IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 22, 2017, accepted December 20, 2017, date of publication January 4, 2018, date of current version March 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2787787

Entity Linking: An Issue to Extract Corresponding Entity With Knowledge Base

GONGQING WU, (Member, IEEE), YING HE, AND XUEGANG HU
Hefei University of Technology , Hefei 230009, China

Corresponding authors: Gongqing Wu (wugq@hfut.edu.cn) and Ying He (ying_he@mail.hfut.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1000901, in part by the Program for Changjiang Scholars and Innovative Research Team in University of the Ministry of Education under Grant IRT17R32, and in part by the National Natural Science Foundation of China under Grant 61673152.

ABSTRACT Entity linking is a task to extract query mentions in documents, and then link them to their corresponding entities in a knowledge base. It can improve the performances of knowledge network construction, knowledge fusion, information retrieval, natural language processing, and knowledge base population. In this paper, we introduce the difficulties and applications of entity linking and focus on the main methods to address this issue. At last, we list the knowledge bases, data sets, and the evaluation criterion and some challenges of entity linking.

INDEX TERMS Datasets, entity linking, evaluation criterion, knowledge base.

I. INTRODUCTION

In this era of rapid development of the Internet, a large amount of data is generated. Big data has become a symbol of the era [1]–[3]. As a consequence, how to get useful knowledge from a large amount of data is a popular topic in current research, more importantly, entity linking plays a key role as a part in this process [4], [5].

Most information we obtain from the Internet in daily life is in the form of web texts. These texts contain a large number of named entities (e.g. person, organization, and place) which are the basic elements of texts. However, these entities are highly ambiguous, so we need to link them to an existing knowledge base so that people can know what the entities refer to and understand the texts more correctly. On the other hand, with the development of the Google Knowledge Graph, automatic knowledge base construction becomes more and more important. The automatic knowledge base construction need to extract information such as entities and relationships between entities from web texts and add them to the knowledge base. Before filling, the most important step is to disambiguate those entities extracted by the system. We call this process as named entity linking or entity linking. Entity linking is a task of linking named entities in web texts to their corresponding entities in a knowledge base (e.g. Wikipedia [6], DBpedia [7], and YAGO [8]). For example, given a text ''Nadal presents a bouquet to Li Na at her retirement ceremony. Li Na runs with tears and enjoys cheers of the whole audience. . .'', entity linking will link the query mention Li Na to her corresponding entity Li Na (tennis player) in the knowledge base rather than Li Na (Professor of Peking University) using a variety of algorithms. Through this process, we can understand that the content of this article is about the tennis player Li Na. Specially; ''we'' not only refers to the human, but also refers to the search engine, artificial intelligence machine, question answering system and so on.

Entity linking is difficult due to the high ambiguity of entity mentions, which includes polysemy and multiword synonym. Polysemy refers to that an entity mention is corresponding to a number of entity concepts. For example, the entity mention ''Li Na'' can refer to Li Na (tennis star) and Li Na (Professor of Peking University). Multiword synonym is that an entity may have many kinds of surface forms, for example, Yao Ming (basketball player) has many alias, such as the moving Great Wall, little giant, Dayao. These ambiguities make it difficult to us to understand the meaning of entity mentions.

The application of entity linking involves many fields, such as search engine retrieval [9]–[11], knowledge fusion [12], [13], knowledge base population [4], [14], [15]. Given a scenario of a hospital with two medical doctors A and B sharing the same name, let us assume an application in which a patient is able to search for information about a doctor. As soon as the patient types the name of the doctor, the system retrieves two profile records. As a consequence, the patient could be doubtful about which record is the correct

one. By means of entity linking, we can help search engines to disambiguate so that the correct searching results are closer to the top of pages of the searching results. In the field of knowledge fusion, entity linking also plays an important role. When we fuse entities from different databases to a unified database, actually, some entities are the same with each other in the expressive meaning but have different surface forms, so it requests us to map these entities into the same entity in the knowledge base first. Then, the information of the entities can be fused. Meanwhile, the retrieval efficiency can be also improved. In addition, knowledge base population has become a popular topic in recent years. The mission of it is to extract new information scattered on the web and then fill the relevant entities into the existing knowledge base. For this purpose, the first phase is exactly to complete the entity linking task. Through entity linking we can determine which entity the information we extract belongs to.

This article will be introduced from the following five aspects. Firstly, we compare entity linking to other similar works in Section II. Then in Section III, we give the definition and framework of an entity linking system. Section IV introduces some main methods of entity linking, including two phases: the candidate entity generation and disambiguation. Next we review the methods of NIL clustering. In the end, the paper introduces knowledge bases, datasets and evaluation methods.

II. FROM NAMED ENTITY RECOGNITION TO ENTITY LINKING

A. NAMED ENTITY RECOGNITION

Named entity recognition is a task of identifying important nouns in the text. The so-called important nouns are person, organization, place and all other entities that are identified by names which are the key point for people to understand the meaning of the text. We call those important nouns named entities [5], [16], [17].

Different from entity linking, named entity recognition only need to identify named entities in the text, and determine their categories. It does not need to know the meaning of these entities, also do not need to disambiguate these entities using a knowledge base. Whereas in the task of entity linking, the named entity disambiguation is regarded as an essential step which can affect the result of entity linking [18], [19].

The Methods of named entity recognition can be divided into rules-based method [17], [20], [21] and statistical method [22]–[25]. We can also combine these two methods to deal with this problem [26]–[28]. A rule-based method needs linguists establish a series of rules according to the structure of the texts, then recognize named entities using pattern matching technique or string matching technique. This kind of method often depends on a specific language environment, rules in the system must be reconstructed when the required corpus changes. In contrast, a statistical method does not require extensive linguistic knowledge. It trains a language model through machine learning methods, and then the named entities in the text are automatically recognized by this model. It requires a large-scale corpus for training model but does not need domain knowledge for rules. There are many available named entity systems which are introduced in paper [5].

B. COREFERENCE RESOLUTION

In a text, an entity often has different expressions, such as in the text ''Li Na is a Chinese tennis player, she was born in Hubei, Wuhan'', ''Li Na'' and ''she'' all refer to the same entity. If we identify this information we can extract that Li Na was born in Hubei, Wuhan. So, identifying those equivalent descriptions of the same entity in a text is very important for understanding the meaning of a text completely.

Coreference resolution refers to dividing different expressions which point to the same real-world entity into the same equivalence set by using contextual information and background knowledge [29], [30]. The equivalence set is called coreference chains. The divided expressions are called mentions. In coreference resolution, the mentions contain nouns, proper nouns and pronouns. The research of coreference resolution is earlier than entity linking. Their difference is that the coreference resolution only clusters the mentions in the text; it does not need to link them to a knowledge base to get more attribute information.

The method of coreference resolution is developed from the rule-based method [31], [32] to the learning-based method [33]–[37]. Hobbs algorithm is one of the earliest rule-based algorithms. It analyzes texts according to their syntactic structure [31]. The central theory proposed by Grosz *et al.* [32] is based on the text structure, which has been used by many scholars in the work of coreference resolution. More details about the rule-based method are introduced in paper [29]. In 1995, coreference resolution was first considered as a binary classification problem, and over the past two decades, classification, clustering, and other learning-based methods have been proposed. More details about learning-based method can be found in paper [29] and [30].

C. WORD SENSE DISAMBIGUATION

Word sense disambiguation is very similar to entity linking. It refers to selecting a correct meaning for a word in a particular context. Words in texts are always assumed to have corresponding senses in a dictionary [38], [39]. The difference is that entity linking deals with named entities and it maps them to the corresponding entities in a knowledge base. Moreover, due to the incompleteness of the knowledge base, the entity mentions may not have the corresponding entities in the knowledge base [40].

