

Cross-Lingual Entity Linking for Web Tables

Xusheng Luo, Kangqi Luo, Xianyang Chen, Kenny Q. Zhu*

Department of Computer Science and Engineering

Shanghai Jiao Tong University

800 Dongchuan Road, Shanghai, China 200240

{freefish_6174, luokangqi, st_tommy}@sjtu.edu.cn, kzhu@cs.sjtu.edu.cn

Abstract

This paper studies the problem of linking string mentions from web tables in one language to the corresponding named entities in a knowledge base written in another language, which we call the cross-lingual table linking task. We present a joint statistical model to simultaneously link all mentions that appear in one table. The framework is based on neural networks, aiming to bridge the language gap by vector space transformation and a coherence feature that captures the correlations between entities in one table. Experimental results report that our approach improves the accuracy of cross-lingual table linking by a relative gain of 12.1%. Detailed analysis of our approach also shows a positive and important gain brought by the joint framework and coherence feature.¹

Introduction

The World Wide Web is endowed with billions of HTML tables, i.e. web tables (Cafarella et al. 2008; Wang et al. 2012), which carry valuable structured information. To enable machines to understand and process such tables (Wang et al. 2012), the first step is to link the surface mentions of the entities in the tables to a standard lexicon or knowledge base, such as Wikipedia, which uniquely identifies entities. This task is known as entity linking in web tables (Bhagavatula, Noraset, and Downey 2015; Wu et al. 2016). In this paper, we also call it “table linking”.

Existing work has been focused on entity linking for web tables in English (Bhagavatula, Noraset, and Downey 2015; Limaye, Sarawagi, and Chakrabarti 2010), or mono-lingual table linking. However, when it comes to linking web tables in other languages, the corresponding non-English knowledge bases are often not comprehensive enough to cover all the entity mentions in the tables at hand. The Chinese Wikipedia, for instance, is just about 1/6 the size of its English counterpart, in terms of the number of entities (articles). This motivates us to link non-English web tables to English knowledge base, in a novel process that we call *cross-lingual table linking*. For example, the movie “邮差

” (Postman) in a Chinese table in Figure 1 is not included in the Chinese Wikipedia but available in the English version. Thus, we can link “邮差” to “Il Postino: The Postman” in the English Wikipedia.

Another important motivation for cross-lingual table linking is to help enrich the facts in the target knowledge base. Knowledge bases in English, albeit larger and better structured than those in other languages, may contain long-tail entities. These are entities associated with very few attributes or relations in the KB, such as Chinese movies or celebrities, since such information is often ignored by the English-speaking Wiki contributors. On the other hand, non-English web tables may be a rich source of semantic relationships among these rare entities. For example, the table in Figure 1 contains the relationship between movies and their countries of origin. “线人”(informant) is a Chinese movie that exists in English Wikipedia (“The_Stool_Piegon_(2010_film)”) and hence Freebase, but it misses a property *film_country*. Now if we can link the mentions in the table to the correct entities in English Wikipedia, then it is easy to infer that the *film_country* property of “The_Stool_Piegon_(2010_film)” is China, thus discovering a new fact.

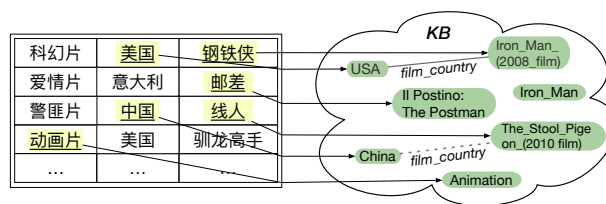


Figure 1: Example of cross-lingual table linking from Chinese to English.

In this paper, we attempt to solve the cross-lingual table linking problem *without* using any non-English knowledge bases. To the best of our knowledge, this is the first attempt that attacks the cross-lingual table linking problem.

There are two naive approaches to accomplish this cross-lingual table linking task. In the first approach, one can use any of the mono-lingual table linking techniques developed thus far to first link the entities to a knowledge base of that language, and then link to the English knowledge base

via inter-language links (Tsai and Roth 2016). For example, Wikipedia provides such inter-language links. This approach may not work because i) the non-English knowledge base may not have all the entities in tables; and ii) many non-English knowledge sources provide no inter-language links.

In the second approach, one can directly translate all the entity names in the non-English web table into English, and then use the mono-lingual table linking techniques to link to an English knowledge base (McNamee et al. 2011). This two-step approach is also not effective because it is analogous to a distant supervised learning, where the association between non-English names and English entities are not directly available for training. If the translation is wrong, the error will propagate in the following linking steps.

In any approach to entity linking (mono- or cross-lingual), a necessary step is to generate a set of candidate entities (Tsai and Roth 2016; McNamee et al. 2011; Bhagavatula, Noraset, and Downey 2015; Wu et al. 2016), and then the problem is transformed to a ranking problem, which aims to pick the entity that is most similar to the mention in the table. The major technical challenge of our task is since the source mention and the target entity come from two different languages, their feature representations are naturally incompatible. To make matters worse, tables offer very limited context for disambiguating a mention in the first place.

