# XLink: An Unsupervised Bilingual Entity Linking System

Jing Zhang, Yixin Cao, Lei Hou, Juanzi Li, and Hai-Tao Zheng

Department of Computer Science and Technology,
Tsinghua University, Beijing 100084, P.R. China
{zhangjinglavener,caoyixin2011,greener}@gmail.com,lijuanzi@tsinghua.edu.cn,zheng.haitao@sz.tsinghua.edu.cn

**Abstract.** Entity linking is a task of linking mentions in text to the corresponding entities in a knowledge base. Recently, entity linking has received considerable attention and several online entity linking systems have been published. In this paper, we build an online bilingual entity linking system XLink, which is based on *Wikipeida* and *Baidu Baike*. XLink conducts two steps to link the mentions in the input document to entities in knowledge base, namely mention parsing and entity disambiguation. To eliminate dependency of language, we conduct mention parsing without any named entity recognition tools. To ensure the correctness of linking results, we propose an unsupervised generative probabilistic method and utilize text and knowledge joint representations to perform entity disambiguation. Experiments show that our system gets a state-of-the-art performance and a high time efficiency.

**Keywords:** Entity Linking System, Entity Disambiguation, Mention Detection

## 1 Introduction

Entities, which describe specific concepts or represents real-world objects, play an important role in Web document, e.g., *persons*, *locations* and *organizations* record key elements of news articles. Unfamiliar entities may affect text understanding. Fortunately, large knowledge bases contain rich information about world's entities, including their semantic classes and mutual relationships. To make Web text more understandable, a critical step is to linking the entity mentions with their corresponding entities in a knowledge base, which is entity linking. Besides text understanding, it can facilitate many different tasks such as question answering [21, 24], knowledge base population [18].

In recent years, various entity linking systems have been published, such as Wikify! [14], AIDA [12], DBpedia Spotlight [13], TagMe [9], Linkify [22]. These systems commonly have two components: mention detection and entity linking. For mention detection, AIDA [12] and Linkify [22] depend on Names Entity Recognition (NER) tools. However, NER tools depend on language heavily [10] and only recognize three types of named entities: *persons*, *locations* and *organizations*, far from covering the types of entities in knowledge base. To address the problem of ambiguity and variation in entity linking, the simplest way is to choose the most prominent entity (i.e., the candidate with the largest number of incoming or outcoming links in Wikipedia) for the given mention. However, different context of mentions leads to different linking results, which is too complex to be solved through entity priority. An alternative idea calculates the contextual similarity for single mention linking, and further employs the topical coherence to collectively link all mentions within a document. But few of these systems considers the features in a unified and effective way. Moreover, these systems mainly use Wikipedia as the knowledge base and rarely handle Chinese documents. Additionally, there emerge many large-scale Chinese encyclopedias, e.g., Baidu Baike, and it's time to conduct entity linking in both Chinese and English.

To address the above issues, we develop a bilingual online entity linking system named XLink. It conducts a language independent process for both Chinese and English documents on-the-fly via two phases: mention parsing and entity disambiguation. Mention parsing detects mentions in input documents and generates candidate entities for each mention. Entity disambiguation phase chooses the correct entity in the candidate set. XLink aims to provide users an online service of linking all important mentions in text to entities in knowledge base correctly and efficiently. In particular, we first use a parsing algorithm to search a pre-built dictionary to detect mentions instead of using NER tagger. Secondly, we design a generative probabilistic entity disambiguation method which models contextual feature, coherence feature and prior feature jointly to guarantee the accuracy of disambiguation. For the system efficiency, we use Aho-Corasick algorithm to parse mentions and introduce word and entity embeddings to ensure the time efficiency of disambiguation phase. In addition, the disambiguation method is unsupervised so that it is easy to deploy the system online.

In summary, the main contributions of this paper can be described as follow:

1. We utilize a pre-built dictionary rather than a NER tagger to detect mentions to avoid language dependency and recognize entities of more types.
2. We propose an unsupervised generative probabilistic model to disambiguate entities for the detected mentions. *Context* of mentions and *entity coherence* as well as *priority* are employed to promote the performance of disambiguation. Experiment shows that the disambiguation algorithm significantly outperforms the state-of-the-art unsupervised approaches.
3. We construct a web service of XLink. Users can enter a text fragment of any types (e.g., news, tweets) in Chinese or English, and XLink adds URLs to the mention labels for visualization on the web page and ranks the results according to disambiguation confidence score.