Moro *et al.* [41] divide methods of word sense disambiguation into three categories which are supervised methods, unsupervised methods, and knowledge-based methods. The supervised methods utilize different features to train a classifier, such as a decision tree [42], a support vector machine [43] and so on. These features are extracted from manually sense-annotated corpus which takes a lot of labors. On the contrary, the unsupervised methods do not need to

annotate the corpus and do not need the support of dictionaries. For example, some of these methods transform the problem of word sense disambiguation into clustering problems [44], [45]. Firstly, the corpus is clustered by features, and then the meanings of new words are classified. The knowledge-based methods also do not need any sense-annotated corpus. They rely on external knowledge resources such as WordNet [46] to perform their methods.

D. ENTITY LINKING

Entity linking also known as named entity disambiguation. The task of named entity linking refers to map named entity mentions in the text to their corresponding entities in a knowledge base.

After overcoming the problem of recognizing an entity in a text, the attentions of researches are changed to entity disambiguation. As early as the 1990s, coreference resolution and word sense disambiguation are the important tasks of Natural Language Processing. They deal with disambiguation of words in text so as to achieve the purpose of accurately understanding the meaning of the text. In 2006 Cucerzan [47] propose a method which uses the Wikipedia knowledge base to do entity disambiguation. Different from word sense disambiguation, it adds the treatment of proper nouns and considers the meanings of target entities. This work is considered to be one of the early works of entity linking. In 2009 the TAC conference re-defines entity linking [48], and after that, entity linking has always been one of the TAC evaluation tasks. At present, entity linking is far from solved.

III. ENTITY LINKING DEFINITION AND SYSTEM FRAMEWORK

In this section, we will introduce some relevant concepts of entity linking.

Entity linking: entity linking is a task which links a query mention in a text to its corresponding entity in a knowledge base. The common phases of entity linking include: candidate entity generation, candidate entity disambiguation and linking result. Given an entity linking task, the first step is to recognize named entities (the recognized named entities are called query mention or mention in this paper) in the documents using named entity recognition tools (e.g. Stanford NER system [49]) and generate the candidate entities for each mention. Then the next step is to disambiguate these candidate entities with a knowledge base. At last, the system returns the ID of the corresponding entity in the knowledge base or NIL (a label that indicate that there is no matching entity in the knowledge).

Formal description of entity linking: Given a set of documents $d = \{d_1, d_2 \ldots\}$ and a knowledge base K, we can get a mention set $M = \{m_1, m_2 \ldots\}$ using the named entity tool. For each $m_i \in M$, we can get a candidate set $C = \{c_1, c_2 \ldots\}$ from a knowledge base. The goal of entity linking is to choose an entity from C, if each $score(c_i)$, $c_i \in C$, is below τ (τ is a threshold), then the target entity e is NIL, otherwise, m will be linked to C where $score(c) = max(score(c_i))$ [10].

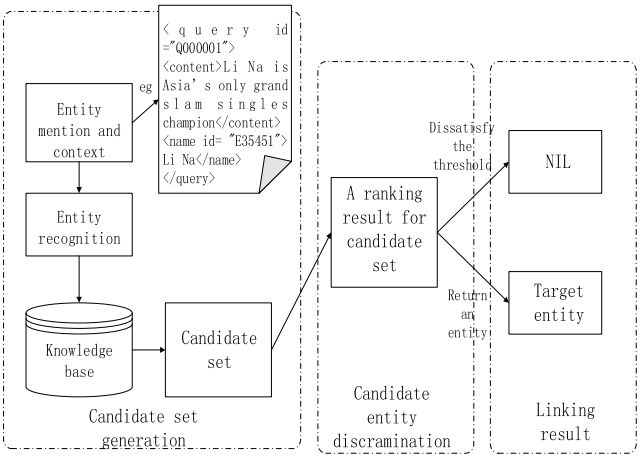

FIGURE 1. A general model of entity linking.

As shown in Fig. 1, the general model of an entity linking system includes the following three parts: candidate entity generation, candidate entity disambiguation and result selection. In following, we will give some concepts of entity linking task and describe this three modules.

Query mention: the surface form of named entity in text. We also call it mention as alternative.

Text: some query texts which contain many named entities. It can be divided into long texts and short texts. The long texts usually contain more than 400 words, such as news articles and the short texts contain an average of less than 200 words, such as tweets [50].

The candidate entity generation module: this module is to select the candidate entities for each query mention in text. It first uses named entity recognition tools to identify the entity mentions and then use the name of entities in conjunction with other features to find the candidate entities in the knowledge base. At last, we can get a series of related entities, for example, the candidate entities for Li Na including Li Na (tennis player), Li Na (professor of the Peking University) and Li Na (singer).

The candidate entity disambiguation module: this module is an important stage of entity linking. It utilizes different methods fusing various features of entities to rank the candidate entities. Features used in the candidate disambiguation phase include: entity popularity [51], [52], entity type [4], [53], the similarity between a query mention and the name of candidate entity [4], [14], the similarity between context of query mention and candidate entity text [54], [55], topic similarity and combination of several features [56], [57]. For example, Ceccarelli *et al.* [58] exploit 27 different features in their article, and classify these features into three kinds of features: singleton, asymmetric feature and symmetric feature.

The linking result module: this module is to select the target entity through the ranking result of the candidate entity disambiguation phase. As the last module shown in Fig. 1, when the score of all candidate entities is below the threshold,

the system will return NIL. However, the threshold is usually set manually, which will easily lead to the problem that the correct target entity is judged to be below the threshold. Thus, some researches work for this issue. Zheng *et al.* [59] use a machine learning method to classify the highest ranked candidate entity, judging whether the highest ranked entity is the target entity. When the highest ranked entity is classified as the positive instance, it will be regarded as the target entity, otherwise, there is no corresponding target entity mapping with the query mention. In another case, the candidate entity with the highest score is selected as the target entity among those candidate entities which satisfy the threshold.

IV. ENTITY LINKING METHOD

In this section, we will review the existing methods in detail. Firstly, we will give an example of ''Li Na'' that we are going to use.

EXAMPLE 1. *An example of ''Li Na'' for entity linking*
Text: *Nadal presents a bouquet to Li Na at her retirement ceremony. Li Na runs with tears and enjoys cheers of the whole audience, and Jiang Shan accompanies her in the ceremonial process.*
Mention*: Li Na, Nadal, Jiang Shan.*
Knowledge base*: A knowledge base released by Tsinghua University*.[1]

In EXAMPLE 1, the Mention is generated by the NLPIR word segmentation system[2]; we select named entities according to the tags of words. The knowledge base is released by Tsinghua University which contains over 800,000 different entities.

A. CANDIDATE ENTITY GENERATION

1) METHOD BASED ON DICTIONARY

One of the most popular ways to address the candidate entity generation problem is a lexical method, which needs to construct a name dictionary based on the information of a knowledge base [54], [60]–[65]. Each name is a key in the dictionary and has a value set of possible entities mapping with it. For example, the mention Michael Jordan has a set of possible entities, such as Michael Jordan (football player), Michael Jordan (machine learning scholar) and so on. Some researches utilize the Wikipedia sources like the titles of entity pages, the titles of redirecting pages, the disambiguation pages and the hyperlinks in Wikipedia articles extracted from Wikipedia to build a surface form dictionary for each entity [54], [60], [61]. In contrast to the process of building dictionaries, some systems use dictionaries that have already been established. Chong *et al.* [62] use the Google lexicon as their dictionary to identify candidate entities. The lexicon lists possible mentions for each entity along with the occurrence probability $p(e|m)$. After building a dictionary, different methods are used to get the candidate entities. For each query mention, systems retrieve the key field of the dictionary. If the key of the dictionary meets the requirement, then the corresponding value set of this key will be added into the candidate entity set of this query mention. In these methods, exact matching as an easy way has been used by many researches to match the query mention and the entity surface form in the knowledge base [54], [64]. Shen *et al.* [64] exploit a direct matching method to match the query mention and the key field where the query mention and the key are exact match with each other. In addition, they reduce the candidate entities by the popularity of entities. However, exact match will cause a low recall rate. Hence, some researches use loose matching methods instead of exact matching [63]. There are many common rules often being used in the loose matching phase, such as the string similarity.

