Cross-Lingual Entity Linking for Web Tables

ABSTRACT

This paper studies the problem of linking string mentions of entity names 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 link all mentions that appear in one table simultaneously. The framework is based on neural networks, aiming to bridge the language gap by vector space transformation and a coherence feature is proposed to capture the correlations between entities in one table. Experimental results show that our approach outperforms all baseline methods.

KEYWORDS

Entity linking, Web table, Joint model

ACM Reference format:

. 2017. Cross-Lingual Entity Linking for Web Tables. In *Proceedings of International Conference on Information and Knowledgement Management, Singapore, Nov 2017 (CIKM' 2017),* 9 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

The World Wide Web is endowed with billions of HTML tables (i.e. Web tables [4, 33]), which carry valuable structured information. To enable machines to process such tables, or to understand the web tables, the first step is to link the entities mentioned 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 [1, 34]. In this paper, we also call it "table linking". Because most of the existing well-known knowledge bases maintain entries primarily in English, much of the table linking work has been focused on web tables in English [1, 17], hence the techniques developed are for mono-lingual scenarios only. In reality, the non-English web also possesses many web tables with equally precious information that would benefit not only people speaking that particular language, but also the English world. For example, information extracted from a table about Chinese celebrities can be used to enrich knowledge in Freebase or IMDB. On the other hand, since knowledge bases in English are more comprehensive and accurate than that of other languages, entity linking systems in foreign languages often fail to identify an entity due to incompleteness of KB in their languages. Incorporating English entities into the target KB can improve the recall of entity linking systems in such cases. Kenny: The above reason seems not compelling enough to make the following claim. Therefore, there is a substantial need to link entities in non-English web tables to English knowledge bases. Figure 1 depicts such a scenario. After cross-lingual table linking, "上海" in the second row is linked to its corresponding real world entity *Shanghai* in a KB (e.g., Freebase). Kenny: Change to an example that shows rare entites in Chienese but available in English KB.

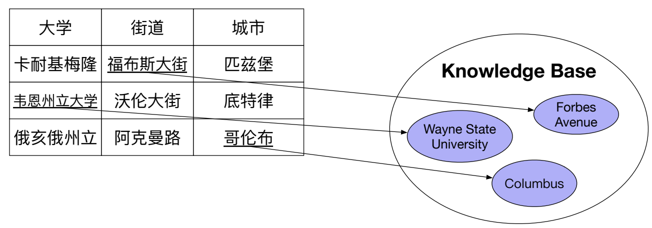

Figure 1: Example of Cross-lingual Table Linking from Chinese to English

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 or culture, and then link to the English knowledge base via inter-language links [31]. For example, Wikipedia is a multi-lingual knowledge base that provides such inter-language links. The problem with this approach is that the non-English knowledge base may not be comprehensive enough to carry all the entities in the web tables in question. The Chinese Wikipedia, for instance, is only about 1/6 the size of the English Wikipedia, by the number of articles. Furthermore, 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 [18]. 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 not correct, the error will propagate in the following linking steps.

In this paper, we attempt to solve the cross-lingual table linking problem without the use of any non-English knowledge bases. That is, our goal is to link the mentions in the non-English table directly to an entity in the English knowledge base. The advantage of this is we do not discard any information of non-English mentions so that our model has the ability to tolerate the error caused by translation. To the best of our knowledge, this is the first attempt that attacks the cross-lingual table linking problem. In all entity linking tasks (mono-lingual or cross-lingual), a necessary step in any approach is to generate a set of candidate entities [1, 18, 31, 34], 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

ACM ISBN 123-4567-24-567/08/06...$15.00
https://doi.org/10.475/123_4

table. The similarity computation requires the feature representation for a mention to be compatible with the representation for an entity. The major technical challenge of our task is since the source mention and the target entity come from two difference languages, their feature representations are naturally incompatible and incomparable. To make matters worse, tables offer very limited context for disambiguating a mention in the first place.

We propose a joint model based on neural network for cross-lingual table linking. We embed mention, context and entity in continuous vector space to capture their deep semantics. Based on that, we employ a linear transformation to do a primary translation between vector spaces of two languages. For each table, we link all the mentions simultaneously, in order to fully utilize the relationships among entities in the same row or column. We encode this 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 can be summarized as follows.

- We formally define the problem of cross-lingual entity linking for Web tables (Section 2);
- 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 3);
- 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 (Section 3.5);
- The framework significantly outperforms several baseline methods, with an accuracy of 63.6%. (Section 5).

2 PROBLEM DEFINITION

Definition 2.1. A *mention table* T_M is a tuple $(\mathbf{m}, P, L_1)$, where: $\mathbf{m}$ is a matrix of mentions and $\mathbf{m}_{ij}$ represents a specific cell in table at i^{th} row and j^{th} column; P is a set of indices $\langle i, j\rangle$ indicating Position of cells needs to be linked; L_1 is the language in which table mentions $\mathbf{m}$ are written.

Definition 2.2. A *Knowledge Base* K is a tuple (E, L_2), where: E is a finite set of unique entities in KB; L_2 is the language in which KB is written.

Definition 2.3. An *entity table* T_E is a tuple $(\mathbf{e}, P, L_2)$, where: $\mathbf{e}$ is a matrix of entities and $\mathbf{e}_{ij} \in E$ represents a specific cell in table at i^{th} row and j^{th} column; P is a set of indices $\langle i, j\rangle$ indicating Position of cells have been linked; L_2 is the language in which linked entities $\mathbf{e}$ are written.

