. To fix this problem, some attention-based methods were proposed, Octavian et al. [8,13,17] proposed a standard attention mechanism, based on LSTM, which tried to capture the individual important words. However, the standard attention mechanism was usually achieved by concatenating vectors and sending them to a multilayer perceptron (MLP), and could be viewed as a process of performing soft selections of individual words independently. It ignored the dependencies between words. Our model uses CRF to capture the dependencies and find the important text spans, rather than individual words.

2.3. Modeling Coherence among Mentions

As we mentioned earlier, the coherence among mentions could also be helpful in entity linking, especially when we can not disambiguate the mention only by its context. Han et al. [2,8,18] tried to use a graph model to calculate the coherence among metions to improve the performance of entity linking. Some probabilistic models were also used in entity linking, such as random walk [19]. However, some researchers [13,20] argued later that all those graph and probabilistic models are time-consuming. So, simplifying the methods of calculating the coherence among metions has become a research hotspot for entity linking. Some researchers have tried to use heuristic algorithms to fix the problem, such as the Hill-climbing Algorithm [20]. More recently, Phan et al. [13] indicated that assuming that all the mentions in the same context should have similar topics is not suitable. Typically, only a few of the mentions in the same context have a high coherence. Using this assumption disambiguates the two mentions which have the highest confidence in an iterative way. Although the method simplifies the calculation a lot, it is not robust enough. In our model, we use the Forward-Backward Algorithm to calculate the topic information, which is robust and less time-consuming.

3. Definition

Formally, given a document D containing a set of words $D = \{w_1, ..., w_i, ...w_t\}$ (where w_i denotes the i-th word in the document), a set of mentions $M = \{m_1, m_2, ..., m_n\}$ (where m_i denotes the i-th mention in the document), and each mention has context $C = \{w_{m-s}..., w_{m-1}, w_m, w_m, w_{m+1}, ...w_{m+s}\}$ (where w_m denotes a word which is a part of a mention and s is the window size of the mention context), if K is a a target knowledge base, then entity linking is a task to find a N-tuple $u = \{e_1, e_2, ..., e_n\}$, $e_i \in C_{m_i} \subseteq K$ (where C_{m_i} is the candidate entity set for m_i and e_i is a candidate entity of C_{m_i}). The entity candidate set is often used to improve the accuracy and efficiency of the entity linking algorithm (see more details in Section 5.2). All the early methods can be formulated by Equation (1).

$$u^* = \arg\max_{u}^{*}(\sum_{i=1}^{n} \varphi(m_i, e_i) + \psi(u)) \quad (1)$$

Here, $\varphi(m_i, e_i)$, called the local score, reflects the likelihood mapping $m_i \rightarrow e_i$, based on the semantic similarity between the mention context and entity which can be computed independently. The model which only takes the local score $\varphi(m_i, e_i)$ into consideration is called the local model. $\psi(u)$ is called the co-occurrence score, and reflects the coherence among the mentions which was often calculated by graph or probabilistic methods in the previous works. The model which makes use of both $\varphi(m_i, e_i)$ and $\psi(u)$ is called the global model. Most of the state-of-the-art models try to find the N-tuple u^* by maximizing the confidence of each assignment $\varphi(m_i, e_i)$, which can be determined by the cosine similarity between the entity embedding of the entity e_i and the mention context representation of the mention m_i, while enforcing the coherence among all the linked entities $\psi(u)$. The coherence between two entities are often calculated by the cosine similarity between embeddings of the two entities. The key difference between these models is how to encode entity and mention context. In our paper, we mainly focus on the representations of entities and mentions.

4. Model

In this section, we first describe how to build the framework of entity embedding (Section 4.1). Then, we show the details of the mention context encoder and our local model (Section 4.2). Last, we address the global model, assuming the coherence among the mentions (Section 4.3). We show the relations among the three parts in Figure 2.

