Short Text Entity Linking with Fine-grained Topics

Lihan Chen[12], Jiaqing Liang[13], Chenhao Xie[13], Yanghua Xiao[14*]
[1]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[2]CETC Big Data Research Institute Co.,Ltd., Guizhou, China
[3]Shuyan Technology, Shanghai, China
[4]Alibaba Group, Zhejiang, China
{lhc825,l.j.q.light,redreamality}@gmail.com,shawyh@fudan.edu.cn

ABSTRACT

A wide range of web corpora are in the form of short text, such as QA queries, search queries and news titles. Entity linking for these short texts is quite important. Most of supervised approaches are not effective for short text entity linking. The training data for supervised approaches are not suitable for short text and insufficient for low-resourced languages. Previous unsupervised methods are incapable of handling the sparsity and noisy problem of short text. We try to solve the problem by mapping the sparse short text to a topic space. We notice that the concepts of entities have rich topic information and characterize entities in a very fine-grained granularity. Hence, we use the concepts of entities as topics to explicitly represent the context, which helps improve the performance of entity linking for short text. We leverage our linking approach to segment the short text semantically, and build a system for short entity text recognition and linking. Our entity linking approach exhibits the state-of-the-art performance on several datasets for the realistic short text entity linking problem.

CCS CONCEPTS

• **Information systems** → **Information extraction**;

KEYWORDS

entity linking; short text; concepts; fine-grained topics

ACM Reference Format:
Lihan Chen, Jiaqing Liang, Chenhao Xie, Yanghua Xiao. 2018. Short Text Entity Linking with Fine-grained Topics. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3269206.3271809

*Yanghua Xiao is corresponding author. This paper was supported by National Key R&D Program of China under No.2017YFC0803700, by National Key Basic Research Program of China under No.2015CB358800, by National Key R&D Program of China under No.2017YFC1201200, by the National NSFC (No.61472085, No.U1509213, No.U1636207), by Shanghai Municipal Science and Technology Commission foundation key project under No.15JC1400900, by Shanghai Municipal Science and Technology project (No.16511102102, No.16JC1420401), Shanghai STCSM's R&D Program under Grant (No.16JC1420400)

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3271809

1 INTRODUCTION

Entity linking is a fundamental task in lots of real applications including information extraction [17] and question answering [34]. The goal of **entity linking**, also known as **entity disambiguation**, is to link the mentions in text, like "Steve", to the correct entity in knowledge graph, like *Steve Jobs* in Wikipedia. The mention is always ambiguous, for example, "Steve" can also be *Steve Wozniak* or many other entities. Thus, how to use the context to disambiguate the mention is the major problem in entity linking.

Entity linking for short text is quite important. A wide range of web corpora are in the form of short text, such as QA queries, search queries and news titles. The applications on these short text such as question answering system and search engine cannot function normally if the mentions are not correctly linked. In this paper, *we focus on the entity linking problem for short texts.* Under this circumstance, the input text, in most cases, is only a short sentence such as "Li Na won 51 singles matches and lost just four in this year", or sometimes even a phrase such as "Li Na's signature song". The contexts of mentions in short text are always scarce. Therefore, the challenge of short text entity linking lies in making full use of context words. Consider previous examples above, the tennis player *Li Na* is inferred from words like "won" and "single matches"; the entity *Li Na (singer)* is inferred from words like "song".

1.1 Weakness of Previous Approaches

Lots of solutions have been proposed for entity linking. In general, these solutions can be divided into two categories, **local approaches** and **global approaches** [32]. The local approaches try to disambiguate each mention in a document separately, utilizing clues such as the textual similarity between the mention context and the candidate entity information; and the global approaches utilize relations of entities in the context to estimate coherence. Local and global approaches could use the *supervised* or *unsupervised* learning paradigms. But most of existing approaches are not effective for short text entity linking.

Weakness of previous entity linking approaches. Supervised approaches recently attract wide research interests [11, 24, 27]. However, the training data for learning are limited in English and unsuitable for short text scenario. First, *there are no high-quality training data for low-resourced languages.* The well-annotated linking data rely on lots of manpower. In English, the AIDA-CoNLL dataset [18] and the anchor texts in the Wikipedia articles are widely-used as training data for supervised entity linking methods. However, in many low-resourced languages (e.g., Chinese), there are no manually-annotated dataset and the anchor texts in the encyclopedia are either scarce or of low quality, and the translation from

English usually incurs low-quality texts. Thus the supervised entity linking methods are hard to function effectively in low-resourced languages. Second, *the existing training data for entity linking are not suitable for the short text setting*. The contexts of the AIDA-CoNLL dataset and the anchor text in Wikipedia articles are long texts, where most disambiguation signals rely on rich context information and global signals such as attributes of the page and links of the page. In our short text entity linking task, the training data of long context are not suitable. Therefore, it is a strong urge for an effective unsupervised methods for short text entity linking on low-resourced languages.

Previous unsupervised methods are incapable of handling the sparsity and noisy problem of short text. Text-based comparison adopted by local approaches will fail because of the sparsity. Most unsupervised approaches are global, they take advantage of entities in the context and disambiguate by their coherence [3, 6, 29]. Some topic-based entity linking models [14] make inference by implicit document topics. These methods heavily rely on a large amount of entities in the rich context. But in short text scenario, there are usually very few entities in the context. In fact, nearly half of the texts contains only 1 entity in search queries [39]. Thus short text scenario poses great obstacles for global approaches.

Weakness of previous short text entity linking approaches. There are a few previous works on short text entity linking. TAGME [10] is a well-known entity annotation system that is effective on short fragments of text. But its disambiguation strategy merely relies on a global voting mechanism of other mentions in the context, which incurs the same problem of global approaches. Thus, it's essential to build connection between words and entities for short text entity linking. Instinctively, word-entity relatedness can be built by their co-occurrence [4], while relevance of words is not considered, that is, if a word w_1 is relevant to an entity e and w_2 is close in meaning to w_1, it is likely that w_2 is relevant to e as well. Training embeddings for entities [11, 15] is a feasible solution. However, the embeddings of long-tailed entities with few information and occurrences, which are the majority of the entire entity set, will be underestimated. An alternative solution is to represent the entity as the centroid of word vectors of its relevant words [5]. Nevertheless, relevant words of entities are noisy and representations by word vector are implicit, which will make the model unpredictable.