Definition 2.4. Cross-lingual table linking seeks to find a mapping $\mathcal{F}$ between a mention table T_M and an entity table T_E, so that each mention $\mathbf{m}_{ij}$ where $\langle i, j\rangle \in M$ is linked to a corresponding entity $\mathbf{e}_{ij}$.

In practice, many web tables contain unlinkable entities such as numbers, dates and times. There are some existing work that deals with the identification of such numerical or temporal entities in Web Tables[28]. In this paper, we assume that all the web tables have been preprocessed to remove all unlinkable columns (or rows), and as a result, every cell in the input table can be linked, unless the knowledge base doesn't contain the corresponding entity, which is shown as grey cell in Figure 2. All the red cells in mention table will be linked to corresponding purple cells in entity table.

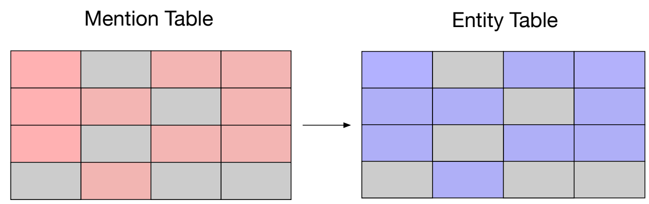

Figure 2: Snapshot of cross-lingual table linking

3 APPROACH

In this section, we describe our neural network based joint model for cross-lingual table linking. We first give an overview of our approach, followed by the method for candidate entity generation. Then we describe translation module in detail. Afterwards, we introduce mention, context and coherence features used in our model respectively. In the last part, we discuss the training process and prediction strategy.

3.1 Model Overview

Figure 3 gives a general view of our neural network based joint model. The reason we call it a "Joint Model" is that the input of neural network is one whole mention table consisting of several cells to be linked, together with a corresponding candidate entity table. All the mentions in the table will be linked simultaneously. The output of our neural network stands for the score of a pair of $\langle T_M, T_E\rangle$, which indicates the probability of choosing that candidate entity table as final result. Specifically, we learn two different features including cell feature and context feature (detailed in Section 3.4) from the pair of mention table and candidate entity table. To make different representations from two language space compatible, we utilize a bilingual translation matrix to transform the non-English vector into English vector space (detailed in Section 3.3). Meanwhile, we learn a third feature called coherence feature (detailed in Section 3.5) only from the candidate entity table. We then combine these three features together to calculate the final score, which is convenient for training and prediction in this linking task (detailed in Section 3.6).

3.2 Candidate Entity Generation

To generate a candidate entity table, we have to generate candidate English entities for each non-English mention at first. Without a reliable non-English knowledge base as the bridge, we use translation tools to produce a set of possible translations for a given non-English mention. Afterwards, we use several heuristic rules to obtain candidate English entities. The set of 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 will be translated into "person of interest" or "suspect tracking", depending on what translation tool to use. The corresponding candidate entity set would contain

