Joint Entity Linking for Web Tables with Hybrid Semantic Matching

Jie Xie[1,2], Yuhai Lu[1,2], Cong Cao[1,2(✉)], Zhenzhen Li[2],
Yangyang Guan[2], and Yanbing Liu[2]

[1] School of Cyber Security, University of Chinese Academy of Sciences,
Beijing, China
{xiejie,luyuhai,caocong}@iie.ac.cn
[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing,
China
{lizhenzhen,guanyangyang,liuyanbing}@iie.ac.cn

Abstract. Hundreds of millions of tables on the World-Wide Web contain a considerable wealth of high-quality relational data, which has already been viewed as an important kind of sources for knowledge extraction. In order to extract the semantics of web tables to produce machine-readable knowledge, one of the critical steps is table entity linking, which maps the mentions in table cells to their referent entities in knowledge bases. In this paper, we propose a novel model *JHSTabEL*, which converts table entity linking into a sequence decision problem and uses hybrid semantic features to disambiguate the mentions in web tables. This model captures local semantics of the mentions and entities from different semantic aspects, and then makes full use of the information of previously referred entities for the subsequent entity disambiguation. The decisions are made from a global perspective to jointly disambiguate the mentions in the same column. Experimental results show that our proposed model significantly outperforms the state-of-the-art methods.

Keywords: Table entity linking · Hybrid semantic matching · Joint disambiguation

1 Introduction

The World-Wide Web contains billions of relational data in the form of HTML tables, i.e. web tables (Cafarella et al. [1, 8]; Lehmberg et al. [2]), which carries valuable structured information. This high-quality relational data is an important data source for knowledge extraction on the Web.

In order to make machines to understand these tables, one of the critical steps is to map the mentions in table cells to their corresponding entities in a given knowledge base (KB), which is called table entity linking or table entity disambiguation. For example, in the web table in Fig. 1, this task aims to link the mention "Louvre" in the first column to the entity "Louvre Museum" in Wikipedia. Table entity linking is an important and challenging stage in table semantic understanding since the mentions in tables are usually ambiguous.

Museum	Location	Annual Visitors
Louvre	🇫🇷 Paris	10,200,000[10][11]
National Museum of China	🇨🇳 Beijing	8,610,092[11]
The Metropolitan Museum of Art	🇺🇸 New York City	6,953,927[11]
Vatican Museums	🇻🇦 Vatican City (🇮🇹 Rome)	6,756,186[11]
National Air and Space Museum	🇺🇸 Washington, D.C.	6,200,000[12][10]

Fig. 1. An example of web table describing the information of museums.

In this paper, we only focus on tables where rows clearly represent separate tuple-like objects, and columns represent different dimensions of each tuple (similar to Fig. 1). Additionally, since this paper does not focus on how to determine which cells can be linked to the knowledge base, we assume that the linkable mentions are already known and perform entity linking on these linkable mentions, excluding un-linkable content, such as numbers, etc.

Compared with entity linking in free-format text, it is more difficult to disambiguate mentions in tables due to the less context of table cells. The existing researches mainly used collective classification techniques [3], graph-based algorithm [4], multi-layer perceptron [5], etc. to solve this problem. These methods do not capture the semantic features of mentions and entities well, and can't yield desired disambiguation effect. In order to better represent mentions and entities, we use a hybrid semantic matching model to capture the local semantic information between table mentions and candidate entities from different semantic aspects.

Since tables have the property of column consistency, that is, cells in the same column have similar contents and belong to the same category, it is natural to jointly disambiguate the mentions in the same column. In addition, we have noticed that mentions usually have different difficulty in disambiguating depending on the quality of the contextual information. If we sort the mentions in the same column and start with mentions that are easier to disambiguate, it would be useful to utilize the information of previously referred entities for the subsequent entity disambiguation.