The rest of this paper is organized as follow. In section 2 we present the definitions. In section 3, we describe the framework of XLink and details of the methods. In section 4, we present evaluations of our system. In section 5 we discuss the related work and finally in section 6 we present our conclusions and future work.

## 2 Problem Definition

We introduce some concept definitions and problem formulation in this section.

**Definition 1.** *A* ***knowledge base*** $\mathcal{KB}$ *contains a set of entities* $\mathcal{E} = \{e_j\}$*. Each entity corresponds a page containing title, textual description, hyperlinks pointing to other entities, infobox, etc.*

**Definition 2.** *A* ***text corpus*** $\mathcal{D}$ *contains a set of words* $\mathcal{D} = \{w_1, w_2, ..., w_{|\mathcal{D}|}\}$*. A mention* $m$ *is a word or phrase in* $\mathcal{D}$ *which may refer to an entity* $e$ *in* $\mathcal{KB}$*. In this paper, we pre-train word and entity representations, and use low-dimensional vectors* $v_w$ *and* $v_e$ *to denote the embedding of word* $w$ *and entity* $e$ *in* $\mathcal{KB}$*.*

**Definition 3.** *An* ***anchor*** $a \in \mathcal{A}$ *is a hyperlink in* $\mathcal{KB}$ *articles, which links its surface text mention* $m$ *to an entity* $e$*.* $\mathcal{A}_{e,m}$ *denotes the set of anchors of mention* $m$ *pointing to entity* $e$*. Anchor Dictionary is the dictionary that we build through utilizing all the anchors in* $\mathcal{KB}$*. Each* $a$ *in the anchor dictionary may refer to a set of entities* $\mathcal{E}_m = \{e_j\}$*.*

**Definition 4.** ***Problem Definition.*** *Given a document* $\mathcal{D} = \{w_1, w_2, ..., w_{|\mathcal{D}|}\}$ *and* $\mathcal{KB}$*, the task is to find out the mentions in* $\mathcal{D}$ *and link them to their referent entities. We resolve the problem into two phases. In Mention Parsing, we detect mentions* $\mathcal{M} = \{m_1, m_2, ..., m_{|\mathcal{M}|}\}$ *and generate a candidate entity set* $\mathcal{C} = \{e_1, e_2, ..., e_{|\mathcal{C}|}\}$ *for each mention* $m_j$*. In Entity Disambiguation, we select the most probable entity* $e_j^*$ *in the candidate set* $\mathcal{C}$ *for each mention* $m_j$*.*

## 3 The Anatomy of XLink

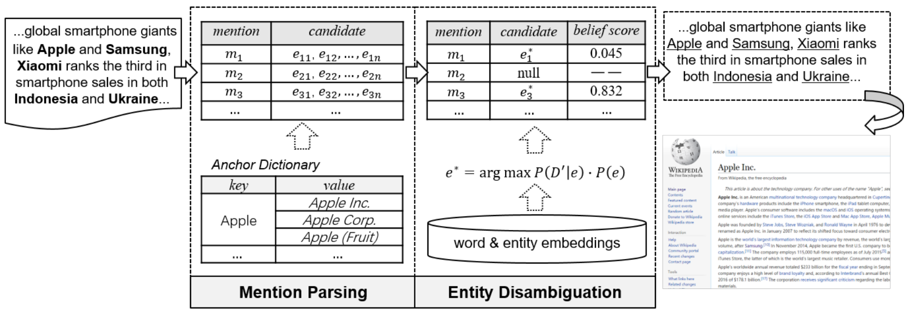


**Fig. 1.** Illustration of XLink

Fig. 1 shows the framework of our proposed XLink system. It receives a textual document $\mathcal{D}$, and tries to link the mentions in $\mathcal{D}$ to entities in $\mathcal{KB}$ in two steps, mention parsing and entity disambiguation. *Mention Parsing* detects mentions by searching the pre-built *Anchor Dictionary*, and generates candidate entities to be disambiguated. *Entity Disambiguation* utilizes a generative probabilistic method to determine which candidate entity $e$ should be linked to the corresponding mention $m$, and pre-trained representations are employed to calculate probabilities.