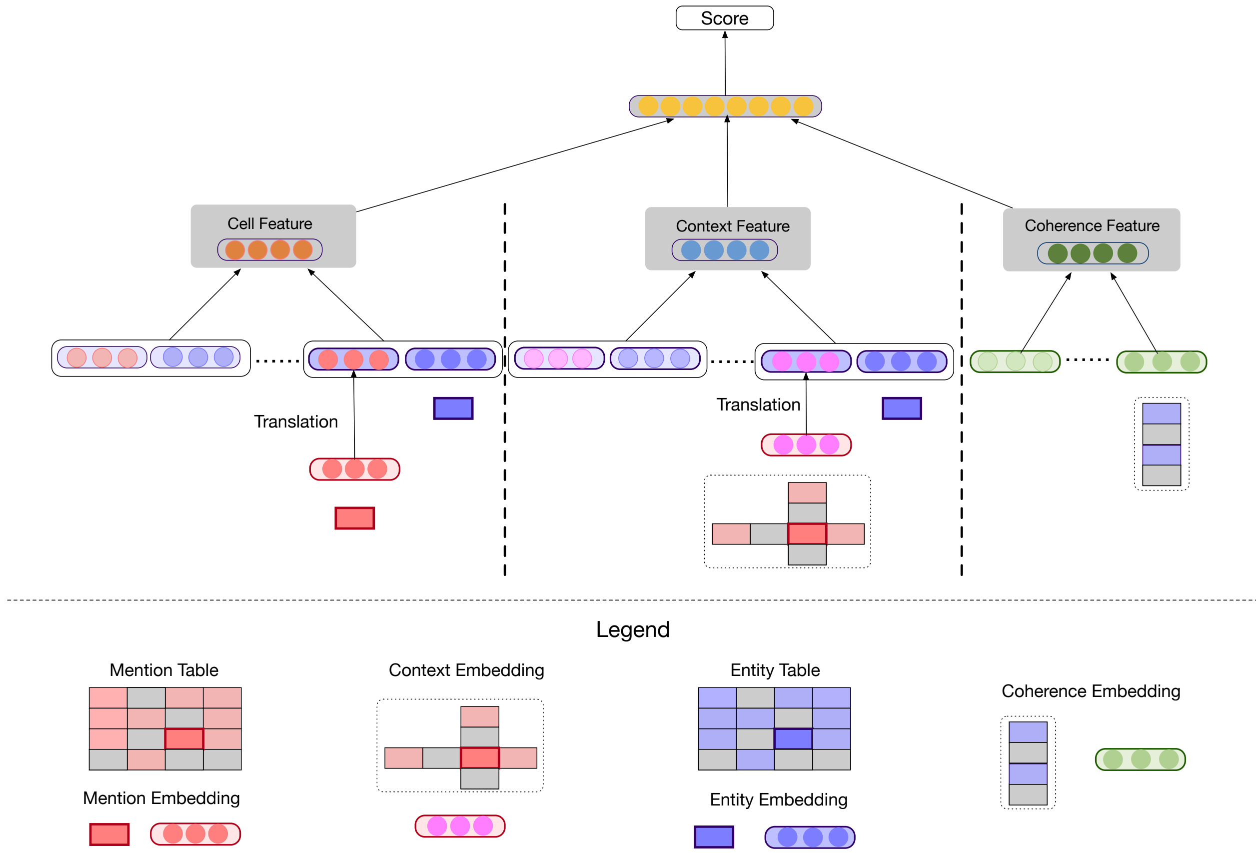

Figure 3: Overview of proposed neural network based joint model.

entities like "person of interest", "person of interest (tv series)" and "suspect (1987 film)".

3.3 Translation Module

As mentioned in model overview, the input of joint model are two tables, mention table and entity table, where each of them is represented as a table of vector embeddings. Since a mention or entity name typically contains up to three words, we simply represent them as the average of embeddings of words they contain. The word embeddings are trained on large scale text corpus. Since mention table and entity table are written in two different languages, we train word embeddings on two corpus of different languages separately. Thus, the embeddings of a mention and its referent entity are naturally incompatible and we can't directly compare or calculate them. To solve this problem, We employ a bilingual translation layer to map embeddings in one language space to another language space. Through this translation layer, a non-English mention embedding v_m can be translated into an English mention embedding $\widetilde{v_m}$ roughly through $\widetilde{v_m} = W_t v_m + b_t$, where W_t is the translation matrix and b_t is the bias. Notice that W_t and b_t are model parameters and will be updated during training so that the translation step will be more and more accurate.

In order to find a good starting point to train the model and jump out of local optima, we train W_t and b_t in advance. We use a small number of bilingual word embedding pairs $\langle v_{wc}, v_{we} \rangle$ to train the parameters. The loss function is as follows.

$$L(W_t, b_t) = \|W_t v_{wc} + b_t - v_{we}\|_2 \tag{1}$$

The list of bilingual word embedding pairs are called translation seeds. We learn a initial translation matrix by minimize the loss and then feed the weights into the model before training.

3.4 Cell and Context Feature

As shown in Figure 3, the output of our model is a score, which represents the linking confidence between a mention table and a candidate entity table. This score comes from two categories. One is to measure how similar or compatible two tables are. We employ two features called cell feature and context feature to capture the compatibility between mention table and candidate entity table.

After translation layer, each mention embedding $v_{\mathbf{m}_{ij}}$ is converted into the same vector space as entity embedding. We concatenate translated mention embedding $\widetilde{v_{\mathbf{m}_{ij}}}$ with entity embedding $v_{\mathbf{e}_{ij}}$ and then go through a fully connected layer to get a hidden feature for a pair of cells $\langle \mathbf{m}_{ij}, \mathbf{e}_{ij} \rangle$. After averaging among all cells which need to be linked, we get the cell feature, which now represents a pair of tables $\langle T_M, T_E \rangle$.

$$f_{cell}(T_M, T_E) = \frac{1}{|P|} \sum_{\langle i,j \rangle \in P} FC([\widetilde{v_{\mathbf{m}_{ij}}}, v_{\mathbf{e}_{ij}}]) \tag{2}$$

Where FC represents fully connected layer, and $[v_1, v_2]$ means vector concatenation.

Context feature follows similar idea. Instead of using mention embedding, we choose context embedding to concatenate with entity embedding. Context embedding for a particular cell $\mathbf{m}_{ij}$ is an average of all mention embeddings of cells which are in the same row or the same column as $\mathbf{m}_{ij}$ (exclude $\mathbf{m}_{ij}$ itself). Detailed equation is as follows.

$$f_{cxt}(T_M, T_E) = \frac{1}{|P|} \sum_{\langle i,j \rangle \in P} FC([\widetilde{vcxt_{\mathbf{m}_{ij}}}, v_{\mathbf{e}_{ij}}]) \tag{3}$$

$$vcxt_{\mathbf{m}_{ij}} = \frac{1}{|\pi_{ij}|}\Big(\sum_{\langle i,k \rangle \in P, k \neq j} v_{\mathbf{m}_{ik}} + \sum_{\langle k,j \rangle \in P, k \neq i} v_{\mathbf{m}_{kj}} \Big) \tag{4}$$

Where π_{ij} is number of cells which are in the same row or column as cell $\mathbf{m}_{ij}$. Regardless table offers rare context information for entity disambiguation, mentions in the same row or column contain strong relatedness and can be regarded as surrounding context.

By learning cell feature and context feature from mention table and candidate entity table, we can capture a general sense of semantic relatedness of all mention-entity pairs from two tables.

3.5 Coherence Feature

The previous two features try to encode the similarity or compatibility of mention table and entity table. Besides, if we look at a correctly linked table, the inner structure of that table is also valuable. The intuition is that entities in the same column (row) tend to own the same type, or we can say that their embeddings are very close. Therefore, we propose a third feature, which captures this correlations among entities of same column.