We thus propose a neural network based joint model for cross-lingual table linking. We embed mention, context and entity in a continuous vector space to capture their semantics. Further, we employ a linear transformation between vector spaces of two languages. For each table, we link all the mentions simultaneously, so as to fully utilize the relationships among entities in the same row or column. We encode these correlations as a *coherence* feature in the model. Furthermore, we design a pairwise ranking loss function for parameter learning and propose an iterative prediction algorithm to link new tables.

The contribution of this paper is summarized below.

- We are the first to define the problem of cross-lingual entity linking for web tables (Section Problem);
- We present a novel neural network based joint model which effectively captures the rich semantics of mention table and referent entity table simultaneously. Based on that, we bridge the gap between different languages in this task (Section Approach);
- We propose a coherence feature in the joint linking model which captures the correlation of entities appearing in the same table and improves the linking accuracy (Subsection Coherence Feature in Approach);
- The framework significantly outperforms several baseline methods, with an accuracy of 62.9%. (Section Experiments).

Problem Definition

The input mention table, denoted by X , is a matrix of surface forms with R rows and C columns. Each mention x_{ij} is represented by a sequence of words written in language L_1 (e.g., Chinese). Given a knowledge base K containing a set of entities e written in language L_2 (e.g., English), our task

is to find the corresponding entity table E , such that each entity $e_{ij} \in K$ correctly disambiguates the surface form x_{ij} .

In practice, many mention tables contain unlinkable cells, such as numbers, dates, times or emerging entities non-existent in the knowledge base. There are some existing works that deal with the identification of such numerical or temporal entities in web tables (Ibrahim, Riedewald, and Weikum 2016). In this paper, we will not focus on judging whether a cell is linkable or not. Let P denote a set of indices (i, j) indicating the position of all linkable cells in a table, and we assume that all linkable positions P are provided along with the mention table X in both training and testing datasets.

Traditional entity linking approaches usually formulate the task by defining a scoring function $S(x, e)$, which measures the relevance between a mention x and the target entity e . Such techniques perform entity linking of each cell independently, but the interaction between neighbourhood cells ignored in the scenario of table inputs. To incorporate the coherence information between target entities in the table, we define another scoring function for the table linking task, defined as follows:

$$\hat{E} = \operatorname{argmax}_{E \in \text{GEN}(X)} S(X, E), \quad (1)$$

where $\text{GEN}(X)$ denotes the set of all candidate entity tables, and the function measures the overall relevance score between the whole input table and a candidate entity table.

Approach

In this section, we describe our joint model for cross-lingual table linking. Figure 2 gives a general view of the model. The reason we call it a “joint model” is that the input of neural network is a mention table X containing all the cells to be linked, together with one candidate entity table E , and the output stands for the relevance score $S(X, E)$.

Specifically, we first generate candidate entities of each single mention, then we learn two different features: the **mention** feature and **context** feature derived from the mention-entity embedding pairs of the table. To make different representations from two language spaces compatible, we utilize a bilingual translation matrix to transform the vector representation from Chinese to English. Meanwhile, we learn a third feature called **coherence** feature only from the candidate entity table. Finally, we discuss the prediction and parameter learning step of this task.

Candidate Entity Generation

We generate candidate English entities for each mention represented in Chinese. Without a reliable Chinese knowledge base as the bridge, we use translation tools to produce a set of possible translations of the given mention. Afterwards, we use several heuristic rules to obtain candidate English entities. Possible candidate entities consist of: 1) exact match of any mention translation; 2) anchor entities of any mention translation in knowledge base; 3) fuzzy match (e.g., edit distance) of any mention translation. Take the Chinese mention “疑犯追踪” as an example, it can be translated to “person

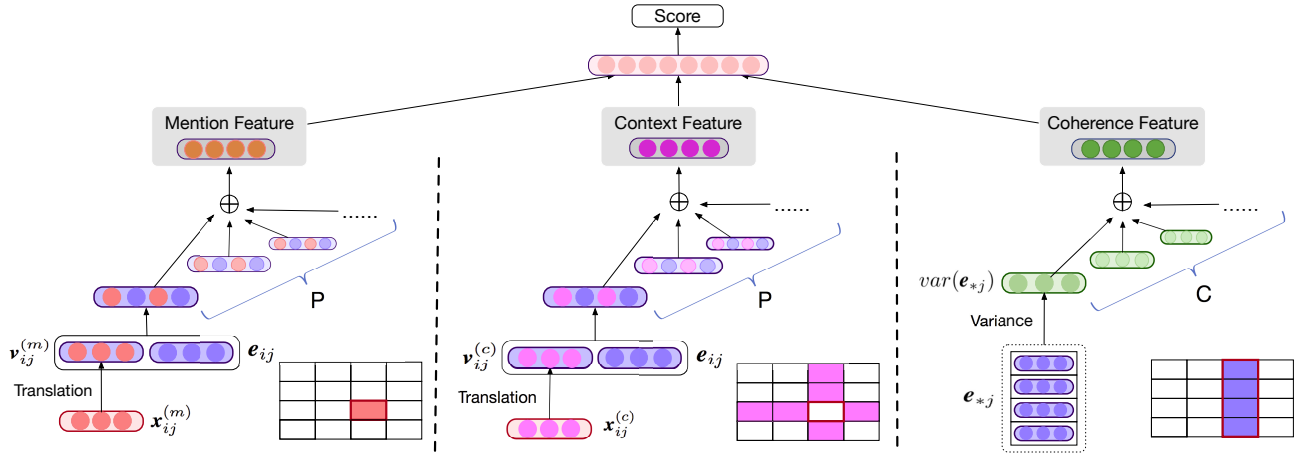


Figure 2: Overview of proposed neural network based joint model.