In this paper, we propose a joint model with hybrid semantic matching for table entity linking, which is called *JHSTabEL* for short. This model consists of two modules: Hybrid Semantic Matching Model and Global Decision Model. The Hybrid Semantic Matching Model encodes the contextual information of each mention and its candidate entities. It uses the representation-based and interaction-based models to capture matching features at abstract and concrete levels respectively, and then aggregates them to obtain the hybrid semantic features, based on which the similarity scores of the mentions and entities are calculated. Before entering the global model, the mentions in the same column are sorted according to local similarity scores. The Global Decision Model uses an LSTM network to encode the local representations of mention-entity pairs and jointly disambiguate the mentions via a sequential manner. In summary, we make the following contributions:

- We propose a hybrid semantic matching model which aggregates complementary abstract and concrete matching features to make full use of the local context.
- We use a global decision model to jointly disambiguate the mentions in the same column. The disambiguation is made from a global perspective.
- We evaluate our model on web table datasets and the experimental results show that our model significantly outperforms the state-of-the-art methods.

2   Related Work

WebTables [1, 8] showed that the World-Wide Web consisted of a huge number of data in the form of HTML tables, and pioneered the study of tables on the Web as a high-quality relational data source. Since then, various efforts have been made to extract semantics from web tables. These efforts usually contain but not limited to three tasks: table entity linking, column type identification and table relation extraction.

Syed et al. [16] presented a pipeline approach, which first inferred the types of columns, then linked cell values to entities in the given KB, finally selected appropriate relations between columns. Mulwad et al. [6] and Limaye et al. [13] described approaches to jointly model entity linking, column type identification and relation extraction tasks using graphical model. These models, which handle all three tasks at the same time, rely on the correctness and completeness of the knowledge base, and therefore may run the risk of negatively affecting the performance of entity linking.

There are also some works that only focus on the task of table entity linking [3–5, 9, 15]. Shen et al. [15] linked the mentions in list-like web tables (multiple rows with one column) to the entities in a knowledge base. Efthymiou et al. [9] proposed three unsupervised annotation methods and attempted to map each table row to an entity in a KB. This work was based on the assumption that the entity columns of tables were already known and their values served as the names of the described entities. Bhagavatula et al. [3] presented TabEL which used a collective classification technique to collectively disambiguate all mentions in web tables. Wu et al. [4] constructed a graph of mentions and candidate entities and used page rank to determine the similarity scores between mentions and candidates. In the above methods, a lot of hand-designed features are applied, which is time-consuming and laborious. Recently, with the popularity of deep learning models, representation learning is used to automatically capture semantic features. Luo et al. [5] proposed a neural network method for cross-language table entity linking. It took some embedding features as inputs and used a two-layer fully connected network to perform entity linking. This model only used simple coherence features and a MLP network to link all mentions in tables, thus cannot achieve desired linking effect.

In this paper, we automatically capture the semantic features of the mentions and candidate entities from different aspects to fully use the local information, and then use a global model to disambiguate the mentions in web tables in a global perspective.

3 Methodology

As shown in Fig. 2, the overall structure of *JHSTabEL* consists of two parts: the hybrid semantic matching model which encodes the contextual information from two different semantic aspects to obtain the local semantic representations and matching scores of the mentions and the candidate entities; the global decision model which makes decisions from a global perspective to jointly disambiguate the mentions in the same column. We will introduce the details of these two parts in this section.
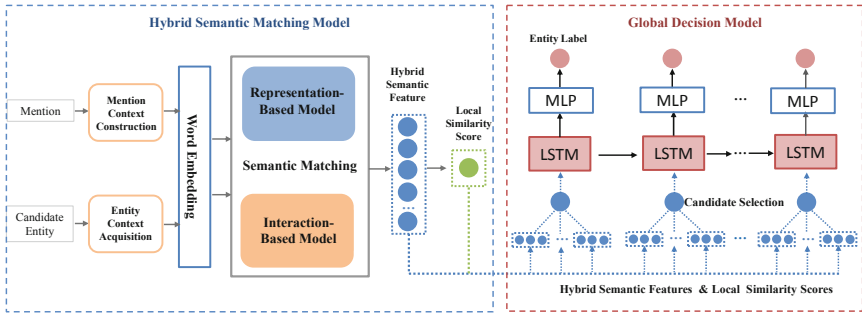

Fig. 2. The overall structure of our proposed model for table entity linking.