Unfortunately, sometimes we can get nothing from a knowledge base for various reasons such as the misspelling of query mention or the query mention is an acronym of an entity. Zhang *et al.* [60] utilize ''did you mean'' of Wikipedia and Wikipedia search engine to get more information. When the query mention is misspelled, the ''did you mean'' can give a suggestion and the Wikipedia search engine can give some entities pages for the unpopular query mention. When a query mention is an acronym of an entity name, besides building a dictionary, Zhang *et al.* [65] add a process of handling acronym and propose two rules to deal with acronym expansion.

2) METHOD BASED ON DIRECTLY SEARCH

Instead of building a dictionary, some researches search the knowledge base directly [14], [66]. Usually, they create an index for their knowledge base for fast search. Some rules are used in this matching step. We list them as follows:

The entity name in the knowledge base is exactly match with the query mention (e.g. name and query mention are both ''Li Na'').

The first letters of the entity name match the query mention (e.g. The query mention is MJ and the entity name is Michael Jordan).

The alias or nickname of the entity is match with the query mention (e.g. Michael Jordan (football player) with his nickname Air Jordan).

The string similarity between entity name and query mention (e. g. Jaccard distance, N-gram distance).

3) METHOD BASED ON PROBABILITY

Some researches use the empirical probability $p(e|m)$ to select candidate entities [67], [68] . Where m is a query mention and e is an entity in a knowledge base. They obtain $p(e|m)$ by deriving from the Wikipedia hyperlinks. Usually, a higher value of $p(e|m)$ indicates a higher likelihood that e can be selected as a candidate entity of m. Give an entity e and a query mention m, the $p(e|m)$ is defined as follows:

$$p(e|m) = \frac{count(m, e)}{count(m)} \tag{1}$$

[1] http://keg.cs.tsinghua.edu.cn/project/ChineseKB
[2] http://ictclas.nlpir.org/downloads

Where $count(m, e)$ is the number of the mention as a link anchor links to entity e, count(m) is the number of the mention as a link anchor in Wikipedia.

B. CANDIDATE ENTITY DISAMBIGUATION

The goal of candidate entity disambiguation phase is to rank the candidate entities selected from the candidate entity generation phase. The general approach is combining different features listed in Section III to rank the candidate entities.

1) METHOD BASED ON SIMILARITY COMPUTATION

The method based on context similarity computation is a direct way to deal with the disambiguation problem. Generally, it compares the surrounding context of the query mention with the candidate entity context in a knowledge base. In the phase of computing the similarity, for each query mention and candidate entity, their contexts will be expressed as a bag of words or some key words of the contexts will be represented as a vector [47], [55], [69]. Then heterogeneous approach will be used to calculate the similarity between two vectors or bag sets, such as cosine similarity, Jaccard similarity and Dice coefficient. Bunescu and Pasca [55] propose a context similarity computing method, in which they chose the entity with the maximum similarity score as the target entity. First, they model the query mention context and the candidate entity Wikipedia page as two bags of words respectively. After that, they calculate the similarity with the cosine similarity model, where the cosine similarity is defined as follows:

$$S_{cosine} = \frac{V_{query} \cdot V_{entity}}{\|V_{query}\| * \|V_{entity}\|} \tag{2}$$

Where V_{query} is the text vector of the query text, V_{entity} is the Wikipedia page vector of each candidate entity. $V_{query} \cdot V_{entity}$ is the inner product of V_{query} and V_{entity}. $||V_{query}||$ and $||V_{entity}||$ are the lengths of the two vectors respectively.

TABLE 1. Cosine similarity between the mentions and the candidate entities.

Mention	Candidate Entities	Cosine Similarity
Li Na	Li Na(Tennis Player)	0.704
	Li Na(Professor of PKU)	0
	Li Na(Singer)	0
Nadal	Nadal(Tennis Player)	0
	Nadal(Photographer)	0
Jiang Shan	Jiang Shan(Tennis Player)	0.609

In Table 1, it shows the cosine similarity between the mentions in the query text and their candidate entities of Example 1. The candidate entities Li Na (Tennis Player) and Jiang Shan (Tennis Player) obtain high scores, so they will be chosen as the target entities for Li Na and Jiang Shan. However, the score of Nadal (Tennis Player) is zero because of the low value of co-occurrence information. In fact, Nadal (Tennis Player) is the target entity as we know.

For the reason that the method based on context similarity computation depends on the co-occurrence of words too much. Meanwhile, Li *et al.* [71] discover that the existing knowledge base cannot provide enough contextual information for entity linking due to the rapidly growing of data and information. They propose a method to mine evidences for entity linking, in which they extract some documents related to the candidate entity via the Google Search API and reason their labels as additional evidences of candidate entities. Hence, the above problem can be solved to some extent.

As the method of context similarity computation regards the context as some common words, it ignores the semantic relation between words. Milne and Witten [72] propose an entity similarity computing method based on Wikipedia link message named WLM (Wikipedia Link-based Measure). Ratinov *et al.* [73] utilize WLM along with a well-known method PMI (Point-wise Mutual Information) to calculate the correlation degree between the query mention and the candidate entity. Specially, the query mention and the candidate entity are two Wikipedia entities. The two measures are defined as follows:

$$WLM = 1 - \frac{\log(\max(|M_1|, |M_2|)) - \log(|M_1 \bigcap M_2|)}{\log(|W| - \log(\min(|M_1|, |M_2|)))} \tag{3}$$

$$PMI = \frac{|M_1| \bigcap M_2|/|W|}{(|M_1|/|W|)*(|M_2|/|W|)} \tag{4}$$

Where M_1, M_2 are two sets of Wikipedia articles which contain two entities. W is the set of all Wikipedia articles.

Sometimes, the query mention is a novel entity, and there are few links between the query mention page and its corresponding entity. In this situation, the method based on Wikipedia link message will perform poorly.

2) METHOD BASED ON MACHINE LEARNING

In methods based on machine learning, researchers usually use some <mention, entity> pairs to train a binary classification model, deciding whether a candidate entity is a positive instance. Zhang *et al.* [60] choose the SVM classifier as their binary classification model and use three features to represent the <query, entity> pair which are lexical features, word category pair and named entity type. They train their model by using a large scale of data generated automatically. These training data are formed by the <query, entity> pairs. They label these pairs by judging whether the Wikipedia article contains links between the mention in the query and the entity (Each article represents an entity or concept in Wikipedia). Pilz and Paaß [74] also consider this task as a binary classification process. They utilize the Latent Dirichlet Allocation model to get the probability distributions $P(e)$ and $P(m)$ where $P(e)$ and $P(m)$ represent the probability distribution of K topics in the entity text and the mention text respectively. Then they use the thematic distances of $P(e)$ and $P(m)$ as the feature vectors of SVM classifier to find the target entity.

Other researchers also use Naive Bayes [70], C4.5 [72], and Binary Logistic classifier [54] as the classification model.

Since the classification method will generate more than one positive instance, many systems cast the disambiguation phase into a rank model [58], [59], [67], [75], [80]. Zheng *et al.* [59] compare two learning to rank models: Ranking Perceptron for pairwise and a listwise model ListNet where Ranking Perceptron uses some labeled pairs like classification method to train the ranking model and ListNet takes all candidate entities as their training data. Ceccarelli *et al.* [58] consider all mentions in the document which train the learning model collectively, in which they score each candidate entity, and the more the number of the candidate entities associated with it, the higher the score of it is. Some researches rank the candidate entities using the probabilistic model [67], [75]. Ganea et al. [67] learn a conditional probability model $p(e|m, c)$ from a corpus of entity-linked documents and find the entity which its score is the best.