of interest” or “suspect tracking”, depending on what translation tool to use. The corresponding candidate entity set would contain entities such as “person of interest”, “person of interest (tv series)” or “suspect (1987 film)”.

Embedding and Translation Module

Let $\mathbf{x}^{(m)}$ denote the mention embedding of the surface form x , and \mathbf{e} denote the entity embedding of a candidate e . Typically, a mention contains up to three words, thus we simply represent $\mathbf{x}^{(m)}$ as the average embedding of the contained words it contains. We train word embeddings and entity embeddings on two corpus of different languages separately.

The vector spaces of embeddings in different languages are naturally incompatible, and it’s hard for us to directly compare or calculate them. To tackle this problem, we employ a bilingual translation layer to map embeddings from one language space to another. Taking the mention embedding $\mathbf{x}^{(m)}$ in Chinese, the layers translates it into an English mention embedding $\mathbf{v}^{(m)}$ through linear transformation: $\mathbf{v}^{(m)} = W_t \mathbf{x}^{(m)} + \mathbf{b}_t$, where W_t is the translation matrix and \mathbf{b}_t is the bias, both of which are model parameters and will be updated during training step.

In addition, we pre-train the translation parameters by leveraging a small number of bilingual word pairs ($w^{(ch)}$, $w^{(en)}$), or call them translation seeds. The loss function of pre-train step is defined as follows:

$$L(W_t, \mathbf{b}_t) = \sum_i \|W_t \mathbf{w}_i^{(ch)} + \mathbf{b}_t - \mathbf{w}_i^{(en)}\|_2. \quad (2)$$

Refer to Section Implementation Details and Experiments for detailed information of the embedding initialization and translation pre-train.

Mention and Context Feature

As shown in Figure 2, the **mention** and **context** feature represents the relevance or compatibility between the mention table X and the entity table E . Both features aggregate the individual features of each cell, and thus share a similar neural network structure.

We first introduce the **mention** feature. For the surface form x_{ij} , we concatenate the translated embedding $\mathbf{v}_{ij}^{(m)}$ with the entity embedding \mathbf{e}_{ij} (Socher et al. 2013; 2015), then feed into a fully connected layer, obtaining the hidden feature between x_{ij} and \mathbf{e}_{ij} at mention level. We apply vector averaging over all cells to be linked and finally produce the hidden mention feature $\mathbf{h}^{(m)}$ between the whole mention and entity table. We formulate the steps as follows:

$$\begin{aligned} \mathbf{f}_{ij}^{(m)} &= \text{ReLU}(W^{(m)}[\mathbf{v}_{ij}^{(m)}; \mathbf{e}_{ij}] + \mathbf{b}^{(m)}) \\ \mathbf{h}^{(m)} &= \frac{1}{|P|} \sum_{(i,j) \in P} \mathbf{f}_{ij}^{(m)}, \end{aligned} \quad (3)$$

where $W^{(m)}$ and $\mathbf{b}^{(m)}$ are model parameters.

The **context** feature follows the similar idea. Instead of using the surface form x_{ij} itself, mentions in the same row or column (excluding itself) contain strong relatedness and hence be regarded as the surrounding context. In this way, we define the context embedding $\mathbf{x}_{ij}^{(c)}$ as the average mention embedding of those surrounding cells:

$$\mathbf{x}_{ij}^{(c)} = \frac{1}{|R+C-1|} \left(\sum_{(i,k), k \neq j} \mathbf{x}_{ik}^{(m)} + \sum_{(k,j), k \neq i} \mathbf{x}_{kj}^{(m)} \right). \quad (4)$$

After applying the translation module, the context embedding $\mathbf{v}_{ij}^{(c)}$ of each cell is used to generate the hidden context feature of the mention-entity table pair, denoted by $\mathbf{h}^{(c)}$. The calculation is almost the same as Eq. (3), but just replacing all the mention embeddings by the context ones. By learning the mention and context features, we can capture a general sense of semantic relatedness of all mention-entity pairs from two tables.

Coherence Feature

The previous two features aim at encoding the relevance or compatibility of the mention-entity table pair. On the other hand, the inner relationship of entities in the correct linked table is also valuable. The intuition is that entities in the

same column (row) tend to own the same type, in which case, lead to similar vector representations. In the example in Figure 1, target entities of the three columns represent “university”, “street” and “city”, respectively. Therefore, we propose a third feature, which captures such correlations among entities of same column.