We calculate the dimension-wise variance for all entity embeddings in the same column to get a coherence embedding for that column. Then we average them among all columns to get the coherence feature for an entity table. This feature represents how closely connected all the entities with the same column, which would help entity disambiguate. For an entity table T_E of size $R \times C$, we get calculate coherence feature as follows.

$$f_{coh}(T_E) = \frac{1}{C} \sum_{j} var(\{v_{\mathbf{e}_{ij}}, i \in [0, R], \langle i,j \rangle \in P\}) \tag{5}$$

Where var is a function to calculate dimension-wise variance for a bunch of vectors.

3.6 Training and Prediction

We apply the learned features for cross-lingual table linking in a ranking framework. Given three feature representations, we calculate a final score after apply a fully connected layer and an output layer.

$$score(T_M, T_E) = W_{out} \cdot FC([f_{cell}, f_{cxt}, f_{coh}]) + b_{out} \tag{6}$$

Where W_{out} is a weight vector and b_{out} is a bias value.

We believe this final score indicates the probability of choosing current candidate entity table as the linking result. Based on that assumption, we design our prediction algorithm as Algorithm 1.

Algorithm 1 Iterative Prediction

Input: Mention table T_M, start entity table T_{Est}, candidate sets $Cand$, learned model f_{model}
Output: Linking result table T_E

procedure PREDICT(T_M, T_{Est}, $Cand$, f_{model})
  $T_E = T_{Est}$
  $score_{max} = f_{model}(T_M, T_{Est})$
  repeat
    for $\mathbf{e}_{ij}$ in T_E **do** ▷ random order
      $T_{tmp} = T_E$
      for ent in $Cand_{ij}$ **do**
        $replace(T_{tmp}, \langle i, j \rangle, ent)$
        $score = f_{model}(T_M, T_{tmp})$
        if $score > score_{max}$ **then**
          $replace(T_E, \langle i, j \rangle, ent)$
  until score converges
  return T_E

T_{Est} is a candidate entity table where each cell is filled with most possible candidate entity from the step of entity candidate generation. $Cand$ is a collection of candidates for mention table T_M. $Cand_{ij}$ represents a list of candidate entities for cell $\mathbf{m}_{ij}$. $replace(T, \langle i,j \rangle, e)$ is a procedure which replace the entity of table T at position $\langle i,j \rangle$ with a new entity e. Basically, this algorithm iteratively replaces each cell with candidate entities, then chooses the best entity as cell linking result, until no replacement will bring up output score.

This iterative prediction algorithm will be used in linking a new mention table after the model is trained. The algorithm works only if the model can produce a higher score when we replace a wrongly-linked entity with the correct entity in candidate entity table. For effectively training the model, we use the idea of learning to rank [3] and devise a pairwise ranking loss function. The basic idea is that the score of a candidate entity table should be larger (ranks higher) than any other candidate entity table with less correctly linked cells.

Given a mention table T_M and a positive entity table T_P, we generate a list of negative entity tables T_Ns by randomly corrupt a random number of cells of T_P. Let T_E equals to T_P and all T_Ns. We define a label list $y_i = r(T_{Ei})$, where $r(T_{Ei})$ represents the correct cell ratio of each entity table T_{Ei}. $s_i = f_{model}(T_M, T_{Ei})$ is the score list for each . Then the likelihood and cost function can be written as:

$$Likelihood = \prod_{i,j} U_{ij}^{\widetilde{U_{ij}}} \cdot (1 - U_{ij})^{(1 - \widetilde{U_{ij}})} \tag{7}$$

$$J = -\sum_{i,j} (\widetilde{U_{ij}} \log U_{ij} + (1 - \widetilde{U_{ij}}) \log(1 - U_{ij})) \tag{8}$$

Where

$$\widetilde{U_{ij}} = \begin{cases} 1 & y_i < y_j \\ 0.5 & y_i = y_j \\ 0 & y_i > y_j \end{cases} \tag{9}$$

$$U_{ij} = sigmoid(s_i - s_j) \quad (10)$$

4 IMPLEMENTATION DETAILS

pre-train **Translation Model Pre-Train**

In order to pre-train of the translation model, we collect a bilingual lexicon of common words using Bing Translate API [1], containing 91,346 translation pairs at word level. Each pair has a confidence score ranging from 0 to 1. We further select those pairs in which both the Chinese and English word perfectly match the name of an article in Wikipedia. Out of 23,863 pairs after filtering, we finally pick all 3655 translation pairs as our pre-train dataset, with score no smaller than 0.5 for each pair.

Rank Net

Parameter Tuning

In our joint model, we tune the following parameters:

- The size of candidates per mention (denoted by N_{cand}) in the range of {1, 3, 5, 10, 20, 30, 40, 50},
- The number of negative entity tables (denoted by N_{tab}) in {9, 19, 49, 99},
- The dimension of cell, context and overall feature (d_{cell}, d_{cont} and d_{out}) in {20, 50, 100, 200}.
- The learning rate η in {0.0002, 0.0005, 0.001},
- The L1- and L2- regularization l_1, l_2 in {0.0001, 0.0002, 0.0005, 0.001}.

All the parameters are tuned on the validation set, the detail evaluation metric is discussed in Section 5.4.

5 EXPERIMENTS

In this section, we evaluate our table linking system and compare with other previous works on the cross-lingual task. We first introduce the datasets and state-of-the-art systems used 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, and perform ablation experiments to investigate the importance of different components used in the whole task. Finally, we discuss and analyze the errors in our system.