### 3.1 Mention Parsing

Mention parsing detects the possible entity mentions in the input document $\mathcal{D}$ by searching a pre-built anchor dictionary. Therefore, dictionary building and parsing algorithm are the major problems.

**Dictionary Building.** In Wikipedia and Baidu Baike, the anchor text of a hyperlink pointing to an entity page provides useful name variations of the pointed entity. An anchor can be a synonym, abbreviation or title of an entity. For example, anchor text "Apple" may point to the page of *Apple Inc.*, *Apple Corps.* or *Apple (fruit)* in different documents. Inversely, the entity *Apple Inc.* also has other anchor texts, such as "Apple Computer Inc.". By extracting anchor texts and their corresponding hyperlinks from $\mathcal{KB}$ articles, we can construct an anchor dictionary, where the keys are mentions and values are candidate entities, as shown in Table 1. A mention may refer to several entities and an entity may have several mentions. Thus, we can generate the candidate entities referred by a mention easily by querying the dictionary.

**Table 1.** Examples of Anchor Dictionary

| **Wikipedia** | | **Baidu Baike** | |
|---|---|---|---|
| *key* | *value* | *key* | *value* |
| Microsoft | *Microsoft* | 微软 | 微软 |
| Microsoft Corporation | *Microsoft* | 微软公司 | 微软 |
| Apple | *Apple Inc.*<br>*Apple* | 苹果 | 苹果公司<br>苹果 |
| Apple Computer Inc. | *Apple Inc.* | 苹果电脑公司 | 苹果公司 |

We define $link(a)$ as the times anchor $a$ occurs as a hyperlink in the $\mathcal{KB}$ articles, and $freq(a)$ as the times that $a$ occurs as either a hyperlink or a plain text. $link\_prob(a) = link(a)/freq(a)$ denotes the probability that an anchor $a$ occurs as a hyperlink and points to some entities in $\mathcal{KB}$. To make the dictionary clean, we filter the useless anchors according to the following rules: 1) the anchors with only one character since they usually convey little information; 2) the anchors with low absolute linked frequency ($link(a) < 2$) since it indicates the entity the anchors point to are not popular enough; 3) the anchors with low relative linked frequency ($link\_prob(a) < 0.01\%$) since it implies that the anchors are usually used as general terms. Finally, we also merge the *redirect* information in the anchor dictionary. *Redirect* maps from misspelling, abbreviation or other frequent forms of entity names to the exact entities.

**Parsing Algorithm.** To accelerate parsing process, we use a fast string searching algorithm to parse mentions in anchor dictionaries, $Aho-Corasick\ Algorith$-

$m$[1]. It is a kind of dictionary-matching algorithm that locates elements of a finite set of strings (the "dictionary") within an input text and it matches all strings simultaneously. Informally, the algorithm constructs a finite state machine that resembles a trie with additional links between the various internal nodes. With the pre-built anchor dictionary, we could construct the automaton off-line. In particular, the complexity of the algorithm is linear in the length of the strings plus the length of the searched text plus the number of output matches, which is efficient for online process.

However, because the automaton find all matches simultaneously, there could be a quadratic number of conflicts (substrings and overlaps). To solve the problem, we design an algorithm to choose a match which could be most probable to be an entity mention. For two conflicting mentions $m_1$ and $m_2$, if $m_1$ is much longer, we regard $m_1$ to be more specific than $m_2$. For example, the mention "香港特别行政区" is more specific than "香港". Besides, if $m_1$ has the same length with $m_2$, we choose the one with greater link probability. We assume a mention name with greater link probability is more likely to be linked in text and has less ambiguity intuitively. Furthermore, the parsing algorithm detects mentions iteratively until there is no conflicting mentions in the text, and experiment shows the algorithm can be terminated after a smaller number of iterations. Finally, we could generate candidate entity set $\mathcal{C} = \{e_1, e_2, ..., e_{|\mathcal{C}|}\}$ for mention $m_j$.