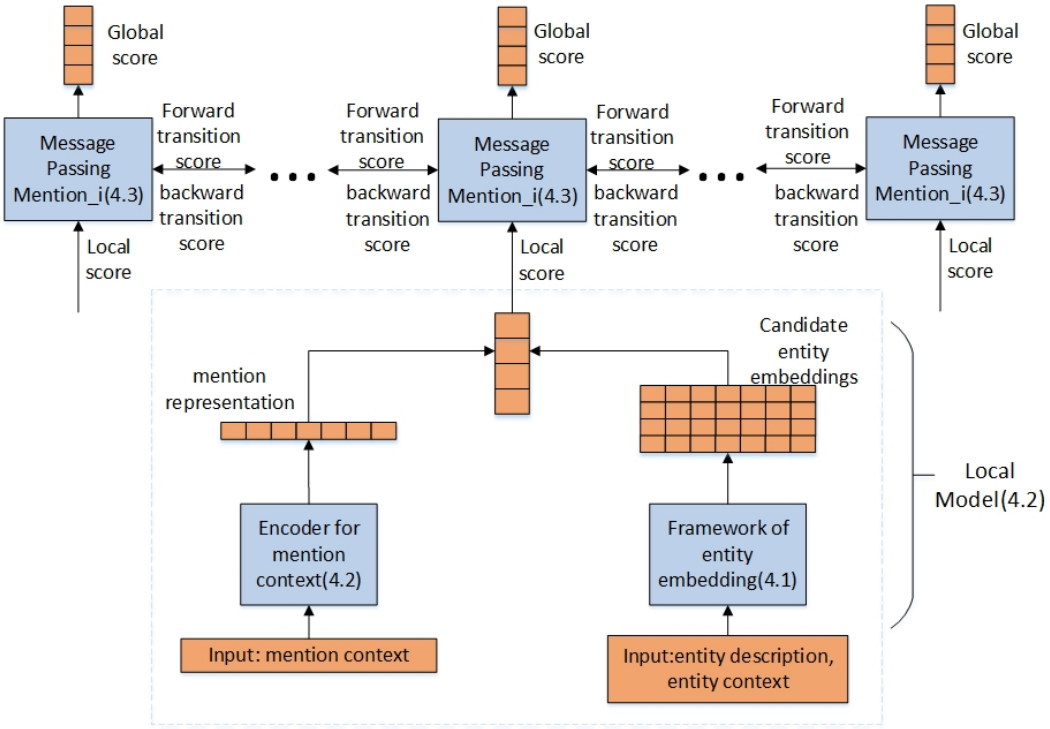

Figure 2. Overview of the model. The input of the encoder for the mention context is the mention context and the inputs for entity embedding are the entity description and the entity context. Firstly, we get the representations of entity and mention context, through the entity embedding framework (Section 4.1) and mention context encoder (Section 4.2), respectively. Then, we get the local scores by computing the cosine similarity between the two representations (Section 4.2). Lastly, we take the coherence among mentions into consideration with a Forward-Backward Algorithm, and gain the global scores based on local scores (Section 4.3).

4.1. Framework of Entity Embedding

The semantics of entity may come from many different aspects such as entity description and entity context. The key to an accurate entity embedding framework is the effective encoding and combination of different information aspects. Inspired by the work of Gupta [4], we use LSTM and CNN to encode the entity context and entity description, respectively. Then, we combine all the different information aspects to get unified dense entity embeddings, based on the assumption that an entity embedding should be similar to all its related encoded representations (e.g., the representations of entity context and of entity description).

4.1.1. Encoder for Entity Context

As Figure 1 shows, the entity contexts are usually short and their word order can not be ignored. LSTM is a model which could effectively capture the word order so, in this case, we employ LSTM as our basic model. The processing is shown in Figure 3.

In more detail, given a hyperlink and its context $C = \{w_1, ..., w_m, ...w_{2s}\}$, we split the context into two parts by the anchor text w_m, $left = \{w_1, w_2, ...w_m\}$, $right = \{w_{2s}, w_{2s-1}, ..., w_m\}$. The left-LSTM is applied to the sequence $left = \{w_1, w_2, ...w_m\}$ with the output h^l, while the right-LSTM is applied to

the sequence $right = \{w_{2s}, w_{2s-1}, ..., w_m\}$ to produce h^r. After that, we concatenate the vector h^l and the vector h^r, and send it to a MLP layer to get the encoded entity context representation v_c. Inspired by the work proposed by Yamada [9], we design a function that enables entity embeddings to contain the information of the entity context:

$$P_{text}(e|v_c) = \frac{\exp(v_c \cdot v_e)}{\sum_{c_k \in C_m} \exp(v_c \cdot v_{c_k})}. \quad (2)$$

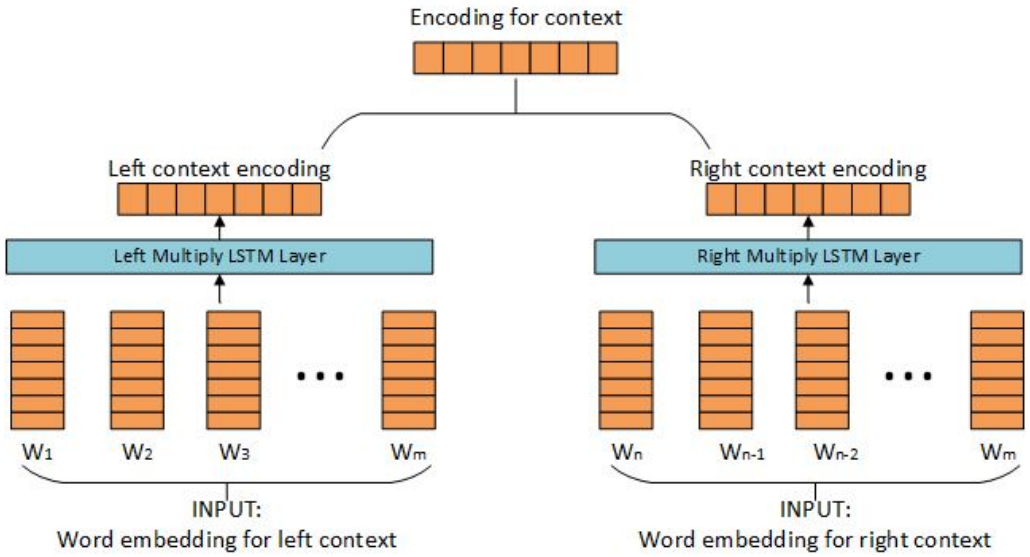

Figure 3. Encoding the entity context information with LSTM. The input of the model is word embeddings of the words in the context, which may come from an annotated corpus (e.g., Wikipedia hyperlinks in our case).

In this equation, e is the corresponding entity to the current mention and v_e is the entity embedding of the entity e. The entity embeddings v_e are randomly initialized and will converge through the course of training. C_m is the entity candidate set of the current mention, and v_{c_k} is the entity embedding of a candidate entity. Here, based on the assumption that the context-encoded representation should be similar to its corresponding entity embedding and dissimilar to other entity embeddings, we maximize the cosine similarity between v_c and v_e, and minimize the cosine similarity between v_c and the embeddings of the other entities in C_m.

4.1.2. Encoder for Entity Description

As Figure 1 shows, the description is usually long and noisy. In our model, we use CNN to encode the entity description, which has proved efficient in handling the text-like entity description in Yoon Kim's work [21]. The framework is shown in Figure 4.

As Figure 4 shows, the input of the model is the word embeddings of the description. We feed it into the CNN and get the encoded description representation v_{desc}. Similar to Equation (3), we encourage v_{desc} to be similar to its corresponding entity embedding v_e and dissimilar to other candidate entity embeddings, and get the expression $p_{desc}(e|v_{desc})$.