1.2 Our Idea and Contribution

We try to solve the problem by mapping the sparse short text to a topic space. In most cases, the insufficient contexts can only provide vague topic information. For example, in the text "read Harry Potter", the common word "read" can be hardly extracted as a keyword from any entities named "Harry Potter". However, from this word, we can infer that the topic of the context is about a *literature* instead of a *movie*. Even though we don't know which exact "Harry Potter" it refers to, the topic coherence principle here can be used as an important signal for the model to prefer entities related with literature to those with the movie. Existing topic models like LDA can generate implicit topics for words, while they are difficult to capture the fine-grained characteristics of entities. Specifically, the implicit topics can detect the relevance of word "read" and topic literature, while it is difficult for them to relate the topic *literature* to the book *Harry Potter*.

We notice that the concepts of entities have rich topic information and characterize entities in a very fine-grained granularity. Hence, we propose to use the concepts of entities as topics to explicitly represent the context. For the example above, the book *Harry Potter* has the concept of *book*. The relevance of the concept *book* and the word "read" can be mined from large entity-related corpus, where the word "read" tends to occur more frequently with the entities with concept *book*. It also works well for the long-tailed entities which have little information. Even if the knowledge in book *Harry Potter* has no relevant information about "read", it can be also related to "read" by means of the concept *book*.

There are three main contributions in this work. First, we propose a comprehensive solution for short text entity recognition and linking. Second, we introduce concepts of entities as explicit fine-grained topics to solve the sparsity and noisy problem of short text. Third, we incorporate our entity linking module into the text segmentation algorithm and solve the problem of unawareness of the semantic, which improve the text segmentation and entity recognition.

2 PROBLEM STATEMENT

In this section, we first formalize our problem. Then we present the framework of our solution.

2.1 Problem Definition

Definition 1. *An **entity mention** is a fragment of text that refers to an entity in the knowledge base.*

Definition 2. *A **segmentation** p of a text s is a sequence of text fragments $p = \{t_i | i = 1, ..., l\}$ such that:*

- *text fragments cannot overlap with each other;*
- *concatenation of all text fragments in p equals s.*

In practical applications, the entity linking task is always preceded by entity recognition process. Our task is to detect entity mentions in an input short text and link them to the correct entities in the knowledge base, including entity recognition and entity linking. We formalize our problem as follows.

Definition 3. ***Entity annotation for short text**. Given a text sequence s, the goal is to detect all the entity mentions in s as $M = (m_1, ..., m_N)$ and produce a mapping from the set of mentions to the set of knowledge base entities $E = \{e_1, ..., e_{|E|}\}$. We denote the output as two N-tuples, M and $\Gamma = (e_1, ..., e_N)$ where e_i is the output entity for mention m_i.*

Local and Global Entity Linking. The essential problem of our entity annotation task is linking the ambiguous entity mentions to the correct entities in the knowledge base they refers to. This process is **entity linking**, which is also known as **entity disambiguation**. In general, the entity linking approaches are divided into two categories, the local and the global approaches [32].

A local entity linking approach disambiguates each mention m_i separately. Specifically, a local approach solves the following optimization problem:

$$\Gamma_l = arg \max_{\Gamma} \sum_{i=1}^{N} \phi(m_i, e_i) \quad (1)$$

$\phi(m_i, e_j)$ is a score function reflecting the likelihood that mention $m_i \in M$ links to the candidate entity $e_j \in E$, which is usually calculated by the textual similarity of the context of m_i and the information of e_j in the knowledge base.

Besides the content of local approaches, global approaches additionally considers that the correct disambiguations should be coherent with each other. The global approaches attempt to solve the problem of:

$$\Gamma_g = arg \max_{\Gamma} [\sum_{i=1}^{N} \phi(m_i, e_i) + \psi(\Gamma)] \quad (2)$$

The coherence function ψ is designed to measure the coherence of all the entities in the context, which usually takes advantage of the similarity and the relatedness of entities in Γ [3, 18, 32]. Compared to local approaches, global entity linking approaches are more advanced because the coherence of entities in the context is a very strong signal for entity linking. For example, in sentence "David and Victoria added spice to their marriage", "David" and "Victoria" can be easily disambiguated by the relation between *David Beckham* and *Victoria Beckham*.

2.2 Solution Framework

We adopt the global approach for entity linking, because it has been proven to be effective in previous solutions [11, 18, 32]. Since multiple mentions in a short text context are not very common; under some circumstances like search queries, texts with only one mention account for nearly a half [39]. Therefore, a robust local approach is also necessary for our task. Besides, there are many subproblems including text segmentation, entity recognition and disambiguation in our task. We develop a comprehensive framework to handle all the above issues in a systematical way.

Algorithm 1: Entity annotation for short texts

Input: The input text sequence, S; the knowledge base, K
Output: The detected mention and linked entity ensembles $\{(m_i, e_i)\}$

1 Detect all the possible candidate entity mentions from S and construct possible mention set $M = \{m_i\}$;
2 **for** *each* $m_i \in M$ **do**
3   Get the m_i's candidate entity set $E_i = \{e_{ik}\}$ from knowledge base K;
4   Compute the linking score $\phi(m_i, e_{ik})$ for all the candidate entities $e_{ik} \in E_i$;
5 Segment S into the segmentation of the maximum probability;
6 Recognize the entity mentions in the segmented text;
7 Perform global entity linking on every segmented mention m_i to the most probable entity e_i, and return the detected mentions and linked entities.

Algorithm 1 illustrates the procedure of our generic solution for the short text entity annotation problem. For an input example below:

Example 2.1. steve met mona simpson in big apple

The first mention detection step (line 1 in Algorithm 1) detects all the possible mentions in the input text, including "steve", "met", "mona simpson", "mona", "simpson", "big apple" and "apple". The second step (line 2-4) is local disambiguation, where we generate candidate entities for each mention and calculate the local disambiguation scores for each possible mention, such as $\phi(''steve'', \textit{Steve Wozniak}) = 0.4$, $\phi(''steve'', \textit{Steve Jobs}) = 0.3$. The third step (line 5) is text segmentation, where we segment the text as "steve, met, mona simpson, in, big apple". The fourth step is entity recognition (line 6), where we recognize the entities "steve", "mona simpson" and "big apple". At last we perform global disambiguation (line 7) and link the mentions to the right entities, "steve" to *Steve Jobs*, "mona simpson" to *Mona Simpson*, "big apple" to *New York City*.