3.1 Preliminaries

Before introducing our model, we firstly make a definition of the table entity linking task. Formally, Given a table T with n rows and m columns, each mention in the table can be represented as $M_{i,j}$, $1 \leq i \leq n$ and $1 \leq j \leq m$ being the indexes of the row and column respectively. We model T as a bi-dimensional array of $n \times m$ cells, limiting the research scope to tables with no column branches into sub columns. Each mention $M_{i,j} \in T$ in the tables has a set of candidate entities $C_{M_{i,j}} = \left\{ e_{i,j}^1, e_{i,j}^2, \ldots, e_{i,j}^r \right\}$, where $e_{i,j}^k$ is the possible referred entity in the given knowledge base. Then the task of table entity linking is to map each mention $M_{i,j}$ to its corresponding target entity $e_{i,j}^+$ or return "NIL" if there is no correct target entity in the KB.

For each mention in the web tables, we need to generate its candidate referent entities from a given knowledge base. Hence, we use several heuristic rules to obtain the candidates: (i) the mention's redirect and disambiguation page in Wikipedia; (ii) exact match of the string mention; (iii) fuzzy match (e.g., edit distance) of the string mention; (iv) entities containing the n-grams of the mention.

To optimize the memory and avoid unnecessary calculations during model training, we use the XGBoost model to simplify the candidate sets. The features used in XGBoost are the edit distance between the mentions and their candidate entities, the semantic similarity between the mention context representations and the entity embeddings, and the statistical features based on the pageview and hyperlinks in Wikipedia. Then we take top K scored entities for each mention based on this model. In

contrast, if the number of candidate entities for a mention is less than K, we complement it with negative examples from its candidate set.

3.2   Hybrid Semantic Matching Model

Given a mention M and its corresponding candidate set $C_M = \{e^1, e^2, \ldots, e^r\}$, we aim to get a local representation and a match score for each mention-entity pair. This is essentially a semantic matching problem between the mention context X_M and the candidate entity context X_e. Due to the scarce context of table cells, we construct the mention context X_M by using the other mentions in the row and the column of the table where the mention exists, and represent them as word embeddings using a pre-trained lookup table [7]. The context X_e of the candidate entity is obtained from the abstract of its corresponding page in Wikipedia and embedded in the same way.

Existing neural semantic matching models can be divided into two categories: representation-based model and interaction-based model. The representation-based model first uses a neural network to construct a representation for a single text, such as a mention context or an entity abstract, and then conducts matching between the abstract representations of two pieces of text. The interaction-based method attempts to establish a local interaction (e.g., cosine similarity) between two pieces of text, and then uses a neural network to learn the final matching score based on the local interaction.

The representation-based and interaction-based models can capture abstract and concrete level matching signals respectively. In this paper, we propose to fuse these two models to perform semantic matching between the mention and entity contexts. The left part of Fig. 2 shows the structure of our local hybrid model. It takes the mention and candidate entity as inputs and generates their corresponding contexts and embeddings, which are passed into the representation and interaction models. Finally, the hybrid semantic features and local ranking scores are acquired from this hybrid model. In the remaining of this section, we will introduce the details of these two sub-models and discuss the advantages of fusing them.

Representation-Based Model. Given the mention context $X_M = \{w_M^1, w_M^2, \ldots, w_M^p\}$ and the candidate entity context $X_e = \{w_e^1, w_e^2, \ldots, w_e^q\}$, we aim to get their abstract representations using siamese LSTM [10] with tied weights. Figure 3 illustrates the architecture of our representation-based model. The mention context embedding Emb_M and the entity context embedding Emb_e are obtained from a pre-trained lookup table [7]. We use two networks $LSTM_a$ and $LSTM_b$ with tied weights to encode the embeddings separately, and take the last hidden states of the LSTM networks as the representations of the word sequences. In this way, we get the mention representation V_M and the entity representation V_e, and feed their concatenation result to a multi-layer perceptron (MLP). The output layer of the MLP produces a feature vector $V_{abs}(M, e)$ of d_{abs} dimension.