5.1 Experimental Setup

Wikipedia and Word Embeddings

We use the Feb. 2017 dump of English Wikipedia [2] and Chinese Wikipedia [3] 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 embedding, all the entities occured 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]]". The advantage is that, by learning embeddings of both common and special words in a uniform vector space, each entity is represented by the embedding of its identical word, which is more precise than the aggregation of word embeddings in the entity's name. Besides, in order to enlarge the number of anchor texts in the corpora, e automatically add more anchor texts to both Chinese and English Wikipedia: for each article page, we simply find all the surface form of phrases exactly matching the article name, and then transform these phrases into an anchor text, linking to the current article. Next we adopt Word2Vec [20] to train word embeddings from both corpus respectively, the embedding dimennsion sets to 100.

Table Linking Dataset

The cross-lingual table linking dataset consists of 116 web tables with Chinese mentions and linked English Wiki articles. The original Chinese tables are created by Wu et al. 2016 [34], which contains 123 tables extracted from Chinese Wikipedia, and each mention is labeled by its corresponding Chinese Wiki article. We tranfrom all the Chinese entities into English via inter-language links of Wikipedia, producing the labeled English entities for 78% of the entire mentions. In addition, we discard long-tail tables, if the shape is smaller than 5*3, or if the number of labeled English entities is smaller than its columns. In total, We collected 3056 mentions from 116 tables, with 2253 mentions been linked to English entities (19.42 mentions per table). We randomly split the dataset into training / validation / testing sets (80 : 12 : 24 tables). We will publish the dataset later.

Translating Tools. [4] [5] [6]

5.2 State-of-the-art Comparisons

Since there are no previous works 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 [1] and Wu et al. 2016 [34]. We call their systems $TabEL_B$ and $TabEL_W$ in short. In order to make a fair comparison in our bilingual scenario, we bridge the language gap as follows. For both systems, we apply the translation procedure in Section 3.2, converting each cell mention into the most likely English surface form, then run the mono-lingual model on these translated English tables and produce the final linking results.

The second branch is cross-lingual text linking, we compare with Zhang et al. 2013 [36], and call it $TextEL$. This work aims at entity linking from foreign languages to English on unstructured texts, and proposed the BLDA model, which models each foreign text and English Wiki article as the probabilistic distribution of latent bilingual topics in a uniform space. In this case, we traverse each mention in row order and flatten the whole table into a word sequence, then mark the word intervals for mentions to be linked. By turning the table into unstructured texts, the previous work is able to learn more flexible context information, but it may hard to capture the correlation of entities in the same column.

5.3 Evaluation of Candidate Generation

In this part, we investigate the translated English mentions from Chinese table inputs. As described in Section 4, English mentions is derived from multiple resources. Compare with different combination of resources, we evaluate the quality by measuring the

[1] http://www.bing.com/translator
[2] http://
[3] http://

[4] http://fanyi.qq.com
[5] http://fanyi.baidu.com
[6] http://translate.google.cn

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. Besides, the English mentions generated by Pinyin is a complementary to those generated from pure translation methods.

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**	0.705	0.723
Pinyin	0.046	0.052	0.053
Trans. + Pinyin	**0.558**	**0.708**	**0.726**

5.4 End-to-End Results

Now we perform the cross-lingual table linking experiment and compare with previous table linking and text linking systems. Recap that not all mentions in our dataset have a gold English entity. we only attempt to link those mentions with English labels (see Definition 2.1). To be consistent with state-of-the-art systems, we report Micro Accuracy and Macro Accuracy as the evaluation metrics. Micro Accuracy, used in $TabEL_B$ [1] and $TabEL_W$ [34], is the fraction of cells where the predicted entity exactly matches the labeled English concept. We also adopt Macro Accuracy, defined as the fraction of correctly linked cells averaged over all tables, this metric avoids the results biased the table with more cells.

In order to keep a fair comparison, due to both $TabEL_B$ and $TabEL_W$ taking only one English mention per cell as the input, we select Baidu translation as the best one (see Section 5.3) and apply this translation setting to all approaches. In addition, we evaluate our approach using the full translating strategy. For the two variations of our approach, we fix $N_{cand} = 30$, $N_{tab} = 49$, $d_{cell} = d_{cont} = 100$, $d_{out} = 200$, $\eta = 0.0002$ and $l_1 = l_2 = 0.0005$, as reaching the highest Micro Accuracy in the validation set. Whereas we use different N_{cand} for the other approaches, tuning separately.

We report the experimental results in Table 2. Our model outperforms the other baseline models, improving the result by up to 13.7%. Our full model even improves the Micro Accuracy by an absolute gain of 0.048, showing the importance of combining multiple translating tools. Both $TabEL_B$ and $TabEL_W$ suffers from the problem of error propagation in the translating step, as the monolingual approaches take translated mentions as direct input, a poor translating quality harms the final result. Whereas our approach takes Chinese mention as the input, such end-to-end approach alleviates the error brought by translation.

We further analyze how the candidate size of a mention effects the table linking result. The intuition is that, when N_{cand} goes larger, the upperbound of the final result increases, however, with more negative candidates been introduced, it's more difficult for a linking system to reach the upperbound. Therefore, we perform the experiment under different candidate size and investigate how does each model handle this tradeoff.

Table 2: Experimental results on Chinese table linking task. All baseline approaches take Baidu as the only translating tool.