In this work, we develop our own entity recognition for several reasons. First, current popular entity recognition tools focus on named entities, while in our application scenarios, there are many other types of entities, such as events such as *June Fourth Incident* and some nominals such as *martial law*. Second, more often than not, the short texts are grammatical incomplete, like a short text search query "france versus who world cup 1998". Therefore the entity recognition tools on complete sentences perform quite poorly in the short text scenario. Third, we find that the text segmentation and entity recognition task can also benefit from entity linking results. Naive text segmentation methods do not consider the semantic of the segments, and we leverage the semantic information from our local disambiguation results to segment the text semantically.

In the solution above, various of external knowledge should be acquired and processed beforehand. Hence our framework, as shown in Figure 1, comprises two parts, **online text processing** and **offline knowledge preprocessing**. The online text processing part, corresponding to the solution in Algorithm 1, contains 3 components:

- **segment resolution** includes mention detection (line 1) and local disambiguation (line 2-4), where mention-to-entity tables and entity information are needed from preprocessing.
- **entity recognition** includes text segmentation (line 5) and entity recognition (line 6), taking the local disambiguation results and the phrase priors from offline mining as inputs.
- **global entity disambiguation** (line 7) needs the entity coherence information which is calculated by the entity relations from knowledge base and linking entities from encyclopedia.

The sources of our offline preprocessing part include a knowledge base, such as YAGO [36] and CN-DBpedia [41], and a web corpus, such as Wikipedia articles and *Baidu Baike* articles. In the next two sections, we will describe the details for the online and offline parts, respectively.

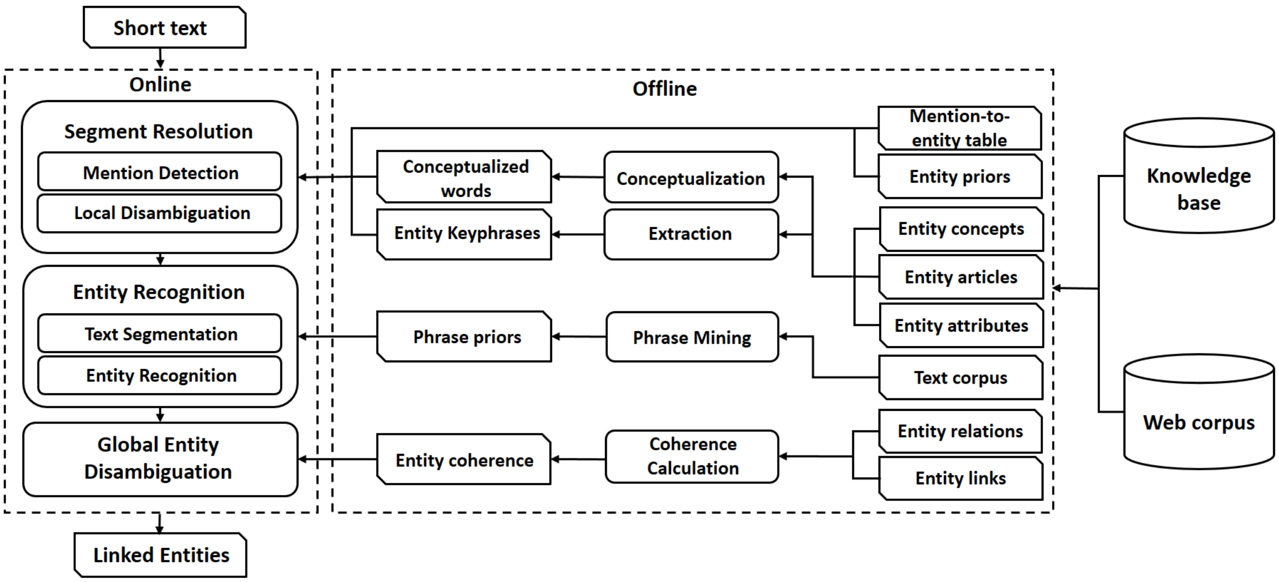

Figure 1: Solution Framework

3 ONLINE TEXT PROCESSING

In general, there are three core components in the online processing, **segment resolution**, **entity recognition** and **global entity disambiguation**. In segment resolution component, we first perform a preliminary local disambiguation for each possible mentions. Then in entity recognition component, we leverage the local disambiguation results for semantic text segmentation and then recognize the entities. In the last global disambiguation component, we utilize a global linking approach to decide the final linking results.

3.1 Segment resolution

In this component, we resolve all the segments that are possible entity mentions in the text and link them to the most possible entity.

First of all, we identify all the possible mentions from input sequence S. And for each detected mention $m_k \in M$, the candidate entity set is generated, which is usually generated from the redirect and disambiguation pages from encyclopedia, the knowledge base dictionary [18] and mention-entity prior statistically estimated from the anchor texts linked to the encyclopedia articles [35].

Since the structure of the input text are not clear and all we need is just a preliminary entity linking. We adopt a local entity disambiguation approach here, which equals to taking the disambiguation of each mention as a separate task. The task of the local entity disambiguation for a single mention m is to find the entity $e \in E_k$ that has the highest compatibility score ϕ. We model it as:

$$\phi(m, e) = f(m, e) \cdot g(m, e) \tag{3}$$

The first term $f(m, e)$ is the context-independent score, which mostly relies on the surface form of the mention and the knowledge about the candidate entity; the second term $g(m, e)$ is the context-dependent score, which is the compatibility score between the context of m and the information of entity e.

3.1.1 *Context-independent score $f(m, e)$*

This score is often given as an mention-entity prior $\hat{p}(e|m)$, which is statistically estimated from the anchor texts linked to the Wikipedia articles [35]. However, there are two problems with the direct estimation. First, the anchor text links are usually noisy and incomplete. There are plenty of false and missing links, especially in low-resourced languages. Second, the prior probability is only reliable for the popular entities. For long-tailed entities or entities that occur less frequently, the mention-entity prior is sparse and underestimated. Therefore we propose a novel context-independent score.

Since the mention-entity prior $\hat{p}(e|m)$ is noisy and unreliable, the candidate entity set is usually combined with other sources like YAGO dictionary [18] and Wikipedia redirect and disambiguation pages, which provide a more reliable mention-to-entity relation. We combine those reliable sources and construct a candidate entity set E_m for the mention m. We combine the sources and rewrite the context-independent score as:

$$f(m, e) = \hat{p}(e|m) \cdot sim(m, e) + \frac{\beta}{|E_m|} \tag{4}$$

We give a uniform probability with weight β to the reliable candidate entities, and we use the string similarity score $sim(m, e)$ to constrain the noisy prior. The high similarity of a mention m and entity e leads to high confidence of the mention-to-entity mapping. But sometimes a very different alias from the entity name is also reasonable. Therefore the heuristic for the string similarity score is:

$$sim(m, e) = 1 - w_s \cdot \frac{ed(m, e)}{\max(|m|, |e|)} \tag{5}$$

We use a weighted edit distance $ed()$ as the string comparison function. As in [28], we define different cost for different operations in edit distance. We define the cost of the deletion from e and insertion to m as a small weight in $(0, 1)$, because the mention tends to be an abbreviation of an entity name and the abbreviation is always as good as an alias. The parameter $w_s \in [0, 1]$ is to control the influence of string comparison. In the case where $w_s = 1$, the similarity score is a standard edit distance string similarity; in the case when $w_s = 0$, the score is irrelevant to the string comparison.