### 3.2 Entity Disambiguation

With the candidate set $\{e_1, e_2, ..., e_{|\mathcal{C}|}\}$ for each mention $m_j$, *Entity Disambiguation* selects the most suitable entity $e_j^*$. Inspired by the works of Han et.al [11], we propose a generative probabilistic method to model the context, coherence and priorities jointly. In the following, we present the model details and representations of words and entities.

**Joint Embedding of Words and Entities.** We utilize the models proposed by [23], jointly learning the embeddings of words and entities and mapping words and entities into the same continuous vector space. It consists of three models based on skip-gram model: 1) the conventional skip-gram model that learns to predict neighboring words given the target word in text corpora; 2) the knowledge base graph model that learns to estimate neighboring entities given the target entity in the link graph of the knowledge base. 3) the anchor context model that learns to predict neighboring words given the target entity using anchors and their context words in the knowledge base. Essentially, the knowledge graph model learns the relatedness of entities and the anchor context model aims to align vectors such that similar words and entities occur close to one another in the vector space. Hence, we can measure the similarity between any pairs of words and entities, which are used to estimate probability distributions.

**Disambiguation Model.** We resolve the entity disambiguation as a generative model. Different from the works of Han et.al [11], we use coherence instead of mention name because experiments show that sometimes mention name leads to wrong disambiguation (details in section 4). Given the mention $m$, we first choose a referent entity $e$ from $\mathcal{KB}$, according to the entity prior popularity distribution $P(e)$, then estimate its context according to the textual context distribution $P(\hat{\mathcal{C}}|e)$, and finally generate the coherent entities of the referent entity via the distribution of related entities $P(\mathcal{N}|e)$. Hence, the probability of $m$ referring to a specific entity $e$ in $\mathcal{KB}$ can be inferred as:

$$P(m, e) = P(e) \cdot P(\hat{\mathcal{C}}|e) \cdot P(\mathcal{N}|e) \tag{1}$$

Given the mention $m \in \mathcal{M}$ in a document $\mathcal{D}$ , the final entity we need to find is the one maximizing the post-prior $P(e|m)$. Thus, entity disambiguation can be resolved as following:

$$\begin{aligned} e^* &= \arg\max_{e \in \mathcal{C}} \frac{P(m, e)}{P(m)} \\ &= \arg\max_{e \in \mathcal{C}} P(e) \cdot P(\hat{\mathcal{C}}|e) \cdot P(\mathcal{N}|e) \end{aligned} \tag{2}$$

where $\hat{\mathcal{C}}$ is the set of textual context around mention $m$, $\mathcal{N}$ is the set of disambiguated entities in $\mathcal{D}$, $\mathcal{C} = \{e_1, e_2, ..., e_{|\mathcal{M}|}\}$ is the candidate set.

$P(e)$ is the distribution of entity prior. Especially, we define the entity prior as the probability it is referred in the whole corpus. In large corpus, the more times an entity is referred, the more popular this entity tends to be. However, entities have different prior probabilities in different domains. Thus we introduce a parameter $\alpha \in [0, 1]$to control the influence of prior:

$$P(e) = \left(\frac{|\mathcal{A}_{e,*}|}{|\mathcal{A}_{*,*}|}\right)^{\alpha} \tag{3}$$

where $\mathcal{A}_{e,*}$ is the set of anchors pointing to entity $e$ and $\mathcal{A}_{*,*}$ is the set of all the anchors in $\mathcal{KB}$. $\alpha = 0$ denotes entity prior has no impact on the result and $\alpha = 1$ indicates the case without no control of entity prior.