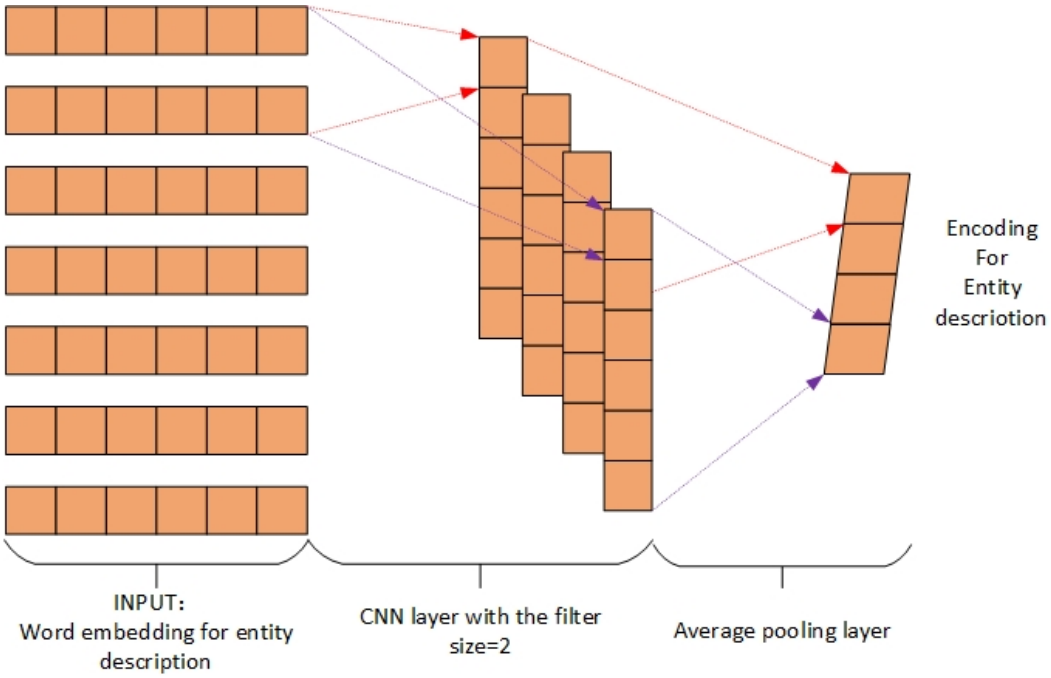

Figure 4. Encoding the description information with CNN. The input of this framework is the word embedding of the entity description.

4.1.3. Combine Different Information Aspects

We design a function to encourage entity embeddings to be similar to all the encoded representation and get a unified dense entity embedding.

$$\{v_e\}, \theta = \arg\max_{v_e,\theta}(P_{text}(e|v_c) + p_{desc}(e|v_{desc})). \tag{3}$$

Here, e is an entity of the target knowledge base K and $\{v_e\}$ are the trained entity embeddings for all the entities in the knowledge base. θ are the training parameters of the model. v_c is the encoded representation of entity context and v_{desc} is the encoded description representation of the entity e. $P_{text}(e|v_c)$ denotes the similarity between the current entity embedding and its context encoded representation (see Section 4.1.1). $p_{desc}(e|v_{desc})$ denotes the similarity between the current entity embedding and its encoded description representation (see Section 4.1.2).

4.2. Attention-Based Local Model

We build our attention-based local model on the insight that only a few text spans are important for disambiguation. Focusing on those text spans can help to reduce noise and improve the performance of entity linking. This model tries to stimulate the processing of human beings by using a CRF layer. Given a mention and its context, people try to find the relevant text spans around the mention to infer its corresponding entity. The model is shown in Figure 5. Furthermore, we design two additional regularizations to guide this model.

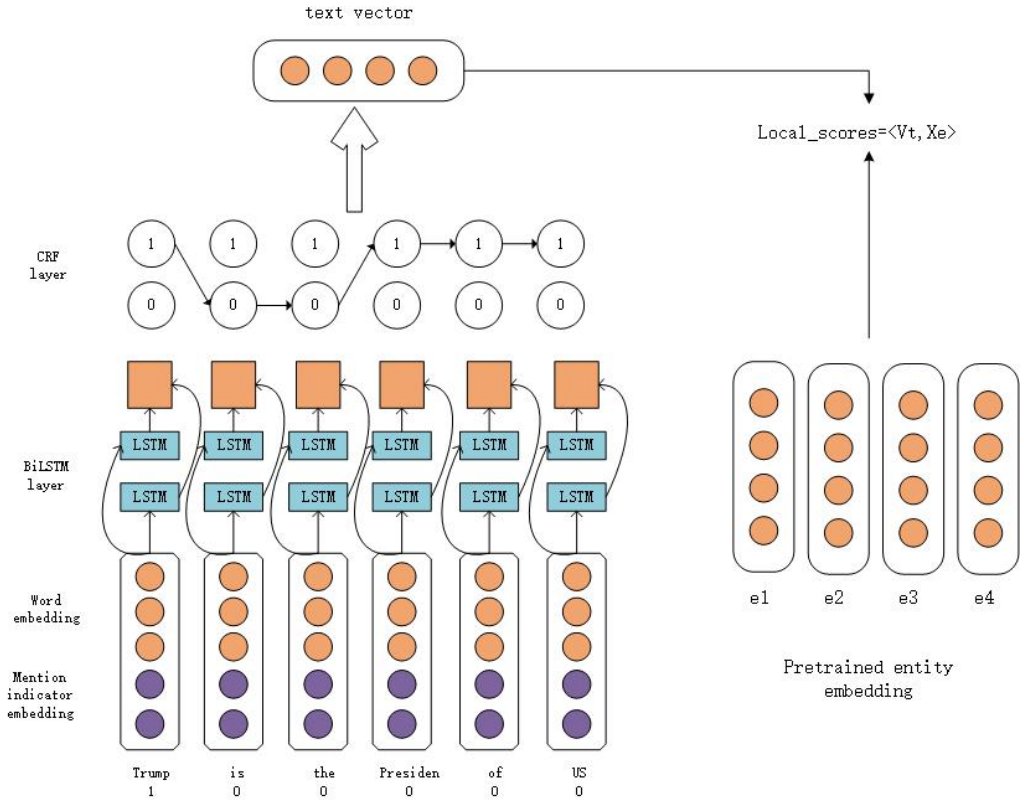

Figure 5. Local model with a novel CRF-based attention mechanism. The inputs of this model are the mention m, its context C, and the candidate entity embeddings for m. It should be noted that all the entity embeddings for the knowledge base K need to be pretrained in Section 4.1. The coherence among mentions is ignored in this section, and mentions are disambiguated independently.