Besides those methods above, the methods based on deep learning has been proposes recently [76], [77], [78]. In contrast to traditional learning method, deep learning does not rely on features designed manually. It can learn the representation of features from large data automatically and contain thousands of parameters. In addition, it can quickly learn new effective feature representations from training data for new applications [79]. Sun *et al.* [76] propose a novel neural network approach. They obtain the semantic representation of mention, context and entity using this deep learning method. Huang *et al.* [77] propose a deep semantic relatedness model based on deep neural networks to measure entity semantic relatedness which improves the accuracy rate by 19.4% and 24.5 on two publicly available datasets respectively.

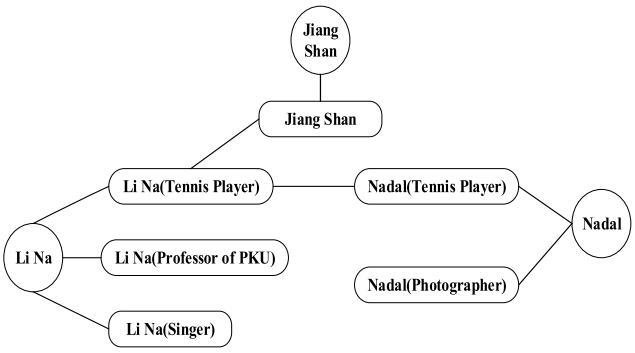

FIGURE 2. A graph for Example 1.

3) METHOD BASED ON GRAPH

A graph based model usually constructs a graph for all mentions and their candidate entities. Different from other methods, it considers the relevance of all mentions and all candidate entities collectively [54], [81]–[85], and a graph for Example 1 is shown in Fig. 2. Han *et al.* [54] propose a Referent Graph approach. The mentions in the document and all candidate entities of each mention constitute the node set of the graph. Each mention node has an edge with its candidate entity. The weight is calculated by the cosine similarity. Meanwhile, among different candidate entity sets, there is an edge between two candidate entities and the weight is calculated by WLM. They reason about the score of a candidate entity by using a random walk model. Gong *et al.* [81] calculate a dense sub-graph which contain all mention nodes and merely one mention-entity edge for each mention. They improve their accuracy by considering the semantic information between mentions in the same document. Blanco *et al.* [83] cast the candidate entity disambiguation problem into an optimization problem on a graph. They define a new graph which is a variant of the Maximum Capacity Representative Set. Liu *et al.* [85] rank the candidate entities by computing the degree of the candidate entity node.

V. NIL CLUSTERING AFTER ENTITY LINKING

The TAC conference presented additional requirements for entity linking which add NIL clustering on the traditional entity linking task in 2011 [86], instead of simply returning NIL to each entity that doesn't exist in the knowledge base which is introduced in Section III. This work involves clustering all entities pointing to NIL that refer to the same entity, and then giving each cluster a unique NIL ID. It is also similar to the task of coreference resolution except that it deals with entities pointing to NIL.

A. METHOD BASED ON STRING MATCHING

Traditional methods commonly solved the NIL clustering problems by matching the surface forms of entities [87], [88]–[92]. These methods utilize different similarity computing measures to calculate the string similarity between entities. Then the entities with high similarity are clustered into a class. Some systems directly judge whether an entity's name is a substring of another entity. Ghosh et al. [87] consider that two entities are highly similar if one of them contains the other. If the condition is satisfied, the new entity will assign to the same NIL set as the previous one, and the algorithm will stop until all entities are traversed. Although this method is simple, it is easy to classify unrelated entities but similar in surface form into one class. In another case, some techniques use string distance computing method to cluster NIL entities. Greenfield *et al.* [90] use Damerau-Levenshtein (DL) distance to compute similarity between entities which can tolerant spelling-error and capture slight local words variations. Torres-Tramón *et al.* [92] apply Monge-Elkan similarity instead. There are also some ways to take the results of string matching methods as the initialization clusters of the method, and then divide them further by other methods. We are going to introduce these approaches next.

B. METHOD BASED ON HIERARCHICAL AGGLOMERATIVE CLUSTERING

Many systems utilize a hierarchical agglomerative clustering (HAC) algorithm for NIL clustering [93]–[95].

This method is a common method to deal with documents clustering problem. The NIL clustering based on a hierarchical agglomerative clustering method usually initializes entities to several different clusters according to the mentions of entities and then merge entities in each clusters until the distance between clusters is smaller than the threshold. Zhang *et al.* [93] first use the dice coefficient to measure the similarity of entities and obtain some clusters. In each cluster there are some entities shares a high dice coefficient. In the second stage, they do their clustering in each cluster obtained previously. First they treat each entity in the set as a cluster and then merge two clusters by computing their similarity. The basic process of Ploch *et al.* [95] is that they merge entities using the cosine similarity, besides they combine three additional methods to cluster NIL entities.

C. METHOD BASED ON GRAPH

Previous methods only take the similarity into consideration when clustering, which ignore the semantic relation between entities. Some systems cluster NIL entities by creating semantic graph of entities. Guo *et al.* [96] take advantage of the results of the entity linking process to build a graph for NIL clustering. After building a graph, they use the hierarchical agglomerative clustering algorithm to clustering the NIL entities. At this stage two similarity measures are used which are attribute similarity and relation similarity. Different from Guo *et al.*, González *et al.* [97] only build a NIL graph for the NIL entities at the NIL clustering stage. In their graph, each node is a cluster and the medoid of each cluster is represented by the first NIL entity. Their goal is to select a correct cluster for each new NIL entity.

At the end of this section, we give Table 2 to make a summary of methods in Section IV and Section V.

TABLE 2. Entity linking method summary.

Phases	Methods	Systems
Candidate entity generation	Method based on dictionary	[54][60][61][62][63][64][65]
	Method based on directly search	[14][66]
	Method based on probability	[67][68]
Candidate entity disambiguation	Method based on similarity computation	[47][55][69][72][73]
	Method based on machine learning	[58][59][60][67][74][75][76][77][78][80]
	Method based on graph	[54][81][82][83][84][85]
NIL clustering	Method based on string matching	[87][88][89][90][91][92]
	Method based on hierarchical agglomerative clustering	[93][94][95]
	Method based on graph	[96][97]

VI. KNOWLEDGE BASE AND EVALUATION

A. KNOWLEDGE BASE

Knowledge Base contains plenty of items of entities. These items are made up of some facts of entities such as the name of entity, category and links between entities. Currently, there is no unified knowledge base in the field of entity linking. The major knowledge base will be illustrated as follows:

Wikipedia: Wikipedia is a semi-structured database. It is one of the most popular encyclopedias in the world. In the Wikipedia knowledge base, there are hundreds of millions of Wikipedia articles. Each article represents an entity or concept which consists of much disambiguation knowledge such as entity pages, each infobox containing entity attributes, hyperlinks and so on. Many systems use Wikipedia as their knowledge base [54], [98]. Han et al. [54] exploit the Jan 30, 2010 English version of Wikipedia as their knowledge base. It includes more than 3,000,000 entities and they built their name-entity dictionary via this knowledge base.

DBpedia: DBpedia is a structured knowledge base. It is formed by the structured data in Wikipedia. The structured data is extracted from tables of inforbox, categorization information, images, geo-coordinates, links to external web pages, disambiguation pages, redirects between pages, and links across different language editions of Wikipedia [7], [99]. DBpedia organize the structured data into the RDF form. It utilizes URI represent an entity and some attributes with their values describe the information of an entity.

YAGO: YAGO is a structured knowledge extracted from Wikipedia, WordNet [100] and GeoNames knowledge base [101]. In YAGO, each Wikipedia article is an entity. The latest version includes 10 million entities and more than 120 million facts about these entities. Additionally, it contains 10 different languages [101].

