IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 28, 2018, accepted March 15, 2018, date of publication March 21, 2018, date of current version April 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2817652

Collective List-Only Entity Linking: A Graph-Based Approach

WEIXIN ZENG[1], XIANG ZHAO[1,2], JIUYANG TANG[1,2] AND HAICHUAN SHANG[3,4]

[1]College of System Engineering, National University of Defense Technology, Changsha 410073, China
[2]Collaborative Innovation Center of Geospatial Technology, Wuhan 430079, China
[3]National Institute of Information and Communications Technology, Tokyo 184-8795, Japan
[4]Institute of Industrial Science, The University of Tokyo, Tokyo 113-8654 Japan

Corresponding author: Xiang Zhao (xiangzhao@nudt.edu.cn)

The work was supported by NSFC under Grant 61402494, Grant 71690233, and Grant 71331008, and in part by the NSF of Hunan Province under Grant 2015JJ4009.

ABSTRACT List-only entity linking (EL) is the task of mapping ambiguous mentions in texts to target entities in a group of entity lists. Different from traditional EL task, which leverages rich semantic relatedness in knowledge bases to improve linking accuracy, the list-only EL can merely take advantage of co-occurrences information in entity lists. State-of-the-art work utilizes co-occurrences information to enrich entity descriptions which are further used to calculate local compatibility between mentions and entities to determine results. Nonetheless, entity coherence is also deemed to play an important part in EL, which is yet currently neglected. In this paper, in addition to local compatibility, we take into account global coherence among entities. Specifically, we propose to harness co-occurrences in entity lists for mining both explicit and implicit entity relations. The relations are then integrated into an entity graph, on which personalized PageRank is incorporated to compute entity coherence. The final results are derived by combining local mention-entity similarity and global entity coherence. The experimental studies validate the superiority of our method. Our proposal not only improves the performance of the list-only EL, but also opens up the bridge between the list-only EL and conventional EL solutions.

INDEX TERMS List-only entity linking, named entity disambiguation, graph-based approach.

I. INTRODUCTION

Entity Linking (EL) is the task of detecting corresponding named *entities* for ambiguous *mentions* in text. Mention refers to character string, such as *Jackson* in the example shown in Fig. 1, the true meaning of which needs to be determined by being linked to an entity, such as the basketball coach *Phil Jackson*. Traditional EL methods leverage *knowledge bases* (KBs), which offer rich semantic information of entities, for robust and accurate disambiguation process. Nevertheless, despite the effectiveness of knowledge-based EL, it might not be applicable in situations where there is insufficient information of entities, such as *entity lists*.

Entity list, as is often the case, consists of a group of closely-related entities, and it exists in various information sources [1]. In contrast to KBs, where complete structure of entities facilitates almost all entity-related tasks, entity list minimizes necessary information to mere co-occurrences of interrelated entities, thus serving as a light-weight alternative in terms of describing entity correlations.

Entity lists can be found useful, for instance, in the scenario concerning detection of emerging stock names. When investors search new stock names in Wikipedia,[1] a frequently updated KB, chances are that there are no corresponding items. In fact, as is shown in [2], for a dataset including 2,468 stock names, merely 340 of them can be found in Wikipedia. Nevertheless, those stocks can be found co-occurring with others in stock lists on financial websites. Thus, the stock lists will be of great use if people have doubts concerning new stocks. There are much more similar situations, such as searching for specific car brands or collecting information about bars in a small town, where the knowledge about target entities is sparse.

Consequently, the demand for list-only EL emerges [1], which targets at solving the problem of mapping ambiguous mentions to entity lists (rather than KBs); Fig. 1 describes an example of list-only EL problem. State-of-the-art method [1]