3.1.2 Context-dependent score $g(m, e)$. The textual similarity between the context around the entity mention and the document associated with the candidate entity is the most intuitive feature for the context-dependent score. Previous works have used many approaches to calculate textual similarity score, including TF-IDF similarity [32] and keyphrase-based similarity [18].

However, the short text context is usually sparse and noisy, unable to perform word-level matching with entity keywords. Deep learning methods based on entity and word embedding [11] seems to be a feasible solution for the sparse context. But recall that there are no suitable training data for either low-resourced languages or the short text entity linking. Therefore, it is essential to exploit the semantic relatedness between the entity and the short sparse text context in an unsupervised manner.

Topic coherence. The insufficient contexts can only provide vague topic information. As discussed in section 1.2, we can only disambiguate the mentions by inferring the topic of the context in many cases. To determine whether the candidate entity is coherent with the context in topic, the context text and the candidate entity should be represented as topic vectors. Existing topic models like LDA can generate implicit topics for words, while they are difficult to capture the fine-grained topics of entities. We notice that the concepts of entities have rich topic information and can characterize entities in a very fine-grained granularity. Entities with the same concept tend to have the same topic about that concept, so do their descriptions and context texts of their mentions. For example, *Jay Chou* have concept *singer*, so its description or context texts will have high probability for words like *sing* or *album*. Since the concepts of entities are known from knowledge base, we need to explicitly represent the mention context as a vector of concepts, also known as conceptualization [19, 40]. The concept vector of the context is a summary of words vector with distance weight:

$$\mathbf{v}_c(m) = \sum_{w \in CT(m)} \frac{\mathbf{v}_c(w)}{D(w, m)} \tag{6}$$

$CT(m)$ is the context word set of mention m. $D(w, m)$ is the distance function, which penalizes the influence of the words far away from the mention. The computation of concept vectors of words $\mathbf{v}_c(w)$ will be discussed in Section 4. And we can acquire the concept vector of the entity $\mathbf{v}_c(e)$. The topic coherence score sim_c is computed as:

$$sim_c(m, e) = \cos(\mathbf{v}_c(m), \mathbf{v}_c(e)) \tag{7}$$

Textual similarity. As mentioned above, the short text context are always sparse and will make textual comparison methods perform poorly. Specifically, traditional textual comparison does not take into account the semantic similarity of words.

In the textual similarity calculation, we relieve the problem of the sparse context by exploiting the semantic relation between words. Word2vec [26] is a technique of training word embeddings in large corpus by their co-occurrence. It's been proven that the word vectors trained contain semantic information, which gives us an approach to compare words instead of simple keyword matching. We adopt the GloVe [30] word vectors, which is an improvement of the original word vectors.

Our assumption here is that only a few words that appear in the context of a mention are discriminative for entity disambiguation, which is similar to the ideas in [11, 23]. For example, in the sentence "There are many famous actors in the Lord of the Rings", the word "actors" is the most discriminative word for the disambiguation of "the Lord of the Rings", which is also the most related context word to the correct candidate entity, and the influence of other words is negligible. Hence, we calculate the textual similarity by the most discriminative word in the context. We define the textual similarity score sim_t as follows.

$$sim_t(m, e) = \max_{w_c \in CT(m), w_d \in KP(e)} \cos(\mathbf{v}_w(w_c), \mathbf{v}_w(w_d))$$

$KP(e)$ denotes the keyphrase set of entity e, whose detail will be discussed in Section 4. $\mathbf{v}_w(\cdot)$ denotes the word's vector, for a phrase, we use the centroid of all the words' vector.

So, we consider both the topic coherence and textual similarity as our signals for disambiguation.

$$g(m, e) = sim_t(m, e) \cdot (1 - \epsilon + \epsilon \cdot sim_c(m, e)) \tag{8}$$

Parameter ϵ is used for smoothing as well as to control the influence of the topic coherence score.

3.2 Entity Recognition

After resolution of all the segments, the text should be segmented properly before the real entities are recognized without conflicts. The entity mentions detected from above section are all possible mentions, they may overlap or destroy the syntactic structure of the text. For example, for text "new york times square", the mentions "new york", "new york times" and "times square" will be detected, but the second one conflicts with others. Hence, the input text should be properly segmented for recognizing correct entities.

3.2.1 Text segmentation. Let $T = t_1, t_2, \ldots$ be a segmentation of input text sequence p as defined above. We use the framework of jieba segmentation [21] by maximizing the following maximum likelihood objective to get the best segmentation.

$$T_{best} = arg \max_T \log P(T) = \sum_{i=1}^{|T|} \log P(t_i) \tag{9}$$

where, $P(t_i)$ is the prior probability of the segment t_i, and is estimated with the phrase prior by the offline phrase mining:

$$P(t_i) = \frac{freq(t_i)}{\sum_j freq(t_j)} \tag{10}$$

$freq(t_i)$ is the frequency of the phrase t_i.

3.2.2 Entity recognition. The text segmentation framework above is derived merely by maximizing the likelihood of the segmentation. It ignores the semantic compatibility between the segment and the context. For example, in short text "hotel california reservation", if we only consider the statistical probability, "hotel california", as a well-known song, will be recognized as an appropriate segment. However, in this case, the segment "hotel california" is not a mention of the song, and the two words should be separated apart. In fact, the local disambiguation score between the mention "hotel california" and the song entity is low because the context is irrelevant, which is a good signal for the text segmentation. Therefore, for a candidate mention segment, the calculation of the segment prior probability should consider the local entity linking score ϕ.