TAC-KBP: The TAC-KBP knowledge base involves a series of knowledge bases released by the TAC conference. They publish a version at each meeting. The knowledge base is generated automatically from Wikipedia articles. By parsing Wikipedia pages and infoboxes, It obtains more than eight hundred thousand entities [102], [103].

TABLE 3. Entity linking datasets comparison.

Dataset	Mentions	Sources	KB
KORE50	143	tweets	YAGO
AIDA-CoNLL	34587	news	YAGO
NEEL	8665	tweets	DBpedia
OKE2016	1043	Wikipedia	DBpedia
AQUAINT	449	news	Wikipedia
TAC-KBP2011	4338	news	Wikipedia

B. DATASETS

We will introduce some well-known public data sets for entity linking and give a comparison of them in Table 3.

KORE50: The KORE50 is extracted from some micro blogging platforms manually, such as Twitter. It contains many synthetic micro corpora about 50 short texts. Every text is made up of few sentences including some ambiguous mentions. Based on the statistics, each text has 3 mentions and 12.6 words on average. Some detail information about KORE50 can be obtained from literature [104].

AIDA-CoNLL: The AIDA-CoNLL dataset is introduced by Hoffart *et al.* [105]. It is derived from the CoNLL 2003 shared task and contains 34,587 entities in total [106], [107].

NEEL: The Named Entity rEcognition and Linking (NEEL) challenge was established in 2013. The size of its dataset expanded from 2013 to 2016. The 2016 dataset consists of 6,025 tweets which extracted from many noteworthy events from 2011 and 2013 as well as tweets extracted from the Twitter firehose in 2014. There are 8,665 entities in total [51], [106], [108], and the corpus is split into a training set and a testing set.

OKE2016: OKE2016 is a dataset provided by the Open Knowledge Extraction Challenge 2016. It consists of 196 sentences extracted from Wikipedia articles. The average length of the sentences is 155 characters. OKE2016 dataset contains 1,043 mentions [109].

AQUAINT: The AQUAINT dataset contains some news documents which are extracted from the Xinhua News Services, the New York Times and the Associated Press. In the AQUAINT dataset, each document has 14.54 mentions on average [67], [107].

TAC-KBP: The TAC-KBP stands for a series of corpus provided by the TAC conference. This conference provides a benchmark dataset every year [48], [86], [102], [103]. The TAC-KBP2011 dataset is a cross language entity link evaluation corpus. In Chinese part, it uses 1 million pieces of news extracted from the Chinese Gigaword corpus which consists of 1,641 characters (PER), 1,327 organization (ORG) and 1,370 geographical entities.

As described in some works on compare the common benchmark datasets [106], [107], [110], we conclude six different datasets from four aspects in Table 3 which include mentions it contains, sources that it extracted from, the Knowledge Base it corresponds to, and systems that use it for evaluation.

C. EVALUATION CRITERION

In Table 4, it shows the notations we will use in the following evaluation measures. The evaluation measures can be divided into two categories in order to evaluate the performance of the algorithm in the whole dataset. There are two average measures available which are the macro average and the micro average as follows. The macro average computes the relevant measure of each text in a document first, and then calculates their arithmetic average, whereas the micro average consider all mentions in a document together when calculate the relevant measure [111]. At present, some systems use the micro average index to calculate their evaluation measures [60], [112]. However, this method gives more importance to texts with more mentions. Thus, some systems also apply macro average index to their evaluation measure [14], [67], [75], [113].

TABLE 4. Some notations for entity linking evaluation.

D	A document containing a number of texts
M	A set of mentions in a document
n_i	A set of mentions in each text
e_{1i}	A set of manually annotated entities that should be linked correctly in each text
e_{2i}	A set of entities that are manually linked to NIL in each text
g_{1i}	A set of linked entities that generated by a system from each text
g_{2i}	A set of entities generated by a system that are linked to NIL in each text
E_1	A set of manually annotated entities that should be linked correctly in a document
E_2	A set of entities that are manually linked to NIL in a document
G_1	A set of linked entities that generated by a system from a document
G_2	A set of entities generated by a system that are linked to NIL in a document

In the evaluation phase, most systems use the evaluation measure *Accuracy* which computes as Eq. 5 and Eq. 10 [14], [60], [64], [75]. The accuracy refers to the ratio of the number of entities that are correctly linked to the total number of entities in a document. Some systems adopt three measures to evaluate the quality of their approaches. These three measures include *Precision*(P), *Recall*(R), and F_α*-measure* [54], [67], [81], [84], [111], [112]. Usually, they evaluate entities linked to knowledge base (InKB) and entities that are linked to NIL respectively. We use the measures of the micro average as an example. Eq. 6 is the precision of system dealing with entities (InKB), which means the ratio of entities that are correctly linked to the knowledge base to the linked entities generated by a system. Eq. 7 is the recall of system dealing with entities (InKB), which means the ratio of entities that are correctly linked to the knowledge base to the entities that should be correctly linked. Eq. 8 is the precision of a system dealing with entities linked to NIL, which means the ratio of entities that are correctly linked to NIL to the entities linked to NIL generated by a system. Eq. 9 is the recall of system dealing with entities linked to NIL, which means the ratio of entities that are correctly linked to NIL to the entities should be linked to NIL. Some systems put new entities into an existing knowledge base through the method of entity clustering so the evaluation of entities linked to NIL is important, it can promote the knowledge base filling sometimes.

However, *Precision* (P) and *Recall* (R) are two interacting values. In order to improve the *Precision* rate, the *Recall* rate will be reduced, and vice versa. To take into account both of them, F_α*-measure* put them together. Eq. 15 is designed for calculating F_α*-measure*. Usually, the value of α is 1. x stands

TABLE 5. Different measures on micro average.

The Micro Average Measure	
$Accuracy_{micro} = \frac{\lvert E_1 \cap G_1 \rvert + \lvert E_2 \cap G_2 \rvert}{\lvert M \rvert}$	(5)
$P_{InKB\_micro} = \frac{\lvert E_1 \cap G_1 \rvert}{\lvert G_1 \rvert}$	(6)
$R_{InKB\_micro} = \frac{\lvert E_1 \cap G_1 \rvert}{\lvert E_1 \rvert}$	(7)
$P_{NIL\_micro} = \frac{\lvert E_2 \cap G_2 \rvert}{\lvert G_2 \rvert}$	(8)
$R_{NIL\_micro} = \frac{\lvert E_2 \cap G_2 \rvert}{\lvert E_2 \rvert}$	(9)

TABLE 6. Different measures on macro average.

The Macro Average Measure	
$Accuracy_{macro} = \frac{\sum_{i=1}^{\lvert D \rvert} \frac{\lvert e_{1i} \cap g_{1i} \rvert + \lvert e_{2i} \cap g_{2i} \rvert}{\lvert n_i \rvert}}{\lvert M \rvert}$	(10)
$P_{InKB\_macro} = \frac{\sum_{i=1}^{\lvert D \rvert} \frac{\lvert e_{1i} \cap g_{1i} \rvert}{\lvert g_{1i} \rvert}}{\lvert D \rvert}$	(11)
$R_{InKB\_macro} = \frac{\sum_{i=1}^{\lvert D \rvert} \frac{\lvert e_{1i} \cap g_{1i} \rvert}{\lvert e_{1i} \rvert}}{\lvert D \rvert}$	(12)
$P_{NIL\_macro} = \frac{\sum_{i=1}^{\lvert D \rvert} \frac{\lvert e_{2i} \cap g_{2i} \rvert}{\lvert g_{2i} \rvert}}{\lvert D \rvert}$	(13)
$R_{NIL\_macro} = \frac{\sum_{i=1}^{\lvert D \rvert} \frac{\lvert e_{2i} \cap g_{2i} \rvert}{\lvert e_{2i} \rvert}}{\lvert D \rvert}$	(14)

for InKB_micro NIL_micro InKB_macro or NIL_macro.

$$F_\alpha = \frac{(\alpha^2 + 1)^* P_x^* R_x}{\alpha^2 * (P_x + R_x)} \tag{15}$$

VII. CHALLENGES AND CONCLUSION

In this paper, we review and summarize the significance and application of entity linking. Then, we analyze some main methods of entity linking. At last, the knowledge base, datasets, and the evaluation criterion are described. Through the research of the entity linking technology, we put forward the future challenges as follows.