$P(\hat{\mathcal{C}}|e)$ is the textual context distribution given $e$. An entity is more likely to appear in a context similar with its sense. For example, in text “In 2001, Michael Jordan and others resigned from the Editorial Board of Machine Learning”, we can know the mention “Michael Jordan” refers to the professor *Michael I. Jordan* from the context “Machine Learning”. Hence, following [23], we use a cosine similarity of context word vector and entity vector to estimate the distribution. We average the vectors of context words to represent the context vector:

$$\vec{vc} = \frac{1}{|W_{\hat{C}}|} \sum_{w \in W_{\hat{C}}} \vec{v_w} \tag{4}$$

where $W_{\hat{C}}$ is the set of context words. We obtain the words of context via searching a pre-built word dictionary which is indexed to a trie using Aho - Corasick algorithm. The word dictionary is the same as the words corpus used to train the word embeddings.

$P(\mathcal{N}|e)$ is the distribution of context entities given $e$. Entities in context share one or few same topics, and related entities are close in vector space. Thus the distribution can be seen as the topical *coherence*. Similar to [17] and [23], we adopt a two-step method to calculate *coherence* iteratively. Firstly, we find unambiguous entities as context entities with a popularity $P(e|m) > \varepsilon$, where $P(e|m) = |\mathcal{A}_{e,m}|/|\mathcal{A}_{*,m}|$, and we set $\varepsilon$ to 0.95 empirically. Secondly, we use the predicted unambiguous entities as context entities to calculate $P(\mathcal{N}|e)$ again. To estimate $P(\mathcal{N}|e)$, we use cosine similarity between the vector of context entities and the vector of target entity, namely,

$$\vec{v_N} = \frac{1}{|E_N|} \sum_{e \in E_N} \vec{v_e} \tag{5}$$

where $E_N$ is the unambiguous entities.

It should be noted that there are two ways to perform disambiguation according to the ordering of candidates, namely L2R (left to right) and S2C (simple to complex). L2R is more efficient to apply because there is no need to rank the candidate lists. While S2C need to rank candidate lists again according to the size of candidate list of each mention.

## 4 Experiments

### 4.1 Dataset and Settings

We use CoNLL, a popular named entity disambiguation dataset [12], to test the performance of XLink. It is based on NER data from CoNLL 2003 shared task, and consists of 946, 216 and 231 training, development and test documents respectively. Because our disambiguation method is unsupervised, we test on the entire dataset, including mentions having valid entries to Wikipedia and we ignore mentions with NIL entities. To make the comparison fair, we use a public dataset PPRforNED[1] [16] to generate candidates for the entity disambiguation.

The version of Wikipedia dump we use is Apr. 1, 2016. In preprocess, we first tokenize the Chinese corpus using ANSJ[2] and we discard tokens that appears lower than 5 times. Finally, we learn the representations of 2.9 million words and 5.1 million entities in Wikipedia, 4.5 million words and 5.7 million entities in Baidu Baike. For both Wikipedia and Baiku corpus, the dimension of embedding vectors is 300 and the window size is 10 as suggested by [23]. Negative samples $g = 5$ and learning rate $\eta$ is set to 0.025 which linearly decreases with the iterations of training. We train the models by iterating the corpus 10 times.

### 4.2 Evaluation of Mention Parsing

We build anchor dictionaries for both Wikipedia and Baidu Baike separately, and achieve 3,922,720 mention-candidates entries in Wikipedia and 2,210,817 entries in Chinese. Because both corpora are comprehensive encyclopedias with

[1] https://github.com/masha-p/PPRforNED
[2] https://github.com/NLPchina/ansj_seg

lots of general terms, the mentions detected may have redundancy, which are not as helpful as specific entities. Thus, we discard the detected mentions simply via a threshold of $link\_prob(m)$. As Figure 2 shows, with threshold grows, the precision decreases and the recall increases. And when the threshold is around 0.04, we have the best F1 score while precision is 0.672 and recall is 0.691.