If segment t_i is a possible mention of an entity, i.e. $t_i = m_k$, we consider using the compatibility score ϕ of the best candidate entity e_k as a constraint on the prior probability of t_i. However, the possible mention can also be a non-entity word. For example, "watch" can be either an entity mention, or a verb. So, we first use a pos-tagging tool to detect the type of all the words before text segmentation. And regard a segment as an entity mention only if its pos-tagging is nominal. If a segment is identified as a mention, its prior probability is calculated as following.

$$P(t_i) = \frac{pop(e_k)}{\sum_j pop(e_j)} \cdot \lambda\phi(m_k, e_k) \tag{11}$$

where, e_k is the linking result of local disambiguation, which has the highest ϕ score in all the candidate entities of m_k. λ is the parameter for controlling the influence of ϕ. The term $pop(e)$ is the popularity of the entity e. There are several measurements for entity's popularity, such as number of views of the entity page in encyclopedia or the number of linked entities of the entity page. We use the entity's page view as the popularity of the entity in our experiments. The denominator is the sum of all the entity popularity, which takes the same form of phrase prior. The higher the entity linking probability is, the more likely the segment t_i will be segmented as a mention.

With the probability for each segment, the segmentation with maximum likelihood objective can be solved by dynamic programming in time complexity of $O(|p|^2)$.

After the best segmentation is found, the mention segments are recognized as entities.

3.3 Global Entity Disambiguation

The preliminary entity disambiguation in Section 3.1 is a local entity disambiguation approach which only considers local text information around the mention. When all the mentions in the text are recognized, a global entity disambiguation can help capture entity coherence among mentions co-occurred . Since the local score function is well-defined before, the main task in this section is to define the coherence function ψ. Following previous works [3, 18, 32], we define the function as the sum of the pairwise coherence between two entities $coh(e_i, e_j)$. Then the global disambiguation objective Eq 2 is formulated as:

$$\Gamma_g = arg\max_{\Gamma}[\sum_{i=1}^{N} \phi(m_i, e_i) + \sum_{e_i \in \Gamma, e_j \in \Gamma} coh(e_i, e_j)] \tag{12}$$

The calculation of the entity coherence coh will be discussed in section 4.3. With the coherence function ψ defined above, the global entity disambiguation is an NP-hard optimization problem [32], which most of previous works try to optimize. However, here in our scenario, the time cost will no longer be a problem due to two reasons. First, the number of mentions in short text context are quite limited. Second, we perform a local disambiguation beforehand, we filter out the candidate entities of each mention with low local disambiguation scores in the global disambiguation phase. In practice, we keep the search space of Γ below 1000. Specially, we take candidate entities with top-10 local disambiguation score for each mention if there is 3 mentions in the context ($N = 3$); we take top-31 of the first two if the third mention has only 1 candidate entity. Because the entity coherence coh is calculated offline and N is always small in short text scenario (≤ 5), the calculation is quite efficient.

4 OFFLINE KNOWLEDGE PREPROCESSING

The offline knowledge preprocessing consists of three main components, as **word conceptualization**, **entity keyphrase extraction** and **entity relatedness calculation**.

4.1 Word Conceptualization

In this section, we describe how to conceptualize a word to a concept vector.

Conceptualization. As mentioned before, we consider concepts as topics. Hence, the article of an entity is about the topics of its concepts. Then we calculate the relatedness of a word w and a concept c as:

$$r(w, c) = \sum_{e \in E} \frac{n(w, e) \cdot r(e, c)}{\sum r(e, c')} \tag{13}$$

$n(w, e)$ denotes the count of word w occurring in the article of entity e. $r(e, c)$ denotes the weight of entity e and concept c, which is 1 for the knowledge base without concept weights.

Concept clustering. The concept space is rather large (millions), and many concepts implies the same topic, like *singer*, *male singer* and *female singer*. We first employ the K-Medoids algorithm to cluster the concepts into clusters [19]. The distance of two concepts is defined as:

$$d(c_1, c_2) = 1 - \cos(E(c_1), E(c_2)) \tag{14}$$

$E(c)$ is the entity distribution of concept c. After clustering, each topic is represented as a concept cluster $C = (c_1, c_2, ...)$. The word-concept relatedness and entity-concept weight are aggregated at the cluster level and so is the concept vector, which means each dimension corresponds to a concept cluster instead of a single concept.

4.2 Entity Keyphrase Extraction

The knowledge base contains rich information for each entity, for example, the corresponding article, attributes and category information. These are the inputs for an offline data-mining step to determine characteristic keyphrases for each entity. For an entity e, its keyphrase set $KP(e)$ consists of following three parts.

Concepts. Concepts of an entity are very descriptive features. We take the concepts of an entity as a part of the its keyphrase set.

Attributes. Entity's attributes are also strong features for the entity. We sort the attribute names and values by their inverse document frequency and use the top ones as keyphrases.

Article. The entity article in the encyclopedia contains rich information of the entity, while it's very noisy. We rank the words by a modified TF-IDF manner. For each word w, we weight its relevance with the entity e by a BM25 [33] manner:

$$r(w,e) = \frac{n(w,D_e)\cdot(k_1+1)}{n(w,D_e)+k_1\cdot(1-b+b\cdot|D_e|/avgdl)}\cdot IDF(w) \quad (15)$$

where D_e is the article of e and $avgdl$ is the average length of all the articles in the encyclopedia. We set $k_1 = 2$ and $b = 0.75$. Since the words tend to be noisy when the article is long. For the frequency score $n(w, D_e)$, we only consider the occurrence of the word in the article.

$$n(w,D_e) = \sum_{para\in D_e}\frac{n(w,para)}{\sqrt{offset(para)}} \quad (16)$$

where $para \in D_e$ denotes a paragraph in D_e and $n(w, para)$ is the word count of w in $para$, $offset(para)$ is the offset of the $para$ in D_e, which we use to punish the words occurring later in the article because we notice that words occur in the front of the article are usually more relevant.