$$V_{abs}(M, e) = MLP([V_M; V_e]) \tag{1}$$

In this way, we extract abstract-level features V_{abs} of the local contexts. We can also calculate the local similarity between mention and candidate entity using the abstract-level features. However, if only this representation-based approach is used, the concrete matching signals (e.g., exact match) are lost, since the matching happens after their individual representations. So next we will introduce an interaction-based model to better capture the concrete matching features to complement the representation-based model.
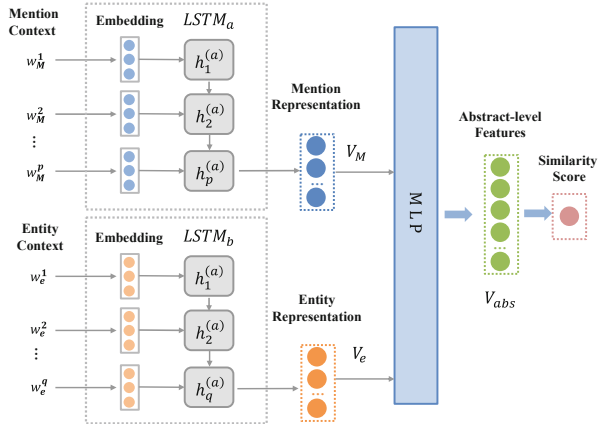

Fig. 3. The architecture of representation-based model using siamese LSTM.

Interaction-Based Model. Inspired by the latest advances in information retrieval [11, 12], we propose to use an interaction-based approach to capture the concrete-level features. The interaction-based model using Conv-KNRM [12] attempts to establish local interactions (e.g., cosine similarity) and get concrete-level features between mention and entity contexts. As shown in Fig. 4, the Conv-KNRM model first composes n-gram embeddings using CNN networks, and then constructs translation matrices between n-grams of different lengths in the n-gram embedding space. It uses a kernel-pooling layer to count the soft matches of word or n-gram pairs and gets the concrete level features.

The Conv-KNRM model takes the mention context embedding Emb_M and the entity context embedding Emb_e as inputs. The convolutional layer applies convolution filters to compose n-grams from the text embeddings. For each window of h words, the filter sums up all elements in the h words' embeddings $Emb_{i:i+h}$, weighted by the filter weights. Using F different filters of size h gives F scores for each window position, represented by a score vector $\overrightarrow{g}_i^h \in \mathbb{R}^F$. Each of the values in $\overrightarrow{g}_i^h$ describes the text in the i_{th} window in a different perspective:

$$\overrightarrow{g}_i^h = relu(W^h \cdot Emb_{i:i+h} + \overrightarrow{b}^h), i = 1 \ldots n. \tag{2}$$

Where W^h and $\overrightarrow{b}^h$ are the weights of F convolution filters. Then the convolution feature matrix for h-gram can be obtained by concatenating convolution outputs $\overrightarrow{g}_i^h$.

After getting the word-level n-gram feature matrices, the cross-match layer constructs translation matrices using n-grams of different lengths. For mention n-grams of length h_M and entity n-grams of length h_e, a translation matrix TM^{h_M,h_e} is constructed by calculating their cosine similarity.

$$TM_{i,j}^{h_M,h_e} = \cos\left(\overrightarrow{g}_i^{h_M}, \overrightarrow{g}_j^{h_e}\right) \tag{3}$$

Then the Kernel-pooling is applied to each TM^{h_M,h_e} matrix to generate the concrete feature vector $\phi\left(TM^{h_M,h_e}\right)$, which describes the distribution of match scores between mention h_M-grams and entity h_e-grams.

$$\phi\left(TM^{h_M,h_e}\right) = \sum_{i=1}^{n} \log \vec{K}\left(TM_i^{h_M,h_e}\right) \tag{4}$$

$$\vec{K}\left(TM_i^{h_M,h_e}\right) = \left\{K_1\left(TM_i^{h_M,h_e}\right), \ldots, K_k\left(TM_i^{h_M,h_e}\right)\right\} \tag{5}$$

Where $\vec{K}\left(TM_i^{h_M,h_e}\right)$ applies k RBF kernels to the i-th row of the translation matrix $TM_i^{h_M,h_e}$, and then generates a k-dimensional feature vector. Each kernel calculates how pairwise similarities between n-gram feature vectors are distributed around its mean μ_k. The more similarities closed to its mean, the higher the output value is.

$$K_k\left(TM_i^{h_M,h_e}\right) = \sum_j \exp\left(-\frac{\left(TM_{i,j} - \mu_k\right)^2}{2\sigma_k^2}\right) \tag{6}$$