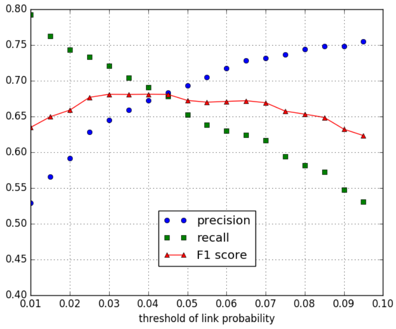


**Fig. 2.** Evaluation of Mention Parsing

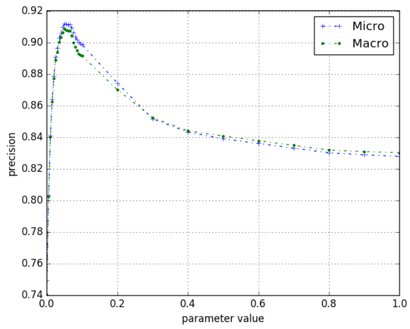


**Fig. 3.** Impact of prior parameter $\alpha$

### 4.3 Evaluation of Disambiguation Entities

To compare our results with state-of-the-art methods, we report Hoffart et al.'s results [12] as they reimplemented two other systems and also ran them over the CoNLL dataset. We also compare with Alhelbawy and Gaizauskas [2] and Shirakawa et al. [19] who carried out their experiments using the same dataset. Additionally, we compare the result in macro-(aggregates over all documents) and micro-(aggregates over all mentions) accuracies. Table 2 shows that our disambiguation result outperforms these state-of-the-art unsupervised methods, and there is no significant difference between L2R and S2C for our system.

**Table 2.** Performance of Unsupervised Methods

| | Cucerzan | Kulkarni | Hoffart | Shirakawa | Alhelbaway | XLink(L2R) | XLink(S2C) |
|---|---|---|---|---|---|---|---|
| MicroP@1 | 0.510 | 0.729 | 0.818 | 0.823 | 0.842 | 0.911 | 0.912 |
| MacroP@1 | 0.437 | 0.767 | 0.819 | 0.830 | 0.875 | 0.908 | 0.909 |

Observing the results, we find our method works well on the cases with bias of popular entities, but can not handle the extreme situations. For example, in text *"Australia will defend the Ashes in a six-test series against England"*, mention "England" refers to the entity *England_cricket_team* but is linked to the entity *England*. Although the value of $P(\mathcal{N}|England\_cricket\_team)$ is 0.9 and $P(\mathcal{N}|England)$ is 0.7 which means the entity has high similarities with context and topical coherence, however, the prior of *England* is 0.032, 46 times of *England cricket team*, thus leading to an error case. Also, we test the performance with the mention name knowledge incorporated in our model. The performance will decline about 2% overall, and the main reason is that it misleads the decisions of entities with priority bias.

Furthermore, to analyze the reason of effectiveness of our model, we conduct experiments to understand how different distributions impact the results. As shown in Table 3, the distribution of $P(e)$ is the major part which achieves the micro-precision of 82.1%. This makes sense because most entities in the dataset are well-known entities. The distribution of $P(\hat{\mathcal{C}}|e)$ promotes the performance 3.8% in micro precision and 3.4% in macro-precision. And $P(\mathcal{N}|e)$ gives a improvement of the performance for 5.3% in micro-precision and 5.0% in macro-precision. The results show that both $P(\hat{\mathcal{C}}|e)$ and $P(\mathcal{N}|e)$ can capture rich information of context, from textual and coherent aspects, and jointly promote the performance of disambiguation.

**Table 3.** Compact of different distributions

| | MicroP@1 | MacroP@1 |
|---|---|---|
| $P(e)$ only | 0.821 | 0.825 |
| $P(e)\&P(\hat{\mathcal{C}}|e)$ | 0.859 | 0.859 |
| $P(e)\&P(\hat{\mathcal{C}}|e)\&P(\mathcal{N}|e)$ | 0.912 | 0.909 |

As we analyzed before, the prior probability greatly affects the result of entity disambiguation. Thus we introduce a parameter to control the importance to the overall probability. Figure 3 shows the disambiguation precision under different values of $\alpha$ on the dataset of CoNLL, and we can see both micro and macro precision increase a lot with increasing of $\alpha$ and decrease quickly after the peaks. Thus we set $\alpha$ to 0.05 experimentally.

### 4.4 Evaluation of Time Efficiency

We test time efficiency of mention parsing and entity disambiguation separately as shown in Figure 4. In mention parsing, the time required is linear of amount of mentions $|\mathcal{M}|$ in the documents. In entity disambiguation, the time needed is linear of the amount of all candidates in a document which is the amount of mentions $|\mathcal{M}|$ multiply average candidate sets size $|\mathcal{C}|$. Thus, the overall time complexity is $\mathcal{O}(|\mathcal{M}| + |\mathcal{M}| \cdot |\mathcal{C}|)$.