In this section, we disambiguate the mention m by its context $C = \{w_1, w_2, ..., w_{2s}\}$. Here, s is the window size of the mention context. Sometimes, once we know the mention, we can nearly infer its corresponding entity. To highlight the current mention words, we introduce a binary variable $i_t \in \{0, 1\}$, which is called the mention indicator, to indicate whether the i-th is a part of the current mention or not. As Equation (4) shows, the input V of the model is the concatenation of the word embedding and mention indicator embedding.

$$V = [v_1, v_2, ..., v_{2s}]. \tag{4}$$

$$v_t = [W_{emb}(w_t), W_{mask}(i_t), t \in [1, 2s]], W_{emb} \in R^{d \times d_1}, W_{mask} \in R^{2 \times d_2}. \tag{5}$$

Here, d is the size of the word dictionary, d_1 denotes the dimension of word embedding, $W_{emb}(w_t)$ denotes the mapping of a word onto a vector, d_2 denotes the dimension of the mention indicator embedding, and $W_{mask}(i_t)$ denotes the mapping of a mention indicator onto a vector. V is sent to a Bi-LSTM layer. For each unit of BiLSTM, we get two hidden vectors: The forward hidden vector $\overrightarrow{h_t}$ for forward transmission of information, and the backward hidden vector $\overleftarrow{h_t}$ for backward transmission of information. We concatenate the two vectors $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ as the final output of each unit of BiLSTM layer, shown in Equation (6)

$$R = [r_1, r_2, ..., r_{2s}]. \tag{6}$$

$$r_t = [\overrightarrow{h_t}, \overleftarrow{h_t}], t \in [1, 2s]. \tag{7}$$

As Figure 5 shows, for every word in the given context, we introduce a binary variable $z \in \{0, 1\}$ to indicate whether the word is important or not. The importance state changes, along with the mention context. For a certain mention context, we could use a path $P = \{z_1, z_2....z_{2s}\}$ to denote a

possible importance state changing, where z_i is the importance state corresponding to the i-th word in the mention context. The space of the possible paths P grows exponentially, corresponding to the number of given words. Formally, the importance distribution of the given mention context is calculated by Equation (8)

$$Pr(P|R) = \frac{1}{Z(R)} \prod_{z_i \in P} f(z_i, R), \tag{8}$$

$$Z(R) = \sum_{P'} \prod_{z_{i'} \in P'} f(z_{i'}, R), \tag{9}$$

where Pr is formed by z_i, $f(z_i, R)$ is the potential function of the path P, and $Z(R)$ is a normalization factor calculated by summing the potential functions of all the possible paths. In detail, $f(z_i, R)$ is formed by two potentials on the vertices of Equations (11) and (12) for an undirected graphical model, respectively.

$$\prod_{z_i \in P} f(z_i|R) = \prod_{i=1}^{2s} f_1(z_i|R) \prod_{i=1}^{2s-1} f_2(z_i, z_{i+1}|R). \tag{10}$$

$$f_1(z_i|R) = \exp(W^v_{z_i} \cdot r_i + b). \tag{11}$$

$$f_2(z_i, z_{i+1}|R) = \exp(W^e_{z_i, z_(i+1)}). \tag{12}$$

We know from Equations (10)–(12) that the importance of the current word is dependent on the current word itself and its adjacent words. Here, $W^v \in R^{2 \times 2d_h}$ is the state characteristic matrix, which maps context representation to the feature score of the importance state, and d_h is the dimension of r_t. Further, $W^e \in R^{2 \times 2}$ is a transition matrix which denotes the transmission between every adjacent word's importance state. After we get the importance distribution of every word in the context, we get the mention context encoded representation v_{text} by weighted summarization, which is shown in Equation (13)

$$v_{text} = \sum_z p(z) g(R, z), \tag{13}$$

$$g(R, z) = \sum_i 1(z_i = 1) \cdot r_i, \tag{14}$$

where $g(R, z)$ is a feature function, which is defined based on the selection of path P. However, enumerating all the possible paths would be very time-consuming and so, in this paper, we simplify the calculation by dynamic programming with a message passing model and Equation (13) can be rewritten as Equation (15)

$$v_{text} = \sum_{i=1}^{2s} (p(z_i = 1) r_i). \tag{15}$$

As mentioned earlier, entity linking can be regarded as a ranking problem. The entity which has the most similar semantics to the given mention will be picked as the mention's predicted entity. Here, we maximize the local score of the corresponding entity of the current mention and minimize the local score of other entities in the candidate set of the mention. The local score is formulated in Equation (16), and the optimal object of the local model is shown in Equation (17)

$$\varphi(m, e) = \exp(v_{text} \cdot v_e). \tag{16}$$

$$opt(m, e) = \frac{\varphi(m, e)}{\sum_{c_k \in C_m} \varphi(m, c_k)}. \tag{17}$$

The key difference between our attention mechanism and the standard attention mechanism is that our model can capture important text spans, rather than individual words in the given context; which is more similar to human thought. Inspired by the framework proposed by Bailin Wang [14],

we introduce two regularizations to improve the performance of our model, based on the assumptions that only a few text spans are important for entity linking:

$$\Omega_1(z) = \sum_i \sum_{i \neq j} \max(0, W^e_{ij} - W^e_{ii}), \text{ and} \tag{18}$$

$$\Omega_2(z) = \sum_i^n p(z_i = 1). \tag{19}$$

Ω_1 is designed to discourage the importance state change between every two adjacent words, which enforces our model to focus on the text spans rather than individual words. Ω_2 is designed to discourage the number of important spans, which also enforces our model to focus on the text span, which is really important for entity linking.