Then each of the translation matrices is pooled to a k-dimensional vector, and the concatenation of these vectors produces a scoring feature vector $\phi(TM)$.
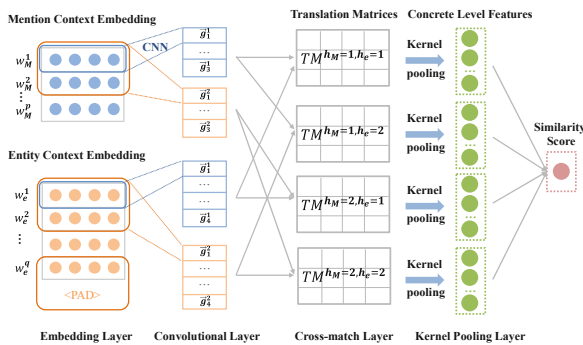

Fig. 4. The architecture of interaction-based model using Conv-KNRM.

In this way, we capture the concrete features $V_{con}(M, e)=\phi(TM)$ based on the word-level n-gram interactions between mention and entity. These features can complement the abstract features for a better semantic representation.

Hybrid Semantic Matching. We use the two sub-models introduced above to capture the abstract and concrete level features respectively, and combine them to get the hybrid semantic features. Then we pass the concatenation result to a MLP network to get the local similarity score for each mention-entity pair.

$$sim(M, e) = MLP([V_{abs}(M, e); V_{con}(M, e)]) \tag{7}$$

In order to better distinguish the correct entity from the wrong entities in the candidate set when training the hybrid model, we use the hinge loss function, which can rank the correct entity higher than others. The loss function of the hybrid model is defined as follow:

$$L_{local} = \sum_{M} \sum_{e^+, e^- \in C_M^{+,-}} max(0, \gamma - sim(M, e^+) + sim(M, e^-)) \tag{8}$$

Where $C_M^{+,-}$ is the set of pairwise preferences of M and e^+ ranks higher than e^-. $\gamma > 0$ is the margin parameter, indicating that the score of the positive target entity e^+ is at least a margin γ higher than the negative entity e^-.

Through the hybrid semantic matching model, we obtain the hybrid semantic features and local similarity scores of the mentions and candidate entities, which will serve as inputs to the subsequent global decision model.

3.3 Global Decision Model

The global decision model aims to enhance the topical consistency among the mentions in the same column. As shown in the right part of Fig. 2, the global decision model takes the hybrid semantic features and local similarity scores acquired from the hybrid semantic matching model as inputs, and uses an LSTM network to deal with mentions in a sequence manner. The LSTM network can maintain a long-term memory on features of entities selected in previous states. Therefore, the column consistency information can be fully utilized when disambiguating entities.

Inspired by [14], we sort the mentions in the same column when disambiguating them. In table entity linking task, it is natural to divide all the mentions in a table into multiple segments according to the column they belong to. Then the mentions in a segment are sorted according to the local similarity scores, the one with a higher score is placed first. We take the maximum local similarity between the mention and its corresponding candidate entities as the criterion for each mention when sorting. Then an LSTM network is used to deal with these sorted segments in a sequence manner. In this way, we can start with mentions that are easier to disambiguate and utilize the information provided by previously selected entities to disambiguate subsequent mentions.

The local similarity score indicates the probability of an entity being the target entity of the mention. Therefore, at each time step, we randomly select a candidate for the mention based on this probability, and take the corresponding hybrid representations of the mention and the selected entity as inputs to LSTM network. Then the output at each time step is passed into a MLP network to produce the label for the selected entity. The objective function of the global decision model is defined as follow:

$$L_{global} = -\frac{1}{n}\sum_{x} [y \log y' + (1 - y) \log(1 - y')] \qquad (9)$$

Where $y \in \{0, 1\}$ is the actual label of the candidate entity and $y' \in \{0, 1\}$ is the predicted one. In this way, the mentions in the same column are disambiguated jointly.

4   Experiment

In this section, we conduct several experiments to evaluate our model *JHSTabEL* on the sampled web tables. Firstly, we compare it with the state-of-the-art methods, and then discuss the effect of various components of our proposed model.