## 5 Related Work

As far as we know, the first Web-scale entity linking system is SemTag, built by Dill et al. [8]. With the knowledge sharing communities appearing, such as Wikipedia, and the development of information extraction techniques, such as instance matching [25] and knowledge linkingc̃itepan2016domain, more and more knowledge bases have been constructed automatically, such as YAGO [20], DBpedia [3] and Freebase [4]. These knowledge bases are usually used as resources for entity linking task and many entity linking systems are based on them. Wikify! [14] and Cucerzan [7] are early works employing Wikipedia to identify

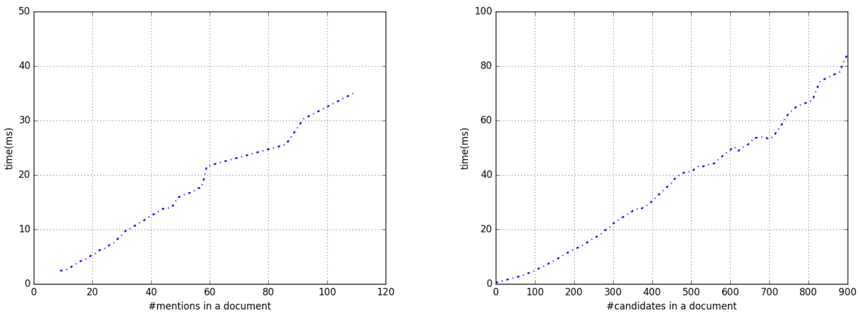


**Fig. 4.** Time cost of Mention Parsing (left) and Entity Disambiguation (right)

and link entities. Wikify! firstly proposed *link probability* to extract keywords as a preprocess of entity disambiguation. Miline&Witten [15] proposed an approach that yielded considerable improvements by hinging on two main ingredients: 1) a measure of relatedness of two pages based on overlap between their in-linking pages in Wikipedia; 2) a notion of coherence between a page and unambiguous pages. Following [15], Ferragina ans Scaiella [9] designed and implemented TagMe, which annotates a short plain-text with pertinent hyperlinks to Wikipedia pages efficiently, and proposed a vote scheme to capture the collective agreement among the entities referred that utilizes the method to compute relatedness in [15]. TagMe showed its competitive performance in short text disambiguation compared with Miline&Witten. Meanwhile, another entity linking system based on DBpedia, called DBpedia Spotlight [13], was proposed, taking full advantage of the DBpedia ontology for specifying which entity should be annotated. It used a part of speech tagger to disregard any spots that are only composed of verbs, adjectives, adverbs and prepositions while spotting. In disambiguation, it weighted the resources of DBpedia using product of term frequency and inverse candidate frequency based on a vector space model. Linkfy [22] is implemented as a script emphasizing the helpfulness of entity linking systems, using supervised machine-learning methods with a broad set of features, including link probability features, entity features, entity class features, topical coherence features, textual features and mention occurrence features. However, it relies on NER tools to recognize entities compared with other systems.

As representation learning becoming a base method to represent semantic elements in nature language preprocessing, increasingly more researchers focus on modeling words and entities to a united space to address the task of entity linking. Yamada et al. [23] learns the embeddings of words and entities separately, then maps them to one space via anchors in Wikipedia. Following, Cao [5] proposes a method to model the representation of mentions, which learns multiple sense embeddings for each mention by jointly modeling words from textual contexts and entities. Compared with traditional methods, representation based methods show their competitiveness in entity disambiguation tasks as our experiments show.

## 6 Conclusion and Future Work

In this paper, we present a bilingual online entity linking system XLink. XLink provides precise and efficient entity linking service in both English and Chinese. We use a pre-built anchor dictionary to detect mentions instead of NER tagger tools, and propose a generative entity disambiguation method to choose a correct entity among the candidate entities. Currently, XLink regards mention parsing and entity disambiguation separately, and we will focus on jointly optimizing these two phases in the future. Also, categories of Wikipedia and Baidu Baike conveys useful domain knowledge, and how to exploit the category information for domain specific entity linking [6] is an interesting future direction.