Approach	Micro Acc.	Macro Acc.
$TabEL_B$	0.513	0.511
$TabEL_W$	0.517	0.521
$TextEL$	0.476	0.465
Ours (Baidu Only)	0.588	0.583
Ours (Full Trans)	**0.636**	**0.630**

Figure 4 shows the trend of each approach, using Baidu translation only. We also display upperbound (Hits@n) in the figure. From the results, we observe that the Micro Accuracy increases when N_{cand} is small, and then decreases when N_{cand} goes larger. Our approach and $TabEL_B$ keeps a stable performance with a subtle performance decreasing, while $TextEL$ drops dramatically, even if the candidate size is smaller than 10. Two main reasons are: 1) the BLDA model is unsupervised, though equivalent articles between Chinese and English Wikipedia pair together as the input documents, the model doesn't observe any explicit (mention, entity) pair for learning, 2) the similarity between the mention and the entity is only determined by their topic distributions, without features derived from entity names, or from coherence information. These reasons make $TextEL$ much more sensitive to noisy candidates. Compared with baseline systems, our approach is more adaptive to different size of candidates, with less error propagation during the translation step, and can produce promising end-to-end results.

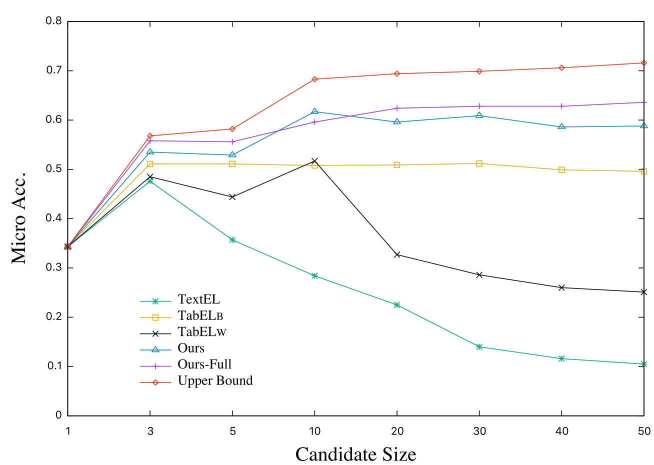

Figure 4: Experimental results of Micro Accuracy varied from different size of candidates, using Baidu translation only.

5.5 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. Table 3 shows the results of feature ablation studies. All features in our model make a positive effect on the final accuracy. Both the cell feature and the context feature directly represents the latent semantic association between the embedding of an individual mention (either from itself or from its neighborhoods) and the candidate entity. Among them, the cell feature is more important in the model 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 is incurred, largely due to the lack of dominant and direct semantic features. Nevertheless, the coherence feature is complementary to the other features, as it aims to discover 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.

As the example shown in Figure 5, the mention "钢铁侠" could be 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 cell + context features only, after applying the coherence feature, the strong correlation with entities in the same column makes the model bias the correct film entity.

Table 3: Micro Accuracies and performance decreasing percentages on the validation set using different feature combinations.

Feature Combination	Micro Acc.	Decrease (%)
Cell Only	0.662	4.85
Context Only	0.654	6.06
Cell + Context	0.679	2.44
Coherence Only	0.228	67.28
Full	0.696	0.00

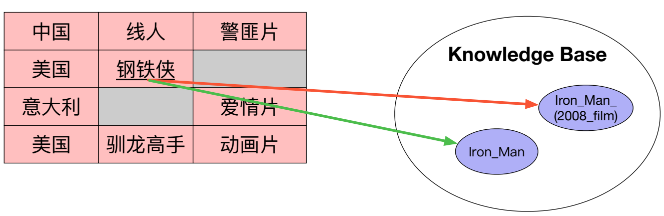

Figure 5: A real example of table linking. The green arrow points to the entity predicted without using coherence feature, and the red arrow points to the entity predicted by using all features.

Joint Model Versus Non-Joint Model

We investigate the effect of joint model without introducing the coherence feature. As discussed before, the cell and context features focus on relationships of individual mentions, then we simply transform the table linking problem into a traditional entity linking problem: given a mention's cell and context embedding (rather than a table of multiple mentions), link the mention to an entity in the knowledge base. Also, the gold linking result of a training table becomes a list of $(mention, entity)$ pairs.

We construct a non-joint model as the baseline: during training step, we adopt hinge loss as the optimizing function, in order to maximize the margin λ between the gold entity e^+ and other entities e^- of a mention m, defined as follows:

$$l_m = \max_{e^- \in Cand(m)} \max\{0, \lambda + score(m, e^-) - score(m, e^+)\}. \quad (11)$$

In this model, $score(\cdot, \cdot)$ is similar with Eq. (6), but remove the coherence feature, and don't need to average cell and context features over mentions. The parameter λ is tuned in {1.0, 2.0, 3.0, 4.0}. We also perform another ablation test, where we apply the joint model, but the pairwise rank loss is replaced by the hinge loss.

Table 4 shows the comparison of Micro Accuracy on the testing set. We find out that when both using hinge loss, the non-joint model outperforms the joint model. We believe that hinge loss is not suitable for use in our joint model, because: 1) all the negative candidates of each mention are used in the non-joint model, however, in the joint model, negative entity tables are generated by random corruption (see Section 3.6), some negative candidates are not sampled and hence cannot be observed by the model; 2) hinge loss focus on the margin between the positive entity table and the nearest negative entity table (only a few corruption), thus many negative tables with many corruptions become unimportant, which shrinks the size of training data to some extent. Compared with the hinge loss, the pairwise ranking model imposes the intuition that the more corruption a table has, the lower score it holds. Once negative entity tables are comparable of each other, the training model could make full use of negative entity tables.

Table 4: Experimental results on the testing set under different model specification. Only mention features and context features are used in each model.