4.1   Experiment Setup

Dataset. We use the dataset constructed by Wu et al. [4], which contains 123 tables extracted from Chinese Wikipedia. The mentions in these tables are labeled by their corresponding Wikipedia articles. We represent this dataset as *Dataset-Wu*. In order to better reflect the advantages of the deep learning method, we expand the dataset by randomly collecting 117 tables from the Web. Each mention in these tables is manually mapped to its corresponding entity in Wikipedia. Then these tables are added to *Dataset-Wu* to generate a larger dataset, represented as *Dataset-Xie*. The average size of tables in this dataset is 12 rows, and each table contains an average of 38.2 mentions. Totally, we obtains 9168 mentions from 240 tables. We randomly split the tables into training, validation and testing sets (70%, 10%, 20%) for experiments.

Parameter Setting. The hyper parameters of our model are obtained from the best validated model. For the representation-based model, the number of LSTM cell units is set to 128, the batch size is 64 and the number of MLP layers is 3. For the interaction-based model, the n-gram lengths are $h = 1, 2, 3$, the number of CNN filters is $F = 128$, the number of kernels is set to 11, the first one is exact match kernel $\mu = 1$, $\sigma = 10^{-3}$, and the other 10 kernels equally split the cosine range $[-1, 1]$: the values of μ are $\mu_1 = 0.9, \mu_2 = 0.7, \ldots, \mu_{10} = -0.9$ and the values of σ are set to be 0.1. We set the rank margin $\gamma = 0.1$ for hybrid semantic matching model. For the global decision model, the number of LSTM cell units is 256, the batch size is 32, and the number of MLP layers is 2. We choose a learning rate of 1e-4 and a probability of dropout of 0.9. The dimension of the word embedding used in our experiments is set to 300. To optimize

the memory and avoid unnecessary calculations, we select top K candidate entities for each mention. Our experiments show that the best performance is obtained when $K = 5$.

4.2 Baselines and Evaluation Metric

Baselines. We compare our model *JHSTabEL* with several table entity linking methods, which reported state-of-the-art results: collective classification model (*TabEL (2015)* [3]), graph-based model (*Wu et al. (2016)* [4]) and MLP-based model (*Luo et al. (2018)* [5]). Besides, we feed Luo et al.'s mention features and context features into our proposed global decision model, which is represented as *Luo-fea-Global*. *JHSTabEL (local)* is a degenerate version of our proposed model, it only uses the local hybrid semantic matching model to disambiguate mentions. This local hybrid model fuses the abstract and concrete matching features to rank the candidate entities for table mentions.

Evaluation Metric. In order to be consistent with the state-of-the-art table entity linking methods, we evaluate the results with Micro Accuracy and Macro Accuracy. Micro Accuracy is the fraction of correct linked cells over the whole dataset and Macro Accuracy is the average correct ratio over different tables.

4.3 Experiment Result

Comparing with Previous Work. We compare our proposed model *JHSTabEL* with the baselines on the *Dataset-Wu* and *Dataset-Xie* and report the experimental results in Table 1. From the results, we can notice that the model *JHSTabEL(local)* is comparable to *Luo et al. (2018)*, which jointly disambiguated all mentions in a table and reported the best results so far. This result shows the excellent feature extraction ability of our hybrid semantic matching model, which can better characterize the mention-entity pairs and obtain good results using only local matching. The model *Luo-fea-Global* outperforms *Luo et al. (2018)*, indicating the effectiveness of our global decision model. The model proposed by Luo et al. applied vector averaging over all cells to be linked and concatenated coherence feature to link all mentions in a table in the same time. Due to the simplicity of the coherence feature, this method does not model the correlations between table mentions very well. However, our global decision model uses an LSTM network to maintain the memory of previously selected entities, so as to obtain better joint disambiguation effects. Our full model *JHSTabEL* achieves the best result on both datasets. Compared with *Luo et al. (2018)*, it improves the Micro Accuracy and Macro Accuracy by absolute gains of 0.020 and 0.018 separately on *Dataset-Wu*. Besides, in order to get more reliable results, we enrich the origin dataset (*Dataset-Wu*) and generate a larger dataset (*Dataset-Xie*), which is about 1.95 times of the origin one. Then we perform the same experiments on it. The results still show the superiority of our proposed model.