4.3 Entity Coherence Calculation

The coherence between entities should be calculated for the global entity disambiguation. The coherence in the global entity disambiguation means entities tend to co-occur with relevant or similar entities. Therefore the entity coherence function coh should reflect the relevance or similarity between entities. The coherence of two entities can be measured from different perspectives, one of the most popular measurements is the negative form of Normalized Google Distance [27]:

$$sim(e_1,e_2) = 1 - \frac{log(max(|E_1|,|E_2|)) - log(|E_1\cap E_2|)}{log(|E|) - log(min(|E_1|,|E_2|))} \quad (17)$$

where E_1 and E_2 are the inlink entity set of e_1 and e_2, which are collected from the corresponding encyclopedia article of the entity; E is the complete set of entities. In fact, the negative Normalized Google Distance implies the entity similarity by comparing entities' inlink sets, which is hard to capture the relatedness of two entities. Hence, we additionally consider the direct relation between two entities to capture relatedness between entities.

$$rel'(e_1,e_2) = \sum_{r\in R(e_1,e_2)}\frac{2}{|T(e_1,r)|+|H(r,e_2)|} \quad (18)$$

where $R(e_1, e_2)$ denotes the relation set from entity e_1 to e_2, $H(r, e)$ denotes the head entity set of relation r and tail entity e and $T(e, r)$ denotes the tail entity set of head entity e and relation r. The intuition of this equation is that, relations with multiple entities tend to imply a weak relatedness (e.g. *nationality*), while relations with a single entity may indicate strong connection (e.g. *spouse*). Since the function rel' is sensitive to the position of e_1 and e_2, we use the maximum function to combine the two directions because the reverse relation always implies a duplicate relation, for example, *found* and *foundedBy*.

$$rel(e_1,e_2) = \max(rel'(e_1,e_2), rel'(e_2,e_1)) \quad (19)$$

Combining above two coherence metrics, we write the coherence function as:

$$coh(e_1,e_2) = \gamma\cdot rel(e_1,e_2) + (1-\gamma)\cdot sim(e_1,e_2) \quad (20)$$

The parameter $\gamma \in [0, 1]$ is the weight to balance between relatedness and similarity.

5 EXPERIMENTS

Recall that the short text entity annotation problem is a composed task of text segmentation, entity recognition and entity disambiguation. In this section, we first conduct experiments to prove that our method has the state-of-the-art performance in the short text entity linking task, for both English and low-resourced language. Second, we prove that our semantic text segmentation method is robust and outperforms the popular NLP tools in text segmentation task. Third, we prove that our approach also has robust performance on the entity recognition task.

5.1 Experiment Setup

We conduct experiments for both English and Chinese. For English, we use YAGO [36] as the knowledge base and articles in Wikipedia as our web corpus. For Chinese, we use CN-DBpedia [41] as the knowledge base and articles in *Baidu Baike* as the web corpus. Because *Baidu Baike* and CN-DBpedia contain over 10 million entities, which have much wider coverage than Chinese Wikipedia (around 1 million entities). Nevertheless, the data quality in *Baidu Baike* and CN-DBpedia are low. The anchor text in the entity articles are sparse and noisy, thus can hardly be used as training data for the supervised models. Hence, Chinese is still considered as a low-resourced language in entity linking task.

The concepts of entities are crucial for our approach. For Chinese setting, we use the concepts provided by CN-Probase [37]; for English setting, we use the YAGO's entity types as concepts of entities.

5.2 Effectiveness of Short Text Entity Linking

Existing Datasets in English. Our approach focuses on entity linking problem for short text, including search queries, question answering queries, news titles and so on. Most of the existing datasets on entity linking are based on long text articles, which are not suitable for the task of short text entity linking. We find two English datasets that are suitable for our short text entity linking task.

- KORE50 [16] dataset. It contains 50 short sentences with highly ambiguous entity mentions, which is considered amongst the most challenging dataset for entity disambiguation. Average sentence length (after stop word removal) is 6.88 words per sentence and each sentence has 2.96 entity mentions on an average. Every mention has an average of 631 candidates to disambiguate in YAGO knowledge base [36].
- Webscope [5] dataset. This dataset is constructed from Yahoo's Webscope search query log. It is a publicly available editorial set that contains annotations for 2531 queries distributed across 980 user sessions.

Our Datasets in Chinese. We try to prove that our approach is robust in low-resourced languages, like Chinese. Since there are no existing datasets for short text entity linking, we construct our Chinese datasets for short text entity linking. The entity linking evaluation data of NLPCC 2015 [9] is a good dataset for short text entity linking. Unfortunately, its annotation is not published, therefore we relabeled a subset of it with knowledge base CN-DBpedia [41], and we additionally create three datasets for short text entity linking based on three aspects of short text entity application. Each sample is manually annotated with entities of knowledge base CN-DBpedia by 3 volunteers, and the majority is adopted. The following are the details of those datasets[1].

- NLPCC dataset. The entity linking evaluation data of *NLPCC 2015* consists of 3849 samples of short search queries, which is suitable for the short text entity linking task. We re-annotate a subset of 500 random samples as our evaluation set.
- NTF dataset. News texts, specifically *news titles and first sentences*, are the important sources of short texts. We crawled texts from the new titles and the first several sentences of news from several Chinese news website. We took a single title or sentence as a text sample and invited volunteers to annotate. The sample number is 1299.
- HQA dataset. Knowledge base question answering is an important application of the short text entity linking. Question answering queries can be very ambiguous and noisy. We select 486 *hard question answering* queries with ambiguous mentions as an entity linking test dataset. Since the KBQA task is based on triples of entities, there may be multiple correct answers for a mention. For example, the mention "Dream of the Red Chamber" in question "Who is the director of Dream of the Red Chamber" could refer to many versions of movies or TV show, any linking to those which have the right attribute for the QA task are correct answers.
- CNDL dataset. The daily languages we use are always short for convenience and accordingly ambiguous. Therefore, disambiguating the short daily languages is challenging and meaningful. The CNDL dataset consists of 343 *Chinese daily language* sentences with ambiguous mentions.

Comparison methods. As for the KORE50 and Webscope datasets, there are many results of other methods for comparison. Since the four Chinese datasets above are newly built, we re-implemented several state-of-the-art entity linking methods and used existing APIs for comparison. Chinese is a low-resourced language, which means there are few high-quality annotated data for training. Hence, the comparison methods on Chinese datasets are unsupervised. The details of comparison methods are below.

- Commonness (CMN). It is the mention-entity prior statistically estimated from the anchor texts linked to the encyclopedia articles, which is a popular and strong baseline in many datasets.
- Relation Inference (RI) [6]. This method performs relation inference while linking the entity. It's been proved to be a very strong baseline in several entity linking datasets. We re-implement it based on knowledge base CN-DBpedia and test it on our datasets.
- Random Walk (RW) [3]. This method solves the entity linking task with random walk on the knowledge graph. It has very good performance on several datasets, which is a typical global linking approach. We re-implement the method and test it on our datasets and the Webscope dataset.
- Fast Entity Linker (FEL) [5]. FEL is a classic short text entity linking method, which outperforms all the previous solutions in the Webscope dataset. It represents the entity as the centroid of word vectors of its relevant words, and uses the entity embedding to calculate the relevance between entities and context. We re-implement and test it on our datasets and the KORE50 dataset.
- Baidu Entity Annotation (BEA). It's an API service provided by Baidu [2], which has a fair performance on Chinese short text entity linking. We use their API and test on our dataset.
- *Our method.* We use our method and adjust the parameters based on a small amount of validation data besides of the test set.