Specification	Micro Acc.	Decrease (%)
Non-Joint, Hinge Loss	0.586	2.01
Joint, Hinge Loss	0.574	4.01
Joint, RankNet	0.598	0.00

6 RELATED WORK

Our task could be viewed as a special case of entity linking, with the following restrictions: 1) the queries are all Web tables; 2) the queries and KB entities are in different languages. In Section 6.1 we discuss about works on standard entity linking tasks; in section Section 6.2 we discuss works about Web tables; and in Section 6.3 we discuss works on cross-lingual entity linking systems.

6.1 Entity Linking

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 [2], YAGO [27] and Wikipedia, where each Wikipedia article is considered as an entity. Most works focus on linking to Wikipedia and thus the task is also named as Wikification. The typical procedure of entity linking contains two stages: candidate generation, where the surface forms (mention) in the query text which could be linked to certain entities in the KB are identified, and a set of candidate entities are proposed for each entity mention; candidate ranking, where the candidates for each mention are ranked (usually based on context) and the best one is returned as linking result. 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 [6, 7, 14].

Wikipedia was first explored by Bunescu and Pasca [24], where an SVM kernel was used to compare the lexical context of an entity mention to each candidate's Wikipedia page. Since each entity mention needed to train its own SVM model, the experiment was limited. Later, Mihalcea and Csomai [19] proposed a system called Wikify! for the Wikification task. They applied word sense disambiguation to this task, and experimented with two methods to link detected candidates to a Wikipedia page: 1) comparing the mention's lexical context to content of disambiguation page; 2) training a Naive Bayes classifier for each ambiguous mention.

Later approaches made use of the observation that entity disambiguation in the same document should be related. Cucerzan [8] maximized the agreement between the context data stored for each candidate entity and the contextual information in the document, and also the agreement among the category tags of the candidate entities. Milne and Witten [21] took a similar approach but relied on unambiguous terms in the context. Han and Zhao [10] constructed a large-scale semantic network from Wikipedia, then computed similarity between query and candidate entity based on the semantic network. Ratinov et al. [25] formalized this task into a bipartite graph matching problem and proposed a score function which considered both local similarity and global coherence. Zhang et al. [37] employed a Wikipedia-LDA model and the contexts were modeled as a probability distribution of Wikipedia categories. The similarity score between candidates and entities were computed based on the category distribution. Cai et al. [5] proposed to first enrich the sparsely-linked articles by adding more links iteratively and then use the resulting link co-occurrence matrix to disambiguate the mentions in an input document. Yang and Chang [35] proposed a tree-based structured learning framework, S-MART, which is particular suitable for short texts such as tweets.

Similar to our work, Sun et al. [28] 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 outputted as the score of a <mention, context, candidate> triplet and trained with hinge-loss. On the other hand, our model jointly assign the mentions simultaneously.

6.2 Web Table and Table Entity Linking

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. [4]. They found that there were about 154 million tables on the Web that could be used as a source of high-quality relational data, and implemented several applications such as schema auto-complete and attribute synonym finding. Muñoz et al. [22] proposed methods to mine rdf triplets from Wikipedia tables, and Sekhavat et al. [26] 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.

Syed et al. [29] proposed approaches to infer a partial semantic model of Web tables automatically with Wikitology [30]. Limaye et al. [17] proposed a framework to annotate table cells with entity, type and relation information simultaneously using a graphical model. Ibrahim et al. [12] presented a probabilistic graphical model to capture coherence between cells in tables and candidate entities, concepts or quantities. They devised a system to map table entries and headers to concepts, classes, entities and uniquely represented quantities in the form of <measure, value, unit> triple.

Other works focused solely on table entity linking. Bhagavatula et al. [1] argued that models which jointly address entity linking, column type identification and relation extraction rely on the correctness and completeness of KB, thus may be adverse for the performance of entity linking. They also exploited a graphical model, where cells in the same row or column are connected. They trained a model to rank the candidates of a given cell by its context, i.e. cells connected to the given cell, and the predicted entity for each cell are iteratively updated until convergence. Wu et al. [34] constructed a graph of mentions and candidate entities for each query table, then use page rank [23] to determine the similarity score between mentions and candidates. Besides, they combined multiple knowledge bases in Chinese to enhance the system.

6.3 Cross-Lingual Entity Linking

Starting from 2011 the annual TAC KBP Entity Linking Track has been using the multi-language setting [14–16], where the languages involved are English, Chinese and Spanish. Most systems for this task were adaptations of mono-lingual entity linking systems: either first do entity linking on foreign languages first then translate the results to English via language links, which requires a comprehensive knowledge base in the foreign languages; or first translate the query into English by some machine translation tool then apply English entity linking algorithms, whose performance greatly relies on the machine translator.

Some other systems tried to avoid the usage of such assumptions. McNamee et al. [18] first experimented with cross-lingual entity linking on documents. They first used a machine translation tool developed by Irvine et al. [13] to transliterate the detected query mentions into English and transform the task into a mono-lingual one. Then they extracted some features and ranked the candidates with SVM-rank. However, to train this model, parallel corporas which are well aligned at sentence level were required.

Most methods managed to bridge the language gap through language-independent spaces. Fahrni et al. [9] 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. [37] leveraged a modified version of Latent Dirichlet Allocation, which they call BLDA (Bilingual LDA) and bridged the gap between languages via topic space. They trained the topic model on English-Chinese Wikipedia page pairs (indicated by inter-language links) and disambiguated candidate entities by computing the inner product of the topic distributions of the query text and the entity Wikipedia page. Their approach does not require supervised learning and performs well with a conservative candidate generation stage. Wang et al. [32] 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. [31] trained a multilingual word and title embeddings and ranked entity candidates using features based on these multilingual embeddings. They used canonical correlation analysis [11] to project the embeddings of two languages into the same space, whose goal is the same as the translation layer in our model.