4.3. Global Model

The assumption that all the mentions which appear in the same context share a similar topic is often used to improve the performance of entity linking. This is unlike local models, which ignore the coherence among mentions and disambiguate all the given mentions independently. In this section, we address the coherence among mentions with a Forward-Backward Algorithm and disambiguate all the mentions $M = \{m_1, m_2, ..., m_n\}$, based on our local model at a given time. The document $D = \{w_1, ..., w_i, ...w_t\}$ and all the mentions in the document $M = \{m_1, m_2, ..., m_n\}$ need to be provided to our global model. The structure of our proposed global model is shown in Figure 6.

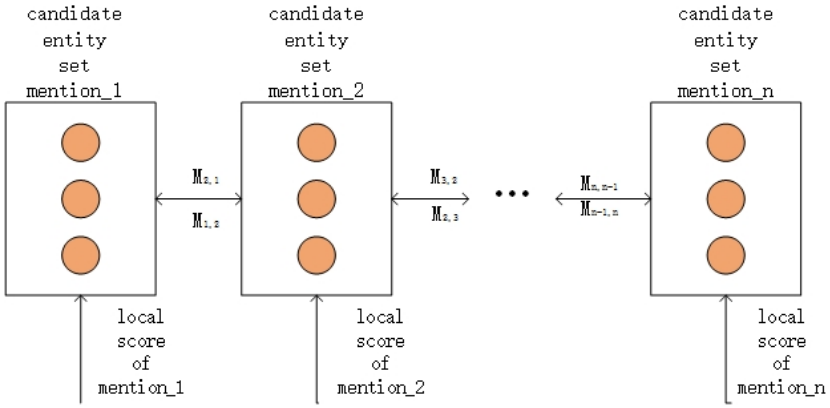

Figure 6. The framework of our global model, where each circle denotes a candidate entity. The global score is defined by transmission score and local score. Here, the local score can be computed as in Section 4.2, and we employ a Forward-Backward Algorithm to compute the transmission score which denotes the coherence among mentions.

Our global model is defined by Equation (20)

$$score(e|m_i) = \varphi(m_i, e) + M_{i-1,i}(e) + M_{i+1,i}(e). \tag{20}$$

To disambiguate all the given mentions, we need find the most semantic matched entity for each mention, while all the chosen entities should have a high coherence. Here, $\varphi(m_i, e)$ denotes the semantic matching rate between the entity e and the i-th mention m_i, which can be calculated by our local model (mentioned in Section 4.2). $M_{i-1,i}(e)$, $M_{i+1,i}(e)$ denotes the coherence between a candidate entity e of m_i and its adjacent entities. We calculate the coherence using dynamic programming with a Forward-Backward Algorithm. More specifically, $M_{i-1,i}(e)$ denotes the forward message passing and $M_{i+1,i}(e)$ denotes the backward message passing. The only difference between $M_{i-1,i}(e)$ and $M_{i+1,i}(e)$

is the direction of message transmission. Here, we only give the details of $M_{i-1,i}(e)$, which can be formulated by Equation (21). We can calculate $M_{i+1,i}(e)$ in a similar way.

$$M_{i-1,i}(e) = \max_{e' \in C_{m_{i-1}}} (\varphi(m_{i-1}, e') + \phi(e', e)), \tag{21}$$

where $M_{i-1,i}(e)$ denotes the message passing from the $i-1$th mention to the entity e of the ith mention, and $e' \in C_{m_{i-1}}$ denotes that e' is a candidate entity for m_{i-1}. Here, $\phi(m_{i-1}, e')$ denotes the semantic matching rate between e' and the context of m_{i-1}. $\phi(e', e)$ denotes the coherence between two entities, which can be formulated by Equation (22)

$$\phi(e', e) = v_{e'} \cdot A \cdot v_e, \tag{22}$$

where $A \in R^{d_e \times d_e}$ is randomly initialized and needed to be trained, v_e and $v_{e'}$ are the corresponding entity embeddings for the entities e and e', and d_e is the dimension of the entity embedding. Following the idea of the local model, the optimal function of our global model is designed as in Equation (23). We encourage our ground-truth entity to have a higher global score than the other candidate entities.

$$opt(e|m_i) = \frac{\exp(score(e|m_i))}{\sum_{c_k \in C_m} \exp(score(e_{c_k}|m_i))}. \tag{23}$$

Our final loss function, for the whole model, is shown in Equation (24)

$$L = -\sum_{i=1}^{n} opt(e|m_i) + \lambda_1 \Omega_1(z) + \lambda_2 \Omega_2(z), \tag{24}$$

where λ_1, λ_2 denote the coefficient of each of the regularizers, and $\Omega_1(z)$, $\Omega_2(z)$ are as mentioned in Section 4.2.

5. Experiments

In this section, we first give the benchmark datasets that we used. Next, we introduce how to obtain the candidate sets for each mention, how to train the entity embedding, and how to train our local model and global model, based on pre-trained entity embedding and word embedding. Last, we show the performance of our entity linking system and the state-of-the-art models that we compared our model to.

5.1. Datasets

Here, we give the details of the datasets involved in our experiments: AIDA-CoNLL [22] is one of the biggest manually-annotated datasets for entity linking. It consists of three parts: AIDA-train (a training set), AIDA-A (a validation set), and AIDA-B (a test set). WNED-CWEB are automatically extracted from clue-web and cleaned by Barbosa [23]. ACE04 and AQUAINT are both cleaned and updated by Guo and Barbosa [23,24]. The statistics of all the datasets are shown in Table 1.