The knowledge bases utilized in the entity linking systems are offline databases or extracted from the online database but lacking in automatic update mechanism. Therefore, many knowledge bases are incomplete because of the slow updating. It requires us to mine more evidences for the entity with little information. Hence, exploring a good method to extract more online information can improve the performance of entity linking [71], [85].

In our daily life, we use social tools like Twitter to express emotions. It brings us more experience but also generates a lot of information. This information is usually expressed in the form of short text. Moreover, these short texts lack enough disambiguation information [5], [114]–[116]. Thus, the disambiguation of mentions on short texts is full of challenges.

Existing methods are applicable to the linking in the same language. Although the multilingual entity linking task has been proposed for years, there are few works studying on the multilingual entity linking. Multilingual entity linking refers to link a mention in one language to an entity in another language [66], [86], [103]. It can promote the integration of knowledge among different languages. However, this task is limited by the problems of translation between languages, the portability of the models and gaining the large-scale training corpus, which lead to the low accuracy of multilingual entity linking. Thus, it is very meaningful to study entity linking among different languages.

To improve the accuracy of entity linking, many systems exploit some complex models to address this problem, such as heavy machine learning models which are introduced in Section IV. As compensation, they will have a higher time complexity. It can be regarded as a new challenge to balance accuracy and computing complexity in the future work.

ACKNOWLEDGMENT

The authors thank the reviewers for their elaborate and incisive suggestions.

REFERENCES

[1] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.
[2] M. Marjani *et al.*, "Big IoT data analytics: Architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.
[3] A. L'Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, "Machine learning with big data: Challenges and approaches," *IEEE Access*, vol. 5, pp. 7776–7797, 2017.
[4] D. Rao, P. McNamee, and M. Dredze, "Entity linking: Finding extracted entities in a knowledge base," in *Multi-Source, Multilingual Information Extraction and Summarization*. Berlin, Germany: Springer, 2013, pp. 93–115.
[5] L. Derczynski *et al.*, "Analysis of named entity recognition and linking for tweets," *Inf. Process. Manage.*, vol. 51, no. 2, pp. 32–49, 2015.
[6] R. Ma, "Wikipedia: The free encyclopedia," *Ref. Rev.*, vol. 16, no. 6, p. 5, 2002. [Online]. Available: https://doi.org/10.1108/rr.2002.16.6.5.273
[7] J. Lehmann *et al.*, "DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.
[8] T. Rebele, F. Suchanek, J. Hoffart, J. Biega, E. Kuzey, and G. Weikum, "YAGO: A multilingual knowledge base from Wikipedia, Wordnet, and Geonames," in *Proc. ISWC*, 2016, pp. 177–185.
[9] M. Cornolti, P. Ferragina, M. Ciaramita, S. Rued, and H. Schutze, "A piggyback system for joint entity mention detection and linking in Web queries," in *Proc. WWW*, 2016, pp. 567–578
[10] R. Blanco, G. Ottaviano, and E. Meij, "Fast and space-efficient entity linking for queries," in *Proc. WSDM*, 2015, pp. 179–188.
[11] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring semantic similarity between words using Web search engines," in *Proc. WWW*, 2007, pp. 757–766
[12] X. L. Dong *et al.*, "From data fusion to knowledge fusion," *Proc. VLDB Endowment*, vol. 7, no. 10, pp. 881–892, 2014.

[13] C. Böhm, M. Freitag, A. Heise, C. Lehmann, A. Mascher, and F. Naumann, "GovWILD: integrating open government data for transparency," in *Proc. WWW*, 2012, pp. 321–324

[14] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin, "Entity disambiguation for knowledge base population," in *Proc. COLING*, 2010, pp. 277–285.

[15] B. Min, M. Freedman, and T. Meltzer, "Probabilistic inference for cold start knowledge base population with prior world knowledge," in *Proc. EACL*, 2017, pp. 601–612.

[16] M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. M. Gómez-Berbís, "Named entity recognition: Fallacies, challenges and opportunities," *Comput. Standards Interfaces*, vol. 35, no. 5, pp. 482–489, 2013.

[17] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticæ Invest.*, vol. 30, no. 1, pp. 3–26, 2007.

[18] G. Luo, X. Huang, C. Y. Lin, and Z. Nie, "Joint named entity recognition and disambiguation," in *Proc. EMNLP*, 2015, pp. 879–880.

[19] A. Sil and A. Yates, "Re-ranking for joint named-entity recognition and linking," in *Proc. CIKM*, 2013, pp. 2369–2374.

[20] L. Chiticariu, R. Krishnamurthy, Y. Li, F. Reiss, and S. Vaithyanathan, "Domain adaptation of rule-based annotators for named-entity recognition tasks," in *Proc. EMNLP*, 2010, pp. 1002–1012.

[21] K. Humphreys *et al.*, "University of Sheffield: Description of the LaSIE-II system as used for MUC-7," in *Proc. MUC*, 1998.

[22] H. Isozaki and H. Kazawa, "Efficient support vector classifiers for named entity recognition," in *Proc. COLING*, 2002, pp. 1–7.

[23] R. Speck and A.-C. N. Ngomo, "Ensemble learning for named entity recognition," in *Proc. ISWC*, 2014, pp. 519–534.

[24] Z. Tang, L. Jiang, L. Yang, and K. Li, "CRFs based parallel biomedical named entity recognition algorithm employing MapReduce framework," *Cluster Comput.*, vol. 18, no. 2, pp. 493–505, 2015.

[25] S. Saha and A. Ekbal, "Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition," *Data Knowl. Eng.*, vol. 85, pp. 15–39, May 2013.

[26] K. Shaalan and M. Oudah, "A hybrid approach to Arabic named entity recognition," *J. Inf. Sci.*, vol. 40, no. 1, pp. 67–87, 2014.

[27] Y. Wu, J. Zhao, and B. Xu, "Chinese named entity recognition combining a statistical model with human knowledge," in *Proc. MultiNER*, 2003, pp. 65–72.

[28] G. Petasis, F. Vichot, F. Wolinski, G. Paliouras, V. Karkaletsis, and C. D. Spyropoulos, "Using machine learning to maintain rule-based named-entity recognition and classification systems," in *Proc. ACL*, 2001, pp. 426–433.

[29] P. Elango, "Coreference resolution: A survey," Dept. Comput. Sci., Univ. Wisconsin, Madison, WI, USA, 2005.

[30] V. Ng, "Machine learning for entity coreference resolution: A retrospective look at two decades of research," in *Proc. AAAI*, 2017, pp. 4877–4884.

[31] J. R. Hobbs, "Resolving pronoun references," in *Proc. Readings Natural Lang. Process.*, 1986, pp. 339–352.

[32] B. J. Grosz, S. Weinstein, and A. K. Joshi, "Centering: A framework for modeling the local coherence of discourse," *J. Comput. Linguistics*, vol. 21, no. 2, pp. 203–225, Jun. 1995.

[33] V. Ng, "Advanced machine learning models for coreference resolution," in *Anaphora Resolution*. Berlin, Germany: Springer, 2016, pp. 283–313.

[34] W. M. Soon, H. T. Ng, and D. C. Y. Lim, "A machine learning approach to coreference resolution of noun phrases," *Comput. Linguistics*, vol. 27, no. 4, pp. 521–544, Dec. 2001.

[35] A. Björkelund and J. Kuhn, "Learning structured perceptrons for coreference resolution with latent antecedents and non-local features," in *Proc. ACL*, 2014, pp. 47–57.