7 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

[1] Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2015. TabEL: entity linking in web tables. In *International Semantic Web Conference*. Springer, 425–441.

[2] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. AcM, 1247–1250.

[3] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*. ACM, 89–96.

[4] Michael J Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment* 1, 1 (2008), 538–549.

[5] Zhiyuan Cai, Kaiqi Zhao, Kenny Q Zhu, and Haixun Wang. 2013. Wikification via link co-occurrence. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 1087–1096.

[6] Amparo Elizabeth Cano, Giuseppe Rizzo, Andrea Varga, Matthew Rowe, Milan Stankovic, and Aba-Sah Dadzie. 2014. microposts2014 neel challenge: Measuring the performance of entity linking systems in social streams. *Proc. of the Microposts2014 NEEL Challenge* (2014).

[7] David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June Paul Hsu, and Kuansan Wang. 2014. ERD'14: entity recognition and disambiguation challenge. In *ACM SIGIR Forum*, Vol. 48. ACM, 63–77.

[8] Silviu Cucerzan. 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. (2007).

[9] Angela Fahrni and Michael Strube. 2011. HITS'Cross-lingual Entity Linking System at TAC 2011: One Model for All Languages.. In *TAC*.

[10] Xianpei Han and Jun Zhao. 2009. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 215–224.

[11] Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika* 28, 3/4 (1936), 321–377.

[12] Yusra Ibrahim, Mirek Riedewald, and Gerhard Weikum. 2016. Making Sense of Entities and Quantities in Web Tables. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 1703–1712.

[13] Ann Irvine, Chris Callison-Burch, and Alexandre Klementiev. 2010. Transliterating from all languages. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*.

[14] Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the TAC 2010 knowledge base population track. In *Third Text Analysis Conference (TAC 2010)*, Vol. 3. 3–3.

[15] Heng Ji, Joel Nothman, and Ben Hachey. 2014. Overview of tac-kbp2014 entity discovery and linking tasks. In *Proc. Text Analysis Conference (TAC2014)*. 1333–1339.

[16] Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian. 2015. Overview of tac-kbp2015 tri-lingual entity discovery and linking. In *Proceedings of the Eighth Text Analysis Conference (TAC2015)*.

[17] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. 2010. Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB Endowment* 3, 1-2 (2010), 1338–1347.

[18] Paul McNamee, James Mayfield, Dawn Lawrie, Douglas W Oard, and David S Doermann. 2011. Cross-Language Entity Linking.. In *IJCNLP*. 255–263.

[19] Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. (2007), 233–242.

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[21] David Milne and Ian H Witten. 2008. Learning to link with wikipedia. (2008), 509–518.

[22] Emir Muñoz, Aidan Hogan, and Alessandra Mileo. 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*. ACM, 533–542.

[23] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[24] Razvan Bunescu Pasca and Marius. 2006. Using Encyclopedic Knowledge for Named Entity Disambiguation. (2006).

[25] Lev Ratinov, Dan Roth, Doug Downey, and Michael R Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. (2011), 1375–1384.

[26] Yoones A Sekhavat, Francesco Di Paolo, Denilson Barbosa, and Paolo Merialdo. 2014. Knowledge Base Augmentation using Tabular Data.. In *LDOW*.

[27] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 697–706.

[28] Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. Modeling Mention, Context and Entity with Neural Networks for Entity Disambiguation.

[29] Zareen Syed, Tim Finin, Varish Mulwad, and Anupam Joshi. 2010. Exploiting a web of semantic data for interpreting tables. In *Proceedings of the Second Web Science Conference*, Vol. 5.

[30] Zareen Saba Syed, Tim Finin, and Anupam Joshi. 2008. Wikitology: Using wikipedia as an ontology. In *proceeding of the second international conference on Weblogs and Social Media*.

[31] Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual wikification using multilingual embeddings. In *Proceedings of NAACL-HLT*. 589–598.

[32] Han Wang, Jinguang Zheng, Xiaogang Ma, Peter Fox, and Heng Ji. 2015. Language and Domain Independent Entity Linking with Quantified Collective Validation.. In *EMNLP*. 695–704.

[33] Jingjing Wang, Haixun Wang, Zhongyuan Wang, and Kenny Q Zhu. 2012. Understanding tables on the web. In *International Conference on Conceptual Modeling*. Springer, 141–155.

[34] Tianxing Wu, Shengjia Yan, Zhixin Piao, Liang Xu, Ruiming Wang, and Guilin Qi. 2016. Entity Linking in Web Tables with Multiple Linked Knowledge Bases. In *Joint International Semantic Technology Conference*. Springer, 239–253.

[35] Yi Yang and Mingwei Chang. 2016. S-MART: Novel Tree-based Structured Learning Algorithms Applied to Tweet Entity Linking. (2016), 504–513.

[36] Tao Zhang, Kang Liu, Jun Zhao, and others. 2013. Cross Lingual Entity Linking with Bilingual Topic Model.. In *IJCAI*.

[37] Wei Zhang, Jian Su, and Chew Lim Tan. 2011. A Wikipedia-LDA Model for Entity Linking with Batch Size Changing Instance Selection.. In *IJCNLP*. 562–570.