Table 1. Accuracies comparison between our model and baselines on two datasets.

Methods	Dataset-Wu		Dataset-Xie	
	Micro Acc.	Macro Acc.	Micro Acc.	Macro Acc.
TabEL (2015)	0.845	0.843	0.850	0.847
Wu et al. (2016)	0.849	0.845	0.854	0.848
Luo et al. (2018)	0.878	0.864	0.884	0.867
Luo-fea-Global	0.885	0.872	0.890	0.875
JHSTabEL(local)	0.877	0.866	0.885	0.865
JHSTabEL	**0.898**	**0.882**	**0.907**	**0.889**

Comparison Between Different Semantic Matching Models. To further explore the differences between two semantic matching models (Representation-Based and Interaction-Based) and discuss the benefits when combing them, we remove the representation model and interaction model from the full model separately and compare their performance with the full model. As shown in Fig. 5, we can observe that *Rep-Based + Global Model* performs comparably with *Int-Based + Global Model* and the full model *JHSTabEL* obtains considerable performance gains on both of the two datasets. This comparison result indicates that these two sub-models capture complementary information for entity disambiguation. In fact, the interaction-based model builds the n-gram level local interactions between texts, thus can capture the concrete matching information. However, the concrete information might be lost in representation-based model as it tends to capture the whole meaning of the text and generate abstract information. So we will benefit a lot by combining different semantic matching signals from these two models.
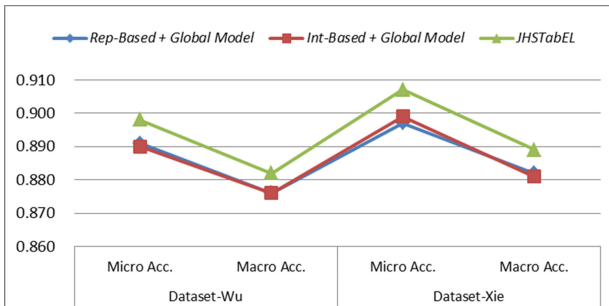
**Fig. 5.** The performance of different semantic matching models.

Effect of Global Decision Model. In order to evaluate whether the global decision model based on LSTM network contributes to disambiguation, we compare the performance with and without the global model. From the results in Table 2, we can see that the accuracies of each local model are greatly improved when combining with the global decision model. This is due to the ability of the global decision model to

leverage the information of previously referred entities, thus making full use of the column consistency information when disambiguating entities.

Table 2. The effect of the global decision model for entity linking in web tables.

Methods	Dataset-Wu		Dataset-Xie	
	Micro Acc.	Macro Acc.	Micro Acc.	Macro Acc.
Rep-Based Model	0.870	0.857	0.879	0.861
Rep-Based + global Model	**0.891**	**0.876**	**0.897**	**0.882**
Int-Based Model	0.868	0.859	0.880	0.858
Int-Based + global Model	**0.890**	**0.876**	**0.899**	**0.881**
JHSTabEL(local)	0.877	0.866	0.885	0.865
JHSTabEL	**0.898**	**0.882**	**0.907**	**0.889**

Influence of Ranking Mentions. In this part, we test whether ranking the ns before feeding them into the global model helps to disambiguate entities in web tables. Firstly, we input the mentions directly into our global model in the order they appear in the columns. Secondly, we use a bi-directional LSTM (Bi-LSTM) to consider both previous and following entities in the same column in order. Finally, we compare these two models with our proposed model which adopts ranking mentions. As the results showed in Fig. 6, our model with ranking mentions achieves the best results on both datasets. Comparing two models that do not use ranking, the model with Bi-LSTM performs only slightly better than the model with LSTM. Although the Bi-LSTM can consider the information of the previous and following entities, it may introduce more noise at the same time. However, our proposed model with ranking mentions allows us to utilize the information of easily disambiguated mentions to help the disambiguation of other mentions, so that we can get better disambiguating performance.
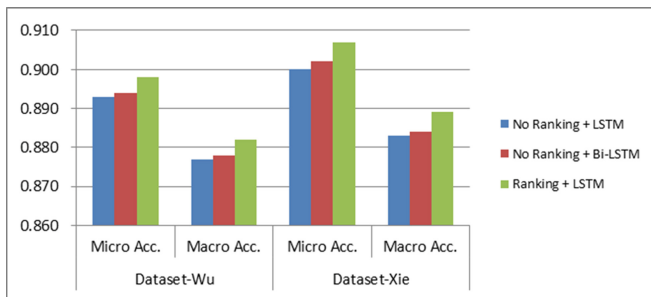
Fig. 6. The influence of ranking mentions for global decision model.