## Acknowledgements

The work is supported by 973 Program (No. 2014CB340504), NSFC key project (No. 61533018，61661146007), Fund of Online Education Research Center, Ministry of Education (No. 2016ZD102), THUNUS NExT Co-Lab, National Natural Science Foundation of China (Grant No. 61375054) and Natural Science Foundation of Guangdong Province （Grant No. 2014A030313745）。

## References

1. Aho, A.V., Corasick, M.J.: Efficient string matching: an aid to bibliographic search. Communications of the ACM 18(6), 333–340 (1975)
2. Alhelbawy, A., Gaizauskas, R.J.: Graph ranking for collective named entity disambiguation. In: ACL (2). pp. 75–80 (2014)
3. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. The semantic web pp. 722–735 (2007)
4. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. pp. 1247–1250 (2008)
5. Cao, Y., Huang, L., Ji, H., Chen, X., Li, J.: Bridging text and knowledge by learning multi-prototype entity mention embedding. In: Proc. ACL (2017)
6. Cao, Y., Li, J., Guo, X., Bai, S., Ji, H., Tang, J.: Name list only? target entity disambiguation in short texts. In: EMNLP. vol. 15, pp. 654–664 (2015)
7. Cucerzan, S.: Large-scale named entity disambiguation based on wikipedia data. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 708–716 (2007)
8. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., et al.: Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In: Proceedings of the 12th international conference on World Wide Web. pp. 178–186 (2003)

9. Ferragina, P., Scaiella, U.: Fast and accurate annotation of short texts with wikipedia pages. IEEE software 29(1), 70–75 (2012)
10. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd annual meeting on association for computational linguistics. pp. 363–370 (2005)
11. Han, X., Sun, L.: A generative entity-mention model for linking entities with knowledge base. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. pp. 945–954 (2011)
12. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 782–792 (2011)
13. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th international conference on semantic systems. pp. 1–8 (2011)
14. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. pp. 233–242 (2007)
15. Milne, D., Witten, I.H.: Learning to link with wikipedia. In: Proceedings of the 17th ACM conference on Information and knowledge management. pp. 509–518 (2008)
16. Pershina, M., He, Y., Grishman, R.: Personalized page rank for named entity disambiguation. In: HLT-NAACL. pp. 238–243 (2015)
17. Ratinov, L., Roth, D., Downey, D., Anderson, M.: Local and global algorithms for disambiguation to wikipedia. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. pp. 1375–1384 (2011)
18. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: Issues, techniques, and solutions. IEEE Transactions on Knowledge and Data Engineering 27(2), 443–460 (2015)
19. Shirakawa, M., Wang, H., Song, Y., Wang, Z., Nakayama, K., Hara, T., Nishio, S.: Entity disambiguation based on a probabilistic taxonomy. Microsoft Research, Seattle, WA, USA, Tech. Rep. MSR-TR-2011-125 (2011)
20. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: WWW (2007)
21. Weston, J., Bordes, A., Chopra, S., Rush, A.M., van Merriënboer, B., Joulin, A., Mikolov, T.: Towards ai-complete question answering: A set of prerequisite toy tasks. arXiv preprint arXiv:1502.05698 (2015)
22. Yamada, I., Ito, T., Usami, S., Takagi, S., Takeda, H., Takefuji, Y.: Evaluating the helpfulness of linked entities to readers. In: Proceedings of the 25th ACM conference on Hypertext and social media. pp. 169–178 (2014)
23. Yamada, I., Shindo, H., Takeda, H., Takefuji, Y.: Joint learning of the embedding of words and entities for named entity disambiguation. arXiv preprint arXiv:1601.01343 (2016)
24. Yao, X., Van Durme, B.: Information extraction over structured data: Question answering with freebase. In: ACL (1). pp. 956–966 (2014)
25. Zhang, Y., Jin, H., Pan, L., Li, J.Z.: Rimom results for oaei 2016. In: OM@ ISWC. pp. 210–216 (2016)