[1] https://www.wikipedia.org/

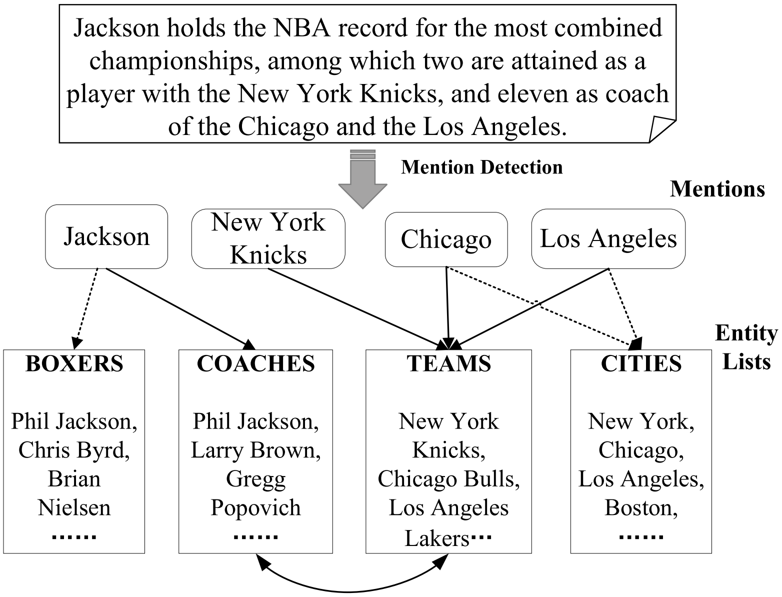

Jackson holds the NBA record for the most combined championships, among which two are attained as a player with the New York Knicks, and eleven as coach of the Chicago and the Los Angeles.
Mention Detection
Mentions
Jackson
New York Knicks
Chicago
Los Angeles
Entity Lists
BOXERS
Phil Jackson, Chris Byrd, Brian Nielsen ……
COACHES
Phil Jackson, Larry Brown, Gregg Popovich ……
TEAMS
New York Knicks, Chicago Bulls, Los Angeles Lakers…
CITIES
New York, Chicago, Los Angeles, Boston, ……

FIGURE 1. Example of list-only entity linking.

addresses the challenge by merely considering the local compatibilities between mentions and entities to determine matching pairs, whereas neglecting the global coherence among entities.

Example 1: As shown in Fig. 1, there is a piece of text with mentions *Jackson*, *New York Nicks*, *Chicago* and *Los Angeles*; and there are 4 sample entity lists to be linked to, namely `Boxers`, `Coaches`, `Teams` and `Cities`. The task of list-only EL is to link mentions to correct entities in the entity lists. It can be seen that entity *Chicago* and entity *Los Angeles* in entity list `Cities` have the same name strings with mention *Chicago* and mention *Los Angeles* in the text. Because of the high mention-entity compatibility, existing method tends to map mentions *Chicago* and *Los Angeles* to entities *Chicago* and *Los Angeles* in the entity list featured `Cities`. However, the true entities for them are *Chicago Bulls* and *Los Angeles Lakers* in entity list `Teams`. Furthermore, it is hard for current method to decide which entity that mention *Jackson* should be linked to, since there are two possible candidate entities with the same name *Phil Jackson* and they are in different entity lists.

Moreover, the dataset used for empirical study might be inappropriate and need a redesign. Current dataset is comprised of documents, which contain mentions to be disambiguated, and a group of entity lists, which include the true entities for mentions. However, each document only contains a single mention for disambiguation, which may not reflect the reality well. A pragmatic scenario may look like the example in Fig. 1, where there are four mentions in one document. Additionally, the entities in different entity lists are dissimilar, making the task much easier to cope with since each mention may well only have one candidate entity. This also deviates from reality and simplifies the problem.

In short, the shortcomings of the existing list-only EL solution are two-fold:

- Entity coherence within or across entity lists was overlooked and not leveraged; and
- Results were supportless for lack of appropriate dataset and deliberate experiment design.

We close the gap and address the deficiencies in this article. In particular, we propose to solve list-only EL task by taking account of the correlations in entities and converting the disambiguation problem to a graph problem. We show the merits of graph-based list-only EL by referring to the example in Fig. 1. It is easy to map mention *New York Knicks* to entity *New York Knicks* in the entity list featured `Teams`. Then by considering the interdependence of entities in the same list `Teams`, mention *Chicago* will be mapped to entity *Chicago Bulls*, and mention *Los Angeles* will be mapped to entity *Los Angeles Lakers*. Additionally, by further taking into account cross-dependence of entities across different entity lists, entity *Phil Jackson* in the `Coaches` entity list, rather than entity *Phil Jackson* in the `Boxers` entity list, will be chosen as the target entity for mention *Jackson*.

To implement graph-based list-only EL, we mainly carry through the following three steps. (1) Pre-processing—including an optional *named entity recognition* process and the candidate entity generation process. This step formalizes raw texts and produces mentions and candidate entities as inputs for later steps. (2) Entity information enrichment. The descriptions of entities are enriched by collecting representative texts from the inputs, which in turn enable the establishment of coherence among entities. (3) Graph-based entity disambiguation. An entity graph is constructed by integrating outputs from earlier steps. We propose a graph-based algorithm Gloel, which implements Personalized PageRank to

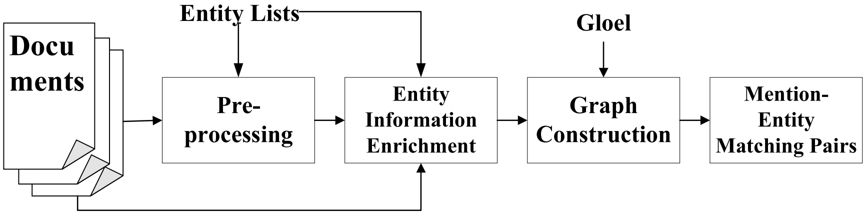


FIGURE 2. **Flowchart of graph-based list-only EL.**

determine how likely an entity is the target entity by taking into consideration both coherences among entities, and compatibilities between mentions and entities. The outputs are a list of pairs comprised of mentions and their most possible entities.