Table 1. Statistics of all the related datasets.

Datasets	Number Docs	Number Mentions
AIDA-train	946	18,448
AIDA-A (valid)	216	4971
AIDA-B (test)	231	4485
WNED-CWEB	320	11,154
ACE04	36	257
AQUAINT	50	727

5.2. Candidate Generation

To disambiguate the mentions provided, computing the semantic similarity between the current mention and all the entities in the knowledge base is impractical. Generating an entity candidate set for each mention is a very important step for both efficiency and accuracy of entity linking. We pick the top 30 popular entities for each mention as its entity candidate set. The most common way of estimating popularity is by using the Wikipedia-based frequencies of particular names in link anchor texts referring to specific entities. In this paper, we build our hyper-link statistics from Wikipedia (February 2014) and a large web corpus [25]. The statistics for each of the datasets is shown in Table 2.

Table 2. Statistics for candidate generation. Gold recall denotes the ratio of the mentions for which the candidate entity set includes the ground truth entity.

Datasets	Number Linkable Mentions	Gold Recall
AIDA-train	18,143	0.98
AIDA-A (valid)	4665	0.97
AIDA-B (test)	4359	0.97
WNED-CWEB	233	0.90
ACE04	10,983	0.93
AQUAINT	694	0.96

The gold recall was calculated by $\frac{\sum_i^n e_{m_i} \in C_{m_i}}{n}$, where e_{m_i} is the ground truth entity of m_i, C_{m_i} is the candidate entity set for m_i, and n is the size of the dataset. From Table 2, we can see that dividing the candidate set into a small number did unnecessary harm to the recall. Our model can obtain a better efficiency in the case of less candidate entities.

5.3. Disambiguation Step

Our entity embedding framework was implemented in the Tensorflow framework. Once we had the target knowledge base, we could train the entity embedding. It was an off-line work. Our global model and local model were both implemented in the Pytorch framework. Only when the mentions and contexts were provided, could we link the mentions to their corresponding entities through the model; thus, it was an online work.

5.3.1. Hyper-Parameters Setting

Parameters for entity embedding model: We used Wikipedia (2014Feb) as our training data. We used existing links in Wikipedia with anchors as mentions and links as true entities. The window size of entity context was set as 5. As for description for each entity, we used the first 150 words in its corresponding Wikipedia page as input. The hyper-parameters of our entity embedding model are summarized in Table 3. We used different combinations of parameter settings to train our model, and the parameters settings with which our model could obtain the best performance are given.

Table 3. Hyper-parameters for the entity embedding model. Here, we choose Adam as our optimizer, with learning rate of 0.05.

Parameters	Search Space	Value
dim of v_e,v_{desc},v_c	{100,200,300}	200
dropout rate	{0.2,0.3,0.4,0.5}	0.4
batch size	{300,600,900,1200}	600

Hyper-parameters for local model and global model: The window size of the mention context was set as 5. The hyper-parameters of our global model are summarized in Table 4.

Table 4. Hyper-parameters for the global model. We use Adam for optimization, with learning rate of 0.005.

Parameters	Search Space	Value
dim of d_1	{200,300,400}	300
dim of d_2	{10,20,30,40}	30
dim of hidden state in Bi-LSTM	{50,100,150,200}	100
dropout rate	{0.2,0.3,0.4,0.5}	0.4
λ_1	[0,0.2] with step size 0.04	0.1
λ_1	[0,0.2] with step size 0.04	0.04

5.3.2. Evaluation Matrix

Here, we design experiments to verify the validity of our main contributions. We give the result of our final global model, compared to other advanced entity linking systems, to show that our model could obtain a competitive, or even better, performance than other advanced entity linking system. Our models are trained on AIDA-train, validated on AIDA-A, and tested on AIDA-B and the other datasets, mentioned in Section 5.1.

Here we focus on six variations of our model: To explore the validity of our attention mechanism, we designed three variations of our local model which disambiguate the given mentions independently, only by each mention context with pretrained entity embedding, making use of both entity context and entity description: (1) **LSTM-MEAN**: Local model without attention mechanism, the importance scores of the words in the mention context were assumed to be the same. (2) **LSTM-A**: Local model with CRF-based attention mechanism, using a CRF layer which captured the important text spans for disambiguating the current mention. (3) **LSTM-A-R**: Based on the model LSTM-A, we introduced the two regularizers, as mentioned in Section 4.2, to enforce the model to focus on the text spans (which is really important for entity linking). To explore the validity of our entity embedding framework, we designed another three variations of our global model, based on **LSTM-A-R**. The three global models differed from the entity embedding framework they involved. (4) **Global Model C**: The entity embedding framework in this model only made use of entity context. (5) **Global Model D**: The entity embedding framework in this model only captured the information of entity description. (6) **Global Model CD**: The entity embedding framework in this model took both entity description and entity context into consideration. We explored the role of coherence among mentions in our linking system, by comparing **LSTM-A-R** and **Global Model CD**.

We compared our model with the following existing advanced models: (1) **Hoffart et al., 2011** [22]: A supervised model, based on feature engineering, which was trained on CoNLL. (2) **Landau et al., 2016** [16]: A recent method which combined benefits of deep learning and feature engineering. (3) **Gupta et al., 2017** [4]: Trained only on Wikipedia without optimizing on a special dataset, this model had a strong generalization ability.

Metrics: We use the F1:

$$precision = \frac{num\_true\_positives}{get\_dataset\_num\_non\_entity\_candidats(dataset)}, \tag{25}$$

$$recall = \frac{num\_true\_positives}{dataset\_num\_mentions}, \tag{26}$$

$$F1 = \frac{2 * precision * recall}{precision + recall}. \tag{27}$$

We assume that all the mentions are provided, and we do not predict NIL (unlinkable) mentions. We only consider a linking to be correct when the first entity in the ranking result is equal to the ground truth.