There have been several works (Eberius et al. 2015; Nishida et al. 2017) on table type classification, which identifies the display form of web tables. In this paper, we mainly focus on tables with type “Vertical Relational” (VR) (Nishida et al. 2017), which just look like Figure 1, where entities in the same column have the same type. Most of web tables can be transformed into this type after classification, which is outside the scope of this paper.

We calculate the element-wise variance for all entity vectors in the same column to get a coherence vector for that column. The average among all columns is the hidden coherence feature $\mathbf{h}^{(coh)}$ of whole entity table:

$$\mathbf{h}^{(coh)} = \frac{1}{C} \sum_j \text{var}(\{e_{ij} | (i, j) \in P\}), \quad (5)$$

where $\text{var}(\cdot)$ calculates element-wise variance for a bunch of input vectors. The coherence feature captures how self-organized the candidate entities are, which complements the previous mention and context features.

Training and Prediction

As mentioned before, we handle the table linking task by defining a scoring function of the mention-entity table pair. To fulfill this, the previous mention, context and coherence features are fed into a two-layer fully connected network, and the final output indicates the final relevance score:

$$\begin{aligned} \mathbf{h}_{out} &= \text{ReLU}(W_{out}[\mathbf{h}^{(m)}; \mathbf{h}^{(c)}; \mathbf{h}^{(coh)}] + \mathbf{b}_{out}) \\ S(X, E) &= \mathbf{u} \cdot \mathbf{h}_{out}, \end{aligned} \quad (6)$$

where W_{out} , \mathbf{b}_{out} and \mathbf{u} are model parameters.

For each mention table in the training set, there will be one positive gold entity table, and several negative (corrupted) entity tables. The negative table is generated automatically from the gold table as follows: we first randomly select some cells to be corrupted, and then replace the entities in those cells by a random entity from the corresponding cell of a candidate table.

There are two possible optimization strategies during training: hinge loss and a pairwise ranking model. For hinge loss, we try to maximize the difference between positive and negative entity tables. For pairwise ranking model, every pair of candidate tables are compared: the table with more correctly linked entities is ranked higher than the other one in the pair. Here we adopt RankNet (Burgess 2010) with Adam stochastic optimizer (Kingma and Ba 2014) as our implementation.

At the time of prediction, ideally we must enumerate all the candidate entity tables to get the global optimal. However, the number of candidate entity tables grow exponentially with respect to the number of cells to be linked, rendering such approach intractable. To this end, we use a local-

Algorithm 1 Local-Search Descent Prediction

Input: Mention table X , linking position P , initial entity table E_0 , candidate generator $Cand(\cdot)$, scoring function $S(\cdot, \cdot)$

Output: Entity table E

```

1: procedure PREDICT( $X, E_0, Cand, S$ )
2:    $E \leftarrow E_0$ 
3:    $s_{max} \leftarrow S(X, E_0)$ 
4:   repeat
5:     Shuffle  $P$ 
6:     for  $(i, j)$  in  $P$  do
7:        $E' \leftarrow E$ 
8:       for  $ent$  in  $Cand(x_{ij})$  do
9:          $e'_{ij} \leftarrow ent$ 
10:         $s' \leftarrow S(X, E')$ 
11:        if  $s' > s_{max}$  then
12:           $e_{ij} \leftarrow ent$ 
13:           $s_{max} \leftarrow s'$ 
14:   until  $s_{max}$  converges
15:   return  $T_E$ 

```

search descent algorithm to approximate the optimal solution. As shown in Algorithm 1, E_0 is the initial candidate entity table, where each cell is filled with the most possible candidate entity produced by the generator $Cand(\cdot)$, and S is the learned scoring function. The predicting step works iteratively. For each round, all cells are visited one-by-one in random order (line 6), and for each cell, the algorithm try to replace the current entity by the one at the local optima, and updates the output table (line 12). The iteration continues until no replacement can improve the relevance score.

Implementation Details

We show the detailed implementations of our model, which include candidate generation, translation model pre-train, and parameter tuning.

Candidate Generation: we first use several translation tools provided by Google², Baidu³ and Tencent⁴. After retrieving English translations, we use several heuristics to find candidate English entities in Wikipedia for each Chinese mention. We add all anchor entities whose anchor text exactly match the each translation with confidence 1.0. Then we remove all the stop words from the translations and from the anchor texts of Wikipedia, and compute the Jaccard similarity between the two modified forms in order to fetch more candidate entities. We use the Jaccard similarity score as the confidence score here.

Translation Model Pre-Train: we collect a bilingual lexicon of common words using Bing Translate API⁵, containing 91,346 translation pairs at word level. Each pair has a confidence score ranging from 0 to 1. We remove the pairs with score less than 0.5, and further select those pairs in which both the Chinese and English word perfectly match the name of an article in Wikipedia. In total 3,655 translation pairs are picked as our pre-train dataset.