5   Conclusion

In this paper we propose a table entity linking model that takes the advantage of the semantic information from different aspects and jointly disambiguates the mentions in web tables. The combination of the different semantic signals can produce better representations for the mentions and candidate entities. By leveraging information from previously referred entities, we can make full use of column consistency to disambiguate mentions. The Comparison with baselines shows that our model outperforms the state-of-the-art solutions, and the experiments on variants of our model also indicate the substantial benefits of the semantic matching models, mention ranking and global decision model. For the future work, we intend to automatically determine whether the content in table cells should be linked to knowledge bases, since such un-linkable information, such as numbers and long sentences, are common in tables.

References

1. Cafarella, M.J., Halevy, A.Y., Wang, D.Z., Wu, E., Zhang, Y.: WebTables: exploring the power of tables on the web. PVLDB **1**(1), 538–549 (2008)
2. Lehmberg, O., Ritze, D., Meusel, R., Bizer, C.: A large public corpus of web tables containing time and context metadata. In: WWW (Companion Volume) 2016, pp. 75–76 (2013)
3. Bhagavatula, C.S., Noraset, T., Downey, D.: TabEL: entity linking in web tables. In: Arenas, M., et al. (eds.) ISWC 2015. LNCS, vol. 9366, pp. 425–441. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25007-6_25
4. Wu, T., Yan, S., Piao, Z., Xu, L., Wang, R., Qi, G.: Entity linking in web tables with multiple linked knowledge bases. In: Li, Y.-F., et al. (eds.) JIST 2016. LNCS, vol. 10055, pp. 239–253. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50112-3_18
5. Luo, X., Luo, K., Chen, X., Zhu, K.Q.: Cross-lingual entity linking for web tables. In: AAAI, pp. 362–369 (2018)
6. Mulwad, V., Finin, T., Joshi, A.: Semantic message passing for generating linked data from tables. In: Alani, H., et al. (eds.) ISWC 2013. LNCS, vol. 8218, pp. 363–378. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41335-3_23
7. Mikolov, T., Chen, K., Corrado, G., Dean, J: Efficient estimation of word representations in vector space. In: ICLR (Workshop Poster) (2013)
8. Cafarella, M.J., et al.: Ten years of webtables. PVLDB **11**(12), 2140–2149 (2018)
9. Efthymiou, V., Hassanzadeh, O., Rodriguez-Muro, M., Christophides, V.: Matching web tables with knowledge base entities: from entity lookups to entity embeddings. In: d'Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10587, pp. 260–277. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68288-4_16
10. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: AAAI, pp. 2786–2792 (2016)

11. Xiong, C., Dai, Z., Callan, J., Liu, Z., Power, R.: End-to-End neural ad-hoc ranking with kernel pooling. In: SIGIR, pp. 55–64 (2017)
12. Dai, Z., Xiong, C., Callan, J., Liu, Z.: Convolutional neural networks for soft-matching n-grams in ad-hoc search. In: WSDM, pp. 126–134 (2018)
13. Limaye, G., Sarawagi, S., Chakrabarti, S.: Annotating and searching web tables using entities, types and relationships. PVLDB **3**(1–2), 1338–1347 (2010)
14. Fang, Z., Cao, Y., Li, Q., Zhang, D., Zhang, Z., Liu, Y.: Joint entity linking with deep reinforcement learning. In: WWW, pp. 438–447 (2019)
15. Shen, W., Wang, J., Luo, P., Wang, M.: Liege: link entities in web lists with knowledge base. In: SIGKDD, pp. 1424–1432 (2012)
16. Syed, Z., Finin, T., Mulwad, V., Joshi, A.: Exploiting a Web of semantic data for interpreting tables. In: Proceedings of the Second Web Science Conference (2010)