[36] H. Lee, A. Chang, Y. Peirsman, and N. Chambers, "Deterministic coreference resolution based on entity-centric, precision-ranked rules," *Comput. Linguistics*, vol. 39, no. 4, pp. 885–916, 2013.

[37] H. Lee, M. Surdeanu, and D. Jurafsky, "A scaffolding approach to coreference resolution integrating statistical and rule-based models," *Natural Lang. Eng.*, vol. 23, no. 5, pp. 733–762, 2017.

[38] R. Navigli, "Word sense disambiguation: A survey," *ACM Comput. Surveys*, vol. 41, no. 2, p. 10, 2009.

[39] A. Raganato, J. Camacho-Collados, and R. Navigli, "Word sense disambiguation: A unified evaluation framework and empirical comparison," in *Proc. EACL*, 2017, pp. 99–110.

[40] A. X. Chang, V. I. Spitkovsky, C. D. Manning, and E. Agirre, "A comparison of named-entity disambiguation and word sense disambiguation," in *Proc. LREC*, 2016, pp. 860–867.

[41] A. Moro, A. Raganato, and R. Navigli, "Entity linking meets word sense disambiguation: A unified approach," *Trans. Assoc. Comput. Linguistics*, vol. 2, no. 1, pp. 231–244, 2014.

[42] Y. Freund and L. Mason, "The alternating decision tree learning algorithm," in *Proc. ICML*, 1999, pp. 124–133.

[43] C. Cortes and V. Vapnik, "Support vector machine," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[44] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proc. ACL*, 1995, pp. 189–196.

[45] D. S. Chaplot, P. Bhattacharyya, and A. Paranjape, "Unsupervised word sense disambiguation using Markov random field and dependency parser," in *Proc. AAAI*, 2015, pp. 2217–2223.

[46] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[47] S. Cucerzan, "Large-scale named entity disambiguation based on Wikipedia data," in *Proc. IJCAL*, 2007, pp. 708–716.

[48] P. McNamee and H. T. Dang, "Overview of the TAC 2009 knowledge base population track," in *Proc. TAC*, 2009, pp. 111–113.

[49] J. R. Finkel, T. Grenager, and C. Manning, "IIncorporating non-local information into information extraction systems by Gibbs sampling," in *Proc. ACL*, 2005, pp. 363–370.

[50] G. Rizzo, B. Pereira, A. Varga, M. van Erp, and A. E. C. Basave, "Lessons learnt from the Named Entity rEcognition and Linking (NEEL) challenge series," *Semantic Web*, vol. 8, pp. 667–700, Jun. 2017.

[51] S. Guo, M. W. Chang, and E. Kiciman, "To link or not to link? A study on end-to-end tweet entity linking," in *Proc. NAACL HLT*, 2013, pp. 1020–1030.

[52] X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, and Y. Lu, "Entity linking for tweets," in *Proc. ACL*, 2013, pp. 1304–1311.

[53] S. Monahan, J. Lehmann, T. Nyberg, J. Plymale, and A. Jung, "Cross-lingual cross-document coreference with entity linking," in *Proc. TAC*. 2011.

[54] X. Han, L. Sun, and J. Zhao, "Collective entity linking in Web text: A graph-based method," in *Proc. SIGIR*, 2011, pp. 765–774.

[55] R. C. Bunescu and M. Pasca, "Using encyclopedic knowledge for named entity disambiguation," in *Proc. EACL*, 2006, pp. 9–16.

[56] X. Han and L. Sun, "An entity-topic model for entity linking," in *Proc. EMNLP-CoNLL*, 2012, pp. 105–115.

[57] B. X. Huai, T. F. Bao, H. S. Zhu, and Q. Liu, "Topic modeling approach to named entity linking," (in Chinese), *J. Softw.*, vol. 25, no. 9, pp. 2076–2087, 2014.

[58] D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, and S. Trani, "Learning relatedness measures for entity linking," in *Proc. CIKM*, 2013, pp. 139–148.

[59] Z. Zheng, F. Li, M. Huang, and X. Zhu, "Learning to link entities with knowledge base," in *Proc. NAACL HLT*, 2010, pp. 483–491.

[60] W. Zhang, J. Su, C. L. Tan, and W. T. Wang, "Entity linking leveraging: Automatically generated annotation," in *Proc. COLING*, 2010, pp. 1290–1298.

[61] Z. Guo and D. Barbosa, "Robust entity linking via random walks," in *Proc. CIKM*, 2014, pp. 499–508.

[62] W.-H. Chong, E.-P. Lim, and W. Cohen, "Collective entity linking in tweets over space and time," in *Proc. ECIR*, Cham, Switzerland, 2017, pp. 82–94.

[63] J. G. Zheng *et al.*, "Entity linking for biomedical literature," *BMC Med. Inf. Decision Making*, vol. 15, no. 1, p. S4, 2015.

[64] W. Shen, J. Wang, P. Luo, and M. Wang, "Linden: Linking named entities with knowledge base via semantic knowledge," in *Proc. WWW*, 2012, pp. 449–458

[65] W. Zhang, Y. C. Sim, J. Su, and C. L. Tan, "Entity linking with effective acronym expansion, instance selection, and topic modeling," in *Proc. IJCAI*, 2011, pp. 1909–1914.

[66] P. McNamee, J. Mayfield, D. Lawrie, D. W. Oard, and D. Doermann, "Cross-language entity linking," in *Proc. IJCNLP*, 2011, pp. 255–263.

[67] O. E. Ganea, M. Ganea, A. Lucchi, C. Eickhoff, and T. Hofmann, "Probabilistic bag-of-hyperlinks model for entity linking," in *Proc. WWW*, 2016, pp. 927–938

[68] X. Pan, T. Cassidy, U. Hermjakob, H. Ji, and K. Knight, "Unsupervised entity linking with abstract meaning representation," in *Proc. NAACL HLT*, 2015, pp. 1130–1139.