²<http://translate.google.cn>

³<http://fanyi.baidu.com>

⁴<http://fanyi.qq.com>

⁵<http://www.bing.com/translator>

Parameter Tuning:

- The size of candidates per mention (denoted by N_{cand}) is in the range of $\{1, 3, 5, 10, 20, 30, 40, 50\}$,
- The number of negative entity tables per mention table (denoted by N_{tab}) is in $\{9, 19, 49, 99\}$,
- The dimension of cell, context and overall features (d_{cell} , d_{cont} and d_{out}) are in $\{20, 50, 100, 200\}$,
- The learning rate η is in $\{0.0002, 0.0005, 0.001\}$,
- We apply dropout layers (Srivastava et al. 2014) on each hidden feature vector to avoid overfitting. The keep probability p of dropout layers is in $\{0.5, 0.6, 0.7, 0.8, 0.9\}$.

Experiments

In this section, we introduce the datasets and previous state-of-the-art systems for comparison in our experiments, and explain how to adapt a mono-lingual entity linker into our scenario. We show the end-to-end results of all the systems on cross-lingual and mono-lingual dataset, and perform ablation experiments to investigate the importance of different components used in the whole task.

Experimental Setup

Wikipedia and Word Embeddings: we use the Feb. 2017 dump of English⁶ and Chinese⁷ Wikipedia as the text corpora for training word and entity embeddings. The dumps contain 5,346,897 English and 919,696 Chinese articles (entities). For the purpose of entity embedding, all the entities occurred in anchor texts are regarded as special words. E.g., the anchor text “Rockets” in the sentence “the Rockets All-Star player James Harden ...” is replace by the special word “[[Houston_Rockets]]” as the entry of the English entity. We adopt Word2Vec (Mikolov et al. 2013) to learn the initial embeddings from both corpus respectively, the embedding dimension is set to 100.

Table Linking Dataset: our cross-lingual table linking dataset consists of 150 web tables with Chinese mentions and linked English Wiki articles. The original Chinese tables are created by Wu et al. (2016), which contains 123 tables extracted from Chinese Wikipedia, and each mention is labeled by its corresponding Chinese Wiki article. We collect another 30 Chinese tables with similar size from Web and transform all the Chinese entities into English via inter-language links of Wikipedia, producing the labeled English entities for 81% of the entire mentions. In addition, we discard long-tail tables, in which the dimension of the table or the number of labeled English entities is too small. In total, we collected 3818 mentions from 150 tables, with 2883 linkable positions (19.22 per table). We randomly split the dataset⁸ into training / validation / testing sets (80 : 20 : 50 tables).

⁶<https://dumps.wikimedia.org/enwiki/>

⁷<https://dumps.wikimedia.org/zhwiki/>

⁸The dataset is available at <https://adapt.seiee.sjtu.edu.cn/table>

State-of-the-art Comparisons

Since there are no previous work that directly handle the cross-lingual table linking, we select comparison systems from two perspectives. The first perspective is mono-lingual table linking, we compare with Bhagavatula et al. (2015) and Wu et al. (2016). We call their systems $TabEL_B$ and $TabEL_W$ in short. In order to make a fair comparison in our bilingual scenario, we convert each mention into the most likely English translation, then run the mono-lingual models on these translated English tables.

The second branch is cross-lingual text linking, we compare with Zhang et al. 2013 (2013), a bilingual-LDA based method, called $TextEL$. In this case, we traverse each mention in row order and flatten the whole table into a word sequence, and mark the word intervals for mentions to be linked. By turning the table into an unstructured piece of text, $TextEL$ is able to learn more flexible context information. However, it may be hard to capture the correlation of entities in the same column.

Evaluation of Candidate Generation

In this part, we investigate the translated English mentions from Chinese table inputs. As described in Section Implementation Details, English mentions are derived from multiple resources. Compare with different combination of resource, we evaluate the quality by measuring the proportion of cells that the correct entity appears in the top- n candidates (Hits@ n).

From the results in Table 1, we observe that ensembling multiple translation resources is able to discover more correct entities without bringing too many noisy candidates.

Table 1: Hits@ n results on candidate entity generation

| Resources | n=1 | n=5 | n=10 |
|-----------|--------------|--------------|--------------|
| Google | 0.463 | 0.585 | 0.596 |
| Baidu | 0.542 | 0.669 | 0.684 |
| Tencent | 0.394 | 0.510 | 0.522 |
| All Trans | 0.558 | 0.708 | 0.726 |

End-to-End Results

Now we perform the cross-lingual table linking experiment and compare with previous table linking and text linking systems. To be consistent with state-of-the-art systems, we report Micro Accuracy and Macro Accuracy as the evaluation metrics. Micro Accuracy is the percentage of correct linked cells over the whole dataset, while Macro Accuracy, defined as average correct ratio over different tables, thus avoiding the bias towards the table with more cells.

Due to both $TabEL_B$ and $TabEL_W$ taking only one English mention per cell as the input, we select Baidu as the best translation tool and apply this setting to all approaches. In addition, we evaluate our approach under the full translating strategy, using either pre-train or without pre-train. For all the variations of our approach, we set $N_{cand} = 30$, $N_{tab} = 49$, $d_{cell} = d_{cont} = 100$, $d_{out} = 200$, $\eta = 0.0002$ and $p = 0.9$ under RankNet optimizer, as reaching the

highest Micro Accuracy in the validation set. For other approaches, we use different N_{cand} , tuning separately.