Table 1: Evaluation on our Chinese datasets.

Model	NTF	NLPCC	HQA	CNDL
CMN	84.9	83.6	57.2	45.0
RI [6]	81.2	83.3	52.3	42.2
RW [3]	72.3	73.1	42.6	45.7
FEL [5]	93.2	92.8	70.3	66.6
BEA [2]	90.4	87.8	64.7	53.0
Our method	**96.6**	**96.1**	**78.5**	**75.1**

Table 2: Evaluation on English datasets.

Model	KORE50	Webscope
CMN	44.4	73.4
TAGME [10]	-	66.8
AIDA [18]	57.6	-
AIDA-KORE [16]	64.6	-
Joint-DiSER-TopN [1]	71.8	-
RW [3]	55.6	68.6
FEL [5]	54.2	83.5
Our Method	**76.4**	**86.4**

Measures. The key measures in our evaluation are precision and recall. Precision is the fraction of mention-entity assignments that match the ground-truth assignment. Recall is the fraction of the ground-truth assignments that the method could compute. Both measures can aggregate over of all mentions (across all texts) or over all input texts (each with several mentions). The former is called micro-averaging, the latter macro-averaging. We choose Mirco-F1 score as our measure for the entity linking task.

Results. We present the entity linking evaluation results of our Chinese datasets in Table 1. We can see that the classic global entity linking approaches RI and RW performs poorly when the contexts are insufficient. Compared to other linking methods, our method has the best performance.

The evaluation results of English datasets are shown in Table 2. Although the context texts are short, the KORE50 datasets have

[1]The datasets are available at https://github.com/cdx666/dataset-cnel

many co-occurring entities. That makes the global approaches consider entity coherence like RW and Joint-DiSER-TopN has better performance than the local approaches like FEL. In the Webscope dataset, the texts are shorter and have few co-occurring entities, the robust local approach is dominant. Our approach outperforms others in both datasets.

5.3 Effectiveness of Text Segmentation

We test the text segmentation performance of our approach on the datasets published in [8]. We choose the two simplified Chinese segmentation datasets, MSR (Microsoft Research) and PKU (Peking University), as our evaluation dataset. We compare our approach with several popular Chinese segmentation tools, including:

- jieba [21]. It is a Chinese segmentation tool based on a prefix dictionary structure to achieve efficient word graph scanning. It uses dynamic programming to find the most probable combination based on the word frequency. For unknown words, a HMM-based model is used with the Viterbi algorithm.
- SnowNLP [20]. It is a simplified Chinese text processing tool including text segmentation. It apply a character-based generative model [38] for text segmentation.

As our segmentation approach is entity-oriented, it will produce many entity name segments which are reasonable but not in the standard answer. For example, our approach recognizes "reform and opening" as an entity segment while in the standard answer of PKU it is segmented apart. Hence, we invited volunteers to relabeled the conflicting segmentations our system outputs and construct modified versions of those datasets. We invited 3 volunteers to evaluate the conflicting segmentations and adopt the majority. And we will release these relabeled datasets, named MSR-Seg-E and PKU-Seg-E, respectively.

The measurement of text segmentation is the precision and recall of the segmenting points. The results are shown in Figure 3. Our approach has the best precision and F1 score.

Table 3: Chinese text segmentation results

	MSR-Seg-E			PKU-Seg-E		
Methods	Prec.	Reca.	F1	Prec.	Reca.	F1
Jieba	89.3	96.2	92.6	93.4	93.6	93.5
SnowNLP	84.4	**96.4**	87.1	94.5	**94.5**	90.7
Our Method	**96.2**	94.7	**95.4**	**97.8**	91.7	**94.7**

5.4 Effectiveness of Entity Recognition

As an entity annotation task, the effectiveness of entity recognition is essential. Our approach also has good performance on entity recognition task. The four datasets (NLPCC, NTF, HQA and CNDL) we published are also measurements for entity recognition task, from which we select 1037 samples that are difficult for entity recognition as our evaluation data for entity recognition task.

We compare our systems with existing popular entity recognition tools and API services on the entity recognition evaluation data. The results are shown in Table 4. As mentioned above, the entities we are concerned about are more than named entities, so our entity recognition task is different from NER problem. To make the tasks as consistent as possible, we filter out the results of Standford NER by the types we are not concerned about, such as 'NUMBER', 'MONEY', 'DATE', 'ORDINAL', 'TIME' and 'PERCENT'. This ensure that the output types of NER method is a subset of entities in our dataset. But there are still some types of entities in our dataset do not belong to named entities. Therefore, the recall score of the NER method tends to be low, while the precision score is confident. Our method has the best performance in both precision and recall.

Table 4: Results of entity recognition task

Methods	Prec.	Reca.	F1
Stanford NER [25]	58.7	39.5	47.2
Baidu Entity Annotation [2]	65.5	78.2	71.3
Our Method	**91.0**	**89.4**	**90.2**

6 RELATED WORK

In this section, we discuss related work in three aspects: entity linking, short text conceptualization, and text segmentation.

Entity Linking. Most previous entity linking methods are incapable of handling the sparsity and noisy problem of short text. Deep models [11, 13, 42] rely on massive anchor text in Wikipedia or high-quality manually-annotated data. However, the training data for learning are limited in English and unsuitable for short text scenario. Traditional approaches [27, 32] rely on superior feature selections. For local approaches, text-based comparison features such as TFIDF [32] and keyphrase matching [18] cannot handle the sparsity of the short context. Global approaches [3, 6], which heavily rely on the relations among the entities in the context, perform poorly in short text scenario, where entities in the context are usually insufficient.

Some researches have been conducted on short text entity linking. TAGME's [10] disambiguation strategy merely relies on the a global voting mechanism of other mentions in the context, which will still suffer from the same problem of global approaches. [5] try to solve the problem by representing the entity as the centroid of word vectors of its relevant words. However, relevant words of entities are noisy and representations by word vector are implicit, which will make the model unpredictable. There are some works on entity linking for tweets [7, 12, 24], while the methods and features used in those are similar to previous works, which suffer from the same problem as mentioned above.

Some works have similar ideas on exploiting type information of entities to help entity linking [13, 31]. Instead of incorporating entity type into a deep model, we use entity concepts as explicitly topics to bridge entities and words in an unsupervised manner.