[69] R. Mihalcea and A. Csomai, "Wikify!: Linking documents to encyclopedic knowledge," in *Proc. CIKM*, 2007, pp. 233–242.
[70] V. Varma *et al.*, "IIIT Hyderabad at TAC 2009," in *Proc. TAC*, 2009.
[71] Y. Li, C. Wang, F. Han, J. Han, D. Roth, and X. Yan, "Mining evidences for named entity disambiguation," in *Proc. SIGKDD*, 2013, pp. 1070–1078.
[72] D. Milne and I. H. Witten, "Learning to link with wikipedia," in *Proc. CIKM*, 2008, pp. 509–518.
[73] L. Ratinov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to wikipedia," in *Proc. ACL-HLT*, 2011, pp. 1375–1384.
[74] A. Pilz and G. Paaß, "From names to entities using thematic context distance," in *Proc. CIKM*, 2011, pp. 857–866.
[75] X. Han and L. Sun, "A generative entity-mention model for linking entities with knowledge base," in *Proc. ACL-HLT. Assoc. Comput. Linguistics*, 2011, pp. 945–954.
[76] Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, and X. Wang, "Modeling mention, context and entity with neural networks for entity disambiguation," in *Proc. IJCAI*, 2015, pp. 1333–1339.
[77] H. Huang, L. Heck, and H. Ji. (2015). "Leveraging deep neural networks and knowledge graphs for entity disambiguation." [Online]. Available: https://arxiv.org/abs/1504.07678
[78] M. Francis-Landau, G. Durrett, and D. Klein. (2016). "Capturing semantic similarity for entity linking with convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1604.00734
[79] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[80] C. Xiong, Z. Liu, J. Callan, and E. Hovy, "JointSem: Combining query entity linking and entity based document ranking," in *Proc. CIKM*, 2016, pp. 2391–2394.
[81] J. Gong, C. Feng, Y. Liu, G. Shi, and H. Huang, "Collective entity linking on relational graph model with mentions," in *Proc. NLP-NABD*, 2017, pp. 159–171.
[82] Y. Guo, W. Che, T. Liu, and S. Li, "A graph-based method for entity linking," in *Proc. IJCNLP*, 2011, pp. 1010–1018.
[83] R. Blanco, P. Boldi, and A. Marino, "Using graph distances for named-entity linking," *Sci. Comput. Programm.*, vol. 130, pp. 24–36, Nov. 2016.
[84] B. Hachey, W. Radford, and J. R. Curran, "Graph-based named entity linking with wikipedia," in *Proc. WISE*, 2011, pp. 213–226.
[85] L. Qiao, Z. Yun, L. Yang, L. Yao, and Q. Zhiguang, "Graph-based collective chinese entity linking algorithm," (in Chinese), *J. Comput. Res. Develop.*, vol. 53, no. 2, pp. 270–283, 2016.
[86] H. Ji, R. Grishman, and H. T. Dang, "Overview of the TAC2011knowledge base population track," in *Proc. TAC*, 2011.
[87] S. Ghosh, P. Maitra, and D. Das, "Feature based approach to named entity recognition and linking for tweets," in *Proc. WWWW*, 2016, pp. 74–76.
[88] T. Cassidy *et al.*, "Cuny-uiuc-sri TAC-KBP2011 entity linking system description," in *Proc. TAC*, 2011.
[89] C. Gârbacea, D. Odijk, D. Graus, I. Sijaranamua, and M. D. Rijke, "Combining Multiple signals for semanticizing tweets: University of Amsterdam at #Microposts2015," in *Proc. WWWW*, 2015, pp. 59–60.
[90] K. Greenfield *et al.*, "A reverse approach to named entity extraction and linking in microposts," in *Proc. WWWW*, 2016. pp. 67–69.
[91] Z. Guo and D. Barbosa, "Entity recognition and linking on tweets with random walks," in *Proc. WWWW*, 2015, pp. 57–58.
[92] P. Torres-Tramón, H. Hromic, B. Walsh, B. R. Heravi, and C. Hayes, "Kanopy4Tweets: Entity extraction and linking for Twitter," in *Proc. WWWW*, 2015, pp. 64–66.
[93] T. Zhang, K. Liu, and J. Zhao, "The NLPRIR entity linking system at TAC 2012," in *Proc. TAC*, 2012.
[94] D. Graus, T. Kenter, M. Bron, E. Meij, and M. de Rijke, "Context-based entity linking—University of Amsterdam at TAC 2012," in *Proc. TAC*, 2012.
[95] D. Ploch, L. Hennig, A. Duka, E. W. De Luca, and S. Albayrak, "GerNED: A german corpus for named entity disambiguation," in *Proc. LREC*, 2012, pp. 3886–3893.
[96] Z. Guo, Y. Xu, F. de Sá Mesquita, D. Barbosa, and G. Kondrak, "ualberta at TAC-KBP 2012: English and cross-lingual entity linking," in *Proc. TAC*, 2012.
[97] E. González *et al.*, "The TALP participation at TAC-KBP 2012," in *Proc. TAC*, 2012.
[98] B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J. R. Curran, "Evaluating entity linking with Wikipedia," *Artif. Intell.*, vol. 194, pp. 130–150, Jan. 2013.
[99] M. Morsey, J. Lehmann, S. Auer, C. Stadler, and S. Hellmann, "DBpedia and the live extraction of structured data from Wikipedia," *Program*, vol. 46, no. 2, pp. 157–181, 1966.
[100] R. Richardson, A. Smeaton, and J. Murphy, "Using WordNet as a knowledge base for measuring semantic similarity between words," in *Proc. AICS*, 1994, pp. 1–15.
[101] F. Mahdisoltani, J. Biega, and F. Suchanek, "Yago3: A knowledge base from multilingual wikipedias," in *Proc. CIDR*, 2014.
[102] H. Ji, J. Nothman, and B. Hachey, "Overview of TAC-KBP2014 entity discovery and linking tasks," in *Proc. TAC*, 2014, pp. 1333–1339.
[103] H. Ji, J. Nothman, B. Hachey, and R. Florian, "Overview of TAC-KBP2015 tri-lingual entity discovery and linking," in *Proc. TAC*, 2015.
[104] J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum, "KORE: Keyphrase overlap relatedness for entity disambiguation," in *Proc. CIKM*, 2012, pp. 545–554.
[105] J. Hoffart, M. A. Yosef, I. Bordino, H. Furstenau, M. Pinkal, and M. Spaniol, "Robust disambiguation of named entities in text," in *Proc. EMNLP*, 2011, pp. 782–792.
[106] M. van Erp *et al.*, "Evaluating entity linking: An analysis of current benchmark datasets and a roadmap for doing a better job," in *Proc. LREC*, vol. 5. 2016, p. 2016.
[107] R. Usbeck *et al.*, "GERBIL: General entity annotator benchmarking framework," in *Proc. WWW*, 2015, pp. 1133–1143
[108] G. Rizzo, A. E. Cano, B. Pereira, and A. Varga, "Making sense of microposts (#Microposts2015) named entity recognition & linking challenge," in *Proc. WWWW*, 2015, pp. 1–3.
[109] J. Plu, G. Rizzo, and R. Troncy, "Enhancing entity linking by combining NER models," in *Proc. Semantic Web Eval. Challenge*, 2016, pp. 17–32.
[110] X. Ling, S. Singh, and D. S. Weld, "Design challenges for entity linking," in *Proc. Trans. Assoc. Comput. Linguistics*, 2015, pp. 315–328.
[111] M. Cornolti, P. Ferragina, and M. Ciaramita, "A framework for benchmarking entity-annotation systems," in *Proc. WWW*, 2013, pp. 249–260
[112] T. Gruetze, G. Kasneci, Z. Zuo, and F. Naumann, "CohEEL: Coherent and efficient named entity linking through random walks," *Web Semantics, Sci., Services Agents World Wide Web*, vols. 37–38, pp. 75–89, Mar. 2016.
[113] J. Zhang, Y. Cao, L. Hou, J. Li, and H. T. Zheng, "XLink: An unsupervised bilingual entity linking system," in *Proc. NLP-NABD*, 2017, pp. 172–183.
[114] P. Basile and A. Caputo, "Entity linking for tweets," *Encyclopedia Semantic Comput. Robot. Intell.*, vol. 1, no. 1, p. 1630020, 2017.
[115] P. Basile, A. Caputo, A. L. Gentile, and G. Rizzo, "Overview of the EVALITA 2016 named entity recognition and linking in Italian tweets (NEEL-IT) task," in *Proc. CEURW*, vol. 1749. 2016, pp. 1–8, paper 7.
[116] B. Ma, Y. Yang, X. Zhou, and L. Wang, "Graph-based short text entity linking: A data integration perspective," in *Proc. IALP*, 2016, pp. 193–197.

GONGQING WU received the bachelor's degree from Anhui Normal University, China, the master's degree from the University of Science and Technology of China, and the Ph.D. degree from the Hefei University of Technology, China, all in computer science.

He is currently an Associate Professor of computer science with the Hefei University of Technology. His current research interests include data mining and Web intelligence.

Prof. Wu has authored or co-authored over 30 research papers. He received the Best Paper Award at the 2011 IEEE International Conference on Tools with Artificial Intelligence and the Best Paper Award at the 2012 IEEE/WIC/ACM International Conference on Web Intelligence.

YING HE received the bachelor's degree in computer science and technology from Northwest Normal University, China, in 2015. She is currently pursuing the master's degree with the Hefei University of Technology. Her research interests are in the field of data mining and Web intelligence.

XUEGANG HU received the B.S. degree from the Department of Mathematics, Shandong University, Shandong, China, and the M.S. and Ph.D. degrees in computer science from the Hefei University of Technology (HFUT), Hefei, China.

He is currently a Professor with the School of Computer Science and Information Engineering, HFUT. His current research interests include data mining and knowledge engineering.