We report the experimental results in Table 2. For the 4 experiments using Baidu translation only, Our model outperforms the other baseline models, improving the result by up to 12.1%. Our full model even improves the Micro Accuracy by an absolute gain of 0.053, showing the importance of combining multiple translating tools. Besides, the pre-train step also raises the Micro Accuracy by 0.023. Both $TabEL_B$ and $TabEL_W$ suffer from the error propagation problem, because only the top translation is considered, whereas our approach generates candidate entities from multiple translated mentions, which alleviates the error brought by translation.

Table 2: Accuracies on cross-lingual table linking task. All baselines take Baidu as the only translating tool.

| Approach | Micro Acc. | Macro Acc. |
|--------------------------|--------------|--------------|
| $TabEL_B$ | 0.512 | 0.507 |
| $TabEL_W$ | 0.514 | 0.519 |
| $TextEL$ | 0.472 | 0.458 |
| Ours (Baidu Only) | 0.576 | 0.573 |
| Ours (Full, - pre-train) | 0.606 | 0.591 |
| Ours (Full, + pre-train) | 0.629 | 0.614 |

We further investigate how the candidate size of a mention effects the table linking result. When N_{cand} goes larger, the theoretical upper bound of the final result increases, but it’s more difficult for the system to reach the upper bound. For analyzing such tradeoff, Figure 3 shows the Micro Accuracy trend of each approach, and we display upper bound (Hits@ n) in the figure. Our approach is more adaptive to different size of candidates, and can produce promising end-to-end results. $TabEL_B$ also keeps a stable performance with a subtle decreasing, while $TextEL$ drops dramatically, even if the candidate size is smaller than 10. The main reason is that BLDA model is unsupervised, which doesn’t observe any explicit (mention, entity) pair for learning.

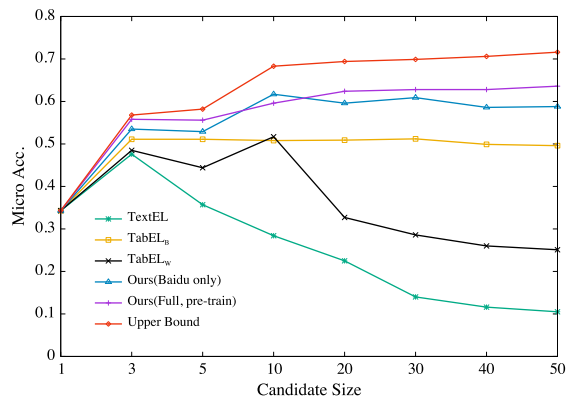


Figure 3: Results of Micro Accuracy by different size of candidates, using Baidu translation only.

To better justify the effectiveness of our model, we perform experiments on mono-lingual scenario. We use the

original table dataset, where target entities are articles in Chinese Wikipedia. Accordingly, we remove the translation layer in our model for handling the mono-lingual task, and all the other settings stay the same. Again we compare with previous table linking systems, where $TabEL_W$ is reported as the state-of-the-art in Chinese table linking task. Experimental results in Table 3 show that our mono-lingual model still outperforms the two baselines, which supports the expressiveness of our NN based joint model for handling general table linking tasks.

Table 3: Accuracies on Chinese mono-lingual table linking.

| Approach | Micro Acc. | Macro Acc. |
|-----------|--------------|--------------|
| $TabEL_B$ | 0.848 | 0.845 |
| $TabEL_W$ | 0.852 | 0.848 |
| Ours-mono | 0.886 | 0.868 |

Ablation Study

In this section, we explore the contributions of the various components of our system.

Feature Variations

We first evaluate table linking results using different feature combinations. As the results shown in Table 4, all features in our model make a positive contribution to the final accuracy. The mention feature is the most important one, since it encodes the most direct information between the mention and the target entity. We observe that when using coherence feature only, a significant decrease in accuracy takes place, largely due to the lack of dominant and direct semantic association between mention-entity pairs. Nevertheless, the coherence feature is complementary to the others, as it aims at discovering the latent correlation in a global perspective, modeling whether different candidate entities in one column are close to each other, for example, sharing the same (or similar) type, even though no explicit type or category information is attached to the entity.

Table 4: Ablation test on validation set.

| Feature Combination | Micro Acc. | Decrease in Acc. (%) |
|---------------------|------------|----------------------|
| Mention Only | 0.604 | 12.7 |
| Context Only | 0.576 | 16.7 |
| Coherence Only | 0.279 | 59.6 |
| Mention + Context | 0.652 | 5.78 |
| Full | 0.692 | 0.00 |

In the example in Figure 1, the mention “钢铁侠” can be linked to either “Iron_Man” (the fictional superhero) or “Iron_Man_(2008_film)” in Wikipedia, while both “驯龙高手” (“How_to_Train_Your_Dragon_(film)”) and “线人” (“The_Stool_Pigeon_(2010_film)”) have less ambiguity. Our model predicts the superhero when using mention + context features only. After applying the coherence feature, the strong correlation between the entities in the same column makes the model bias toward the correct film entity.