Short Text Conceptualization. Conceptualization is a kind of explicit representation for the short text. Short text understanding is very challenging because short text is usually very sparse, noisy, and ambiguous. Conceptualization [19, 40] models try to map this short text to a concept space, and then a comprehensive concept vector can be generated to represent this short text. Unlike the previous works of short text conceptualization just represent the text as a concept vector, we use the conceptualization to help entity recognition and linking, which is a more specific task. Concepts reflect fine-grained characters of entities, and have rich topic information, which can be used to build connection between entities and words.

Text Segmentation. The goal of segmentation is to divide a short text into a sequence of meaningful components. Most short text process approaches like [22] adopt the naive longest-cover method for text segmentation, that is, it prefers the longest terms in a given vocabulary. The longest-cover method does not consider the global structure of the text and cannot handle the problem of overlapping, like "new york times square". Some models [21, 38] are used to segment the text for a global optimization objective with statistics prior. But they are unable to understand the semantics of a short text, and fails in cases like "reservation hotel california". Thus, a good approach to short text segmentation must take semantics into consideration.

7 CONCLUSION

We propose a comprehensive approach for short text entity recognition and linking. Our approach is capable of detecting local topics in short texts and linking entities to knowledge base with extremely little context. We introduce concepts of entities as explicit fine-grained topics to solve the sparsity and noisy problem of short text. We incorporate our entity linking module into the text segmentation algorithm and solve the problem of unawareness of the semantic, which improve the text segmentation and entity recognition. Experiments have proved that our approach has the state-of-the-art performance in the short-text entity linking, text segmentation and entity recognition.

REFERENCES

[1] Nitish Aggarwal and Paul Buitelaar. 2014. Wikipedia-based Distributional Semantics for Entity Relatedness.
[2] Baidu. 2017. Baidu Entity Annotation. https://ai.baidu.com/tech/cognitive/entity_annotation. Accessed January 4, 2018.
[3] Denilson Barbosa. 2017. Robust Named Entity Disambiguation with Random Walks.
[4] Sumit Bhatia and Anshu Jain. 2016. Context Sensitive Entity Linking of Search Queries in Enterprise Knowledge Graphs. In *ESWC*.
[5] Roi Blanco, Giuseppe Ottaviano, and Edgar Meij. 2015. Fast and Space-Efficient Entity Linking for Queries. In *WSDM*.
[6] Xiao Cheng and Dan Roth. 2013. Relational Inference for Wikification. In *EMNLP*.
[7] Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for tweets. *Inf. Process. Manage.* 51 (2015), 32–49.
[8] Thomas Emerson. 2005. The Second International Chinese Word Segmentation Bakeoff. In *SIGHAN@IJCNLP 2005*.
[9] Yansong Feng, Zhe Han, and Kun Zhang. 2015. Overview of the NLPCC 2015 Shared : Entity Recognition and Linking in Search Queries. In *NLPCC*.
[10] Paolo Ferragina and Ugo Scaiella. 2010. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM*.
[11] Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep Joint Entity Disambiguation with Local Neural Attention. In *EMNLP*.
[12] Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To Link or Not to Link? A Study on End-to-End Tweet Entity Linking. In *HLT-NAACL*.
[13] Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity Linking via Joint Encoding of Types, Descriptions, and Context. In *EMNLP*.
[14] Xianpei Han and Le Sun. 2012. An Entity-Topic Model for Entity Linking. In *EMNLP-CoNLL*.
[15] Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. Learning Entity Representation for Entity Disambiguation. In *ACL*.
[16] Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. 2012. KORE: keyphrase overlap relatedness for entity disambiguation. In *CIKM*.
[17] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artif. Intell.* 194 (2013), 28–61.
[18] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust Disambiguation of Named Entities in Text. In *EMNLP*.
[19] Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. 2015. Short text understanding through lexical-semantic analysis. *2015 IEEE 31st International Conference on Data Engineering* (2015), 495–506.
[20] isnowfy. 2013-2014. SnowNLP. https://github.com/isnowfy/snownlp. Accessed May 4, 2018.
[21] Sun Junyi. 2013. jieba segmentation. https://github.com/fxsjy/jieba. Accessed May 4, 2018.
[22] Dongwoo Kim, Haixun Wang, and Alice H. Oh. 2013. Context-Dependent Conceptualization. In *IJCAI*.
[23] Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2015. Plato: A Selective Context Model for Entity Resolution. *TACL* 3 (2015), 503–515.
[24] Xiaohua Liu, Yitong Li, Haocheng Wu, Ming Zhou, Furu Wei, and Yi Lu. 2013. Entity Linking for Tweets. In *ACL*.
[25] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL*.
[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*.
[27] David N. Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *CIKM*.
[28] Erwan Moreau, François Yvon, and Olivier Cappé. 2008. Robust Similarity Measures for Named Entities Matching. In *COLING*.
[29] Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *TACL* 2 (2014), 231–244.
[30] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*.
[31] Jonathan Raiman and Olivier Raiman. 2018. DeepType: Multilingual Entity Linking by Neural Type System Evolution. *CoRR* abs/1802.01021 (2018).
[32] Lev-Arie Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and Global Algorithms for Disambiguation to Wikipedia. In *ACL*.
[33] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.
[34] Denis Savenkov and Eugene Agichtein. 2016. When a Knowledge Base Is Not Enough: Question Answering over Knowledge Bases with External Text Data. In *SIGIR*.
[35] Valentin I. Spitkovsky and Angel X. Chang. 2012. A Cross-Lingual Dictionary for English Wikipedia Concepts. In *LREC*.
[36] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*.
[37] Shuyan Tech. 2018. CNProbase concept api. http://shuyantech.com/api/cnprobase/concept. Accessed May 22, 2018.
[38] Kun Wang, Chengqing Zong, and Keh-Yih Su. 2009. Which is More Suitable for Chinese Word Segmentation, the Generative Model or the Discriminative One?. In *PACLIC*.
[39] Zhongyuan Wang, Haixun Wang, and Zhirui Hu. 2014. Head, modifier, and constraint detection in short texts. *2014 IEEE 30th International Conference on Data Engineering* (2014), 280–291.
[40] Zhongyuan Wang, Kejun Zhao, Haixun Wang, Xiaofeng Meng, and Ji-Rong Wen. 2015. Query Understanding through Knowledge-Based Conceptualization. In *IJCAI*.
[41] Bo Xu, Yong Xu, Jiaqing Liang, Chenhao Xie, Bin Liang, Wanyun Cui, and Yanghua Xiao. 2017. CN-DBpedia: A Never-Ending Chinese Knowledge Extraction System. In *IEA/AIE*.
[42] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. In *CoNLL*.