Table 5 shows F1 scores of the existing model and several variations of our model on AIDA. From the results of **LSTM-MEAN**, **LSTM-A**, and **LSTM-A-R**, we find that our linking system can obtain a great improvement with the help of our attention mechanism, and the regularizers designed in Section 4.2 could enforce our model's focus on the text, an important factor in disambiguating the given mention. From the results of the three variations of our global model, we can see that our global model achieved a better performance with the entity embedding framework, which could capture richer entity information aspects. By comparing **LSTM-A-R** and **Global Model CD**, we see that the coherence among mentions can be used to further improve the system, by use of our algorithm.

Table 5. F1 scores of different models on CoNLL (%). The performance of the state-of-the-art models are reported in [4] (Hoffart et al., Landau et al., Gupta et al.).

Models	AIDA-B	AIDA-A
Hoffart et al., 2011 [22]	81.8	-
Landau et al., 2016 [16]	85.5	86.9
Gupta et al., 2017 [4]	82.9	84.9
LSTM-MEAN	83.4	84.2
LSTM-A	83.8	89.7
LSTM-A-R	84.4	90.7
Global Model C	87.1	90.3
Global Model D	86.9	90.3
Global Model CD	**87.6**	**91.1**

We re-implemented some of the state-of-the-art models and tested on the other datasets. The results are shown in Table 6.

Our model is trained only on AIDA-train without optimizing on a special dataset. The results show that our model obtained better performance than the baselines across different datasets, which means that our model has a strong generalization ability.

Table 6. Results on other datasets.

Models	ACE04	AQUAINT	WNED-CWEB
Hoffard et al., 2011 [22]	56	80	58.6
Ratinov et al., 2011 [26]	83	82	56.2
Fang et al., 2016 [27]	85.3	88.8	-
Global Model CD	86.6	87.3	73.1

5.4. Case Analysis

Here, we give our analysis of the results of our experiments.

In the Figure 7, the x-axis denotes the words in the context and the y-axis denotes the importance score of each word. Here, the word "**China**" is the current mention, and its corresponding entity is "**China national football team**". From this case, we can see that the mention word "**China**" gains the highest importance among all the words in the given context. Generally, we find the mention words usually have a high importance score in almost all the cases. This is because the mention words, themselves, are strongly related to its corresponding entity. You can almost infer the most possible corresponding entity from the mention. Besides the mention words, our attention mechanism could also capture some other text spans which are helpful in entity disambiguation. In this case, according to the statistics, the two most possible entities for the mention "**China**" are "**China (country)**" and "**China national football team**". The attention mechanism captured the text spans, which are helpful for distinguishing the two entities. Here, we find that the words "**championship**", "**their**", and "**them**" gain the highest score, except for the mention word "**China**". The words "**their**" and "**them**" tend to refer to a group of people, rather than a country, and the word "**championship**" has a close relation to sports. Focusing on those words, we could infer that the mention of "**China**", in this

case, refers to the entity "**China national football team**" with a higher probability than the entity "**China (country)**". We find that words about time often obtain a low importance score, such as "**Friday**" in this case. Some other words, such as prepositions like "**on**" and conjunctions like "**But**" usually have low importance scores as well. This is similar to the way that people think. People nearly do not disambiguate the given mentions by these words.

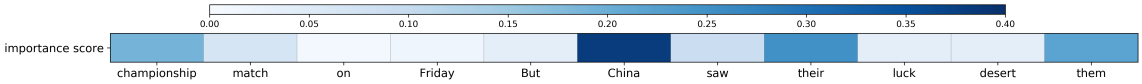

Figure 7. The importance scores distribution of a case in AIDA.

From Table 5, we find that our model obtained a better performance on AIDA-A than AIDA-B. This is because there are some cases of AIDA-B in which it is hard to disambiguate the given mention only by its local context. For example, consider the context "**0 0 2 1 3 Syria 1 0 0 1 1**" and the mention "**Syria**". The ground truth of this case was the entity "**Syria national football team**". However, according to our statistics, the most possible entity for the mention "**Syria**" was the entity "**Syria(country)**". Unfortunately, the context of the mention "**Syria**" consisted of digital numbers, and it is hard to tell whether the mention refers to a country or a football team by these digital numbers. In cases like this, we may need some other information to infer the corresponding entity of the given mention, such as coherence among mentions. There were 47 similar cases in AIDA-B. So, we can see, with the help of coherence among mentions, our global model obtained a much better performance than our local model in the test set AIDA-B. We also analyzed some other wrong cases. In those cases, most of the mentions had a low frequency to map to the correct entity, according to statistics in Wikipedia. Our model tended to link the mention to the entity which gained a high frequency in the statistics.

6. Conclusions

In this paper, we focused on how to build a high-quality representation of mention and entity. To encode entity semantics, we built a function to incorporate different aspects of information about entities, in order to obtain dense unified embeddings. For the representation of mention, we introduced a novel attention mechanism which could capture important text spans. In addition, we calculated the topic information with a Forward-Backward Algorithm, which could be helpful to entity linking, especially when we can not predict the correct entity only by its local context. There are still some avenues for future work. Firstly, the semantics of entities could also come from their type and the relations between entities. Taking all of them into consideration could yield a higher quality entity embedding. Secondly, the assumption that all the mentions in a same context may share several topics would be more suitable, and we believe it is worth further research.

Author Contributions: C.L. designed the algorithm and performed the experiments; C.L. and H.H. analyzed the data. C.L. and F.L. wrote and revised the manuscript; X.S. and F.L. discussed the data and corrected the manuscript; All authors have read and approved the final manuscript.

Funding: This work was supported, in part, by the Dongguan Science and Technology Fund (Grant No. 2016108101008).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sun, Y.; Lin, L.; Tang, D.; Yang, N.; Ji, Z.; Wang, X. Modeling mention, context and entity with neural networks for entity disambiguation. In Proceedings of the IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 1333–1339.