Joint Model Versus Non-Joint Model

Now we investigate the effectiveness of the joint framework. Inspired by Sun et al. (2015), we change our joint

scoring function back to the non-joint style, where the coherence module is removed, and the average operation over different cells is no longer needed. As a comparison, we re-run our joint model but with the coherence module removed, and use either hinge loss or RankNet as the optimizer. Table 5 shows Micro Accuracy results on the testing set. We find that when using hinge loss, the non-joint model even outperforms the joint model. We believe that hinge loss is less effective than RankNet in our joint model, because: i) all the negative candidates of each mention are used in the non-joint model; however, in the joint model, some negative candidates are not sampled and hence cannot be observed by the model; ii) hinge loss focuses on the margin between the positive entity table and the nearby negative entity table (with only a few corruptions), thus other negative tables with more corruptions become less effective in the training. It’s worth mentioning that while the non-joint model is more light-weight at run time, the joint model takes only 6 rounds on average for each prediction, which is acceptable in terms of running time.

Table 5: Micro-accuracies on the testing set under different model specifications.

| Model | Optimizer | Coherence | Micro Acc. |
|-----------|------------|-----------|------------|
| Non-Joint | Hinge Loss | N | 0.586 |
| Joint | Hinge Loss | N | 0.574 |
| Joint | RankNet | N | 0.598 |
| Joint | RankNet | Y | 0.629 |

Related Work

Entity linking has been a popular topic in NLP for a long time as it is the basic step for machines to understand natural language and an important procedure of many complex NLP applications such as information retrieval and question answering. Entity linking requires a knowledge base to which entity mentions can be linked, the most popular ones including Freebase (Bollacker et al. 2008), YAGO (Suchanek, Kasneci, and Weikum 2007) and Wikipedia (Cai et al. 2013), where each Wikipedia article is considered as an entity. Due to its fundamental role in many applications, the task of entity linking has attracted a lot of attention, and many shared tasks have been proposed to promote this study (Ji et al. 2010; Cano et al. 2014; Carmel et al. 2014). Similar to our work, Sun et al. (2015) used neural networks for entity linking. They used a Siamese-like network structure, where the mentions and candidates are separately embedded into vector space, and contexts are modeled by a convolution neural network. A cosine-similarity score is output as the score of a <mention, context, candidate> triplet and trained with hinge-loss. On the other hand, our model jointly assigns the mentions simultaneously.

Different from general entity linking tasks, table entity linking focuses only on entries in tables. The interest in web tables was inspired by Cafarella et al. (2008). Muñoz et al. (2014) proposed methods to mine rdf triples from Wikipedia tables, and Sekhavat et al. (2014) proposed methods to enrich a knowledge base by leveraging tabular data

on the Web. These works and other applications involving web tables could all benefit from our table entity linking system. Bhagavatula et al. (2015) argued that models which jointly address entity linking, column type identification and relation extraction rely on the correctness and completeness of KB, thus may adversely affect the performance of entity linking. They also exploited a graphical model, where cells in the same row or column are connected. Graphical models are also used elsewhere (Limaye, Sarawagi, and Chakrabarti 2010; Ibrahim, Riedewald, and Weikum 2016) Wu et al. (2016) constructed a graph of mentions and candidate entities for each query table, then use page rank (Page et al. 1999) to determine the similarity score between mentions and candidates. Besides, they combined multiple knowledge bases in Chinese to enhance the system.

Starting from 2011 the annual TAC KBP Entity Linking Track has been using the multi-language setting (Ji et al. 2010; Ji, Nothman, and Hachey 2014; Ji et al. 2015), where the languages involved are English, Chinese and Spanish. Most methods managed to bridge the language gap through language-independent spaces. Fahrni et al. (2011) presented HITS’ system for cross-lingual entity linking. Their approach consisted of three steps: 1) obtain a language-independent concept-based representation for query documents; 2) disambiguate the entities using an SVM and a graph-based approach; 3) cluster the remaining mentions which were not assigned any KB entity in step 2. Zhang et al. (2011) leveraged a modified version of Latent Dirichlet Allocation, which they call BLDA (Bilingual LDA) and bridged the gap between languages via topic space. Wang et al. (2015) proposed an unsupervised graph-based method which matches a knowledge graph with a graph constructed from mentions and the corresponding candidates of the query document. Tsai et al. (2016) trained a multilingual word and title embeddings and ranked entity candidates using features based on these multilingual embeddings.

Conclusion

To the best of our knowledge, this is the first piece of work that studies the cross-lingual entity linking problem for web tables. We proposed a neural network based joint model that takes advantage of features extracted from a cell, its context and semantic coherence within a table column. Our experiments show the substantial benefits of using the joint model that predicts the links of all cells at once versus a non-joint model that predicts the cells independently. Our best model achieves an accuracy of 63%, for a task that is significantly more challenging than mono-lingual table linking. Possible future work includes the automatic determination of whether a non-numerical string mention in a cell should or should not be linked. We have ignored this problem in this paper but such un-linkable cells are abundant in web tables, too.

References

Bhagavatula, C. S.; Noraset, T.; and Downey, D. 2015. Tabel: entity linking in web tables. In *International Semantic Web Conference*, 425–441. Springer.

- Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD*, 1247–1250. AcM.
- Burges, C. J. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11(23-581):81.
- Cafarella, M. J.; Halevy, A.; Wang, D. Z.; Wu, E.; and Zhang, Y. 2008. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment* 1(1):538–549.
- Cai, Z.; Zhao, K.; Zhu, K. Q.; and Wang, H. 2013. Wikification via link co-occurrence. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, 1087–1096. AcM.
- Cano, A. E.; Rizzo, G.; Varga, A.; Rowe, M.; Stankovic, M.; and Dadzie, A.-S. 2014. microposts2014 neel challenge: Measuring the performance of entity linking systems in social streams. *Proc. of the Microposts2014 NEEL Challenge*.
- Carmel, D.; Chang, M.-W.; Gabrilovich, E.; Hsu, B.-J. P.; and Wang, K. 2014. Erd’14: entity recognition and disambiguation challenge. In *ACM SIGIR Forum*, volume 48, 63–77. AcM.
- Eberius, J.; Braunschweig, K.; Hentsch, M.; Thiele, M.; Ahmadov, A.; and Lehner, W. 2015. Building the dresden web table corpus: A classification approach. In *Big Data Computing (BDC), 2015 IEEE/ACM 2nd International Symposium on*, 41–50. IEEE.
- Fahrni, A., and Strube, M. 2011. Hits’ cross-lingual entity linking system at tac 2011: One model for all languages. In *TAC*.
- Ibrahim, Y.; Riedewald, M.; and Weikum, G. 2016. Making sense of entities and quantities in web tables. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 1703–1712. AcM.
- Ji, H.; Grishman, R.; Dang, H. T.; Griffith, K.; and Ellis, J. 2010. Overview of the tac 2010 knowledge base population track. In *Third Text Analysis Conference (TAC 2010)*, volume 3, 3–3.
- Ji, H.; Nothman, J.; Hachey, B.; and Florian, R. 2015. Overview of tac-kbp2015 tri-lingual entity discovery and linking. In *Proceedings of the Eighth Text Analysis Conference (TAC2015)*.
- Ji, H.; Nothman, J.; and Hachey, B. 2014. Overview of tac-kbp2014 entity discovery and linking tasks. In *Proc. Text Analysis Conference (TAC2014)*, 1333–1339.
- Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Limaye, G.; Sarawagi, S.; and Chakrabarti, S. 2010. Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB Endowment* 3(1-2):1338–1347.
- McNamee, P.; Mayfield, J.; Lawrie, D.; Oard, D. W.; and Doermann, D. S. 2011. Cross-language entity linking. In *IJCNLP*, 255–263.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.
- Muñoz, E.; Hogan, A.; and Mileo, A. 2014. Using linked data to mine rdf from wikipedia’s tables. In *Proceedings of the 7th ACM international conference on Web search and data mining*, 533–542. AcM.
- Nishida, K.; Sadamitsu, K.; Higashinaka, R.; and Matsuo, Y. 2017. Understanding the semantic structures of tables with a hybrid deep neural network architecture. In *AAAI*, 168–174.
- Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.
- Sekhavat, Y. A.; Di Paolo, F.; Barbosa, D.; and Merialdo, P. 2014. Knowledge base augmentation using tabular data. In *LDOW*.
- Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, 926–934.
- Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2015. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*.
- Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research* 15(1):1929–1958.
- Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, 697–706. AcM.
- Sun, Y.; Lin, L.; Tang, D.; Yang, N.; Ji, Z.; and Wang, X. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *IJCAI*, 1333–1339.
- Tsai, C.-T., and Roth, D. 2016. Cross-lingual wikification using multilingual embeddings. In *Proceedings of NAACL-HLT*, 589–598.
- Wang, J.; Wang, H.; Wang, Z.; and Zhu, K. Q. 2012. Understanding tables on the web. In *International Conference on Conceptual Modeling*, 141–155. Springer.
- Wang, H.; Zheng, J.; Ma, X.; Fox, P.; and Ji, H. 2015. Language and domain independent entity linking with quantified collective validation. In *EMNLP*, 695–704.
- Wu, T.; Yan, S.; Piao, Z.; Xu, L.; Wang, R.; and Qi, G. 2016. Entity linking in web tables with multiple linked knowledge bases. In *Joint International Semantic Technology Conference*, 239–253. Springer.
- Zhang, T.; Liu, K.; Zhao, J.; et al. 2013. Cross lingual entity linking with bilingual topic model. In *IJCAI*.
- Zhang, W.; Su, J.; and Tan, C. L. 2011. A wikipedia-lda model for entity linking with batch size changing instance selection. In *IJCNLP*, 562–570.