2. Han, X.; Sun, L.; Zhao, J. Collective Entity Linking in Web Text: A Graph-based Method. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11), Beijing, China, 24–28 July 2011; ACM: New York, NY, USA, 2011; pp. 765–774. [CrossRef]
3. Zwicklbauer, S.; Seifert, C.; Granitzer, M. Robust and Collective Entity Disambiguation Through Semantic Embeddings. In Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'16), Pisa, Italy, 17–21 July 2016; ACM: New York, NY, USA, 2016; pp. 425–434. [CrossRef]
4. Gupta, N.; Singh, S.; Roth, D. Entity Linking via Joint Encoding of Types, Descriptions, and Context. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 2681–2690. [CrossRef]
5. Guo, Z.; Barbosa, D. Robust Entity Linking via Random Walks. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM'14), Shanghai, China, 3–7 November 2014; ACM: New York, NY, USA, 2014; pp. 499–508. [CrossRef]
6. Chen, Z.; Tamang, S.; Lee, A.; Li, X.; Lin, W.; Snover, M.G.; Artiles, J.; Passantino, M.; Ji, H. CUNY-BLENDER TAC-KBP2010 Entity Linking and Slot Filling System Description. In Proceedings of the Third Text Analysis Conference, TAC 2010, Gaithersburg, MD, USA, 15–16 November 2010.
7. Pappu, A.; Blanco, R.; Mehdad, Y.; Stent, A.; Thadani, K. Lightweight Multilingual Entity Extraction and Linking. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM'17), Cambridge, UK, 6–10 February 2017; ACM: New York, NY, USA, 2017; pp. 365–374. [CrossRef]
8. Ganea, O.E.; Hofmann, T. Deep Joint Entity Disambiguation with Local Neural Attention. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 2619–2629. [CrossRef]
9. Yamada, I.; Shindo, H.; Takeda, H.; Takefuji, Y. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, 11–12 August 2016; pp. 250–259.
10. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112.
11. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1106–1114.
12. Stajner, T.; Mladenic, D. Entity Resolution in Texts Using Statistical Learning and Ontologies. In Proceedings of the Semantic Web, Fourth Asian Conference, ASWC 2009, Shanghai, China, 6–9 December 2009; pp. 91–104. [CrossRef]
13. Phan, M.C.; Sun, A.; Tay, Y.; Han, J.; Li, C. Pair-Linking for Collective Entity Disambiguation: Two Could Be Better Than All. *CoRR* **2018**, abs/1802.01074. Available online: http://xxx.lanl.gov/abs/1802.01074 (accessed on 31 January 2019). [CrossRef]
14. Wang, B.; Lu, W. Learning Latent Opinions for Aspect-level Sentiment Classification. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, LA, USA, 2–7 February 2018; pp. 5537–5544.
15. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, NV, USA, 5–8 December 2013; pp. 3111–3119.
16. Francis-Landau, M.; Durrett, G.; Klein, D. Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016), San Diego, CA, USA, 12–17 June 2016; pp. 1256–1261.

17. Association for Computational Linguistics (ACL). *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7–12, 2016, Berlin, Germany, Volume 1: Long Papers*; The Association for Computer Linguistics: Stroudsburg, PA, USA, 2016.
18. Moro, A.; Raganato, A.; Navigli, R. Entity Linking meets Word Sense Disambiguation: A Unified Approach. *TACL* **2014**, *2*, 231–244. [CrossRef]
19. Usbeck, R.; Ngomo, A.N.; Röder, M.; Gerber, D.; Coelho, S.A.; Auer, S.; Both, A. AGDISTIS—Graph-Based Disambiguation of Named Entities Using Linked Data. In Proceedings of the Semantic Web—ISWC 2014—13th International Semantic Web Conference, Riva del Garda, Italy, 19–23 October 2014; Part I, pp. 457–471. [CrossRef]
20. Kulkarni, S.; Singh, A.; Ramakrishnan, G.; Chakrabarti, S. Collective annotation of Wikipedia entities in web text. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June 28–1 July 2009; pp. 457–466. [CrossRef]
21. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, 25–29 October 2014; A Meeting of SIGDAT, a Special Interest Group of the ACL; pp. 1746–1751.
22. Hoffart, J.; Yosef, M.A.; Bordino, I.; Fürstenau, H.; Pinkal, M.; Spaniol, M.; Taneva, B.; Thater, S.; Weikum, G. Robust Disambiguation of Named Entities in Text. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, John McIntyre Conference Centre, Edinburgh, UK, 27–31 July 2011; A meeting of SIGDAT, a Special Interest Group of the ACL; pp. 782–792.
23. Guo, Z.; Barbosa, D. Robust named entity disambiguation with random walks. *Semant. Web* **2018**, *9*, 459–479. [CrossRef]
24. Gabrilovich, E.; Ringgaard, M.; Subramanya, A. *FACC1: Freebase Annotation of ClueWeb Corpora, Version 1 (Release Date 2013-06-26, Format Version 1, Correction Level 0)*; 2013. Available online: https://www.researchgate.net/publication/267026725_FACC1_Freebase_annotation_of_ClueWeb_corpora_Version_1_Release_date_2013-06-26_Format_version_1_Correction_level_0 (accessed on 29 January 2019).
25. Spitkovsky, V.I.; Chang, A.X. A Cross-Lingual Dictionary for English Wikipedia Concepts. In Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, 23–25 May 2012; pp. 3168–3175.
26. Ratinov, L.; Roth, D.; Downey, D.; Anderson, M. Local and Global Algorithms for Disambiguation to Wikipedia. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 1375–1384.
27. Fang, W.; Zhang, J.; Wang, D.; Chen, Z.; Li, M. Entity Disambiguation by Knowledge and Text Jointly Embedding. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, 11–12 August 2016; pp. 260–269.