Furthermore, we put forward a new procedure to construct datasets applicable to evaluating list-only EL. The experimental results in this new dataset validate the effectiveness of graph-based linking, a popular method of collective linking, and the in-depth analysis shows that compared with existing list-only linking method, our graph-based solution achieves better performance in list-only EL task.

A. CONTRIBUTIONS

The main contributions of this article can be summarized into three ingredients:

- We motivate to revise list-only EL by taking into account relations between entity lists, i.e., global coherence, in addition to local compatibilities between mentions and entities.
- We tackle the problem by a graph-based method and offer a new algorithm Gloel, where Personalized PageRank is adopted to capture global coherence among candidate entities.
- A new dataset construction procedure is presented to cater to the redefined task, and Gloel is experimentally evaluated on top of it, and shown to outperform state-of-the-art method.

B. ORGANIZATION

This paper is organized as follows. In Section 2, the new definition of list-only EL problem and the methodology, which contains three steps, are elaborated. New dataset construction and experiment results are detailed in Section 3. Section 4 summarizes related work and bridges list-only EL with conventional KB-oriented EL, followed by conclusion in Section 5.

II. METHODOLOGY

We start with defining the proposed problem. Existing work defined list-only EL as mapping a *single* mention m_i in document d_i to the corresponding entity $e_{i,j} \in E_j$ in the entity lists. Nonetheless, on the one hand, in most real-life documents, there are more than one mention, differentiating this definition from reality. On the other hand, the ambiguity between entity lists is not stressed, which can turn the problem of mapping mentions to a group of highly ambiguous entities into determining whether the mentions have corresponding entities in the entity lists. And the latter also deviates from the original motivation of EL task, which centres on disambiguating mentions from several possible meanings. We will further elaborate the definition of ambiguity between entity lists via mathematical equations in Section 3.

As a consequence, it is vital to extend the definition of this task so as to cater to broader scenarios. Specifically, we formalize list-only EL problem as follows.

Definition 1 [List-Only Entity Linking]: Given a set of documents $D = \{d_1, \ldots d_n\}$, each of which contains a set of *mentions* $M_i = \{m_{i1}, \ldots m_{is}\}$, an ambiguous set of entity lists $\mathcal{E} = \{E_1, \ldots E_l\}$, the task is to determine the most possible entity $e_{ij,k} \in E_k$ for each mention m_{ij}, or return *NIL* if there is no corresponding entity.

Note that the set of entity lists has to be ambiguous to follow the motivation of EL task. In other words, for the majority of entities, there ought to be at least one more ambiguous entity in the entity lists.

We take the example in Fig. 1 to explain the definition. There are four mentions to be disambiguated in the document. By utilizing list-only EL, mentions *New York Knicks*, *Chicago*, *Los Angeles* should be mapped to entities *New York Knicks*, *Chicago Bulls*, *Los Angeles Lakers* in the entity list featured `Teams` respectively, instead of *New York*, *Chicago*, *Los Angeles* in the entity list featured `Cities`. And mention *Jackson* should be linked to entity *Phil Jackson* in the `Coaches` entity list, rather than entity *Phil Jackson* in the `Boxers` entity list.

OVERVIEW: The specific procedure for graph-based list-only entity linking includes three steps, namely, preprocessing, entity information enrichment and graph disambiguation. As shown in Fig. 2, the former two steps generate inputs, based on which the entity graph is constructed and Gloel is performed to determine results.

A. PREPARATION FOR GRAPH INPUT

This subsection presents treatment of raw text data and generation of inputs for graph construction.

1) PRE-PROCESSING

In the pre-processing step, mentions in the text are detected and the candidate entities are also generated.

Specifically, the initial input for EL is a set of raw documents, either with specified mentions to be disambiguated or without. Under the circumstance where mentions are not pointed out, Named Entity Recognition (NER) should be harnessed to finish the mention detection task. State-of-the-art NER methods utilize Neutral Networks and Deep Learning techniques to achieve better performances, whereas they have not been widely used yet on account of the freshness and complexity. Instead, Stanford NER Tagger, a NER tool which is less accurate but maturer, embraces higher popularity in tasks involving but not focusing on NER. In our experiment, we have already extracted the mentions during dataset construction process.

After obtaining mentions, the following step is to retrieve possible candidate entities for each mention. Take Fig. 1 for instance, for mention *Chicago*, both entities *Chicago Bulls* and *Chicago* should be generated as candidates. In order to improve recall and generate more candidate entities, most KB-oriented EL methods tend to take advantage of name dictionaries embedded in KBs, or use alias dictionaries built from collecting Wikipedia redirecting and disambiguation pages. However, considering the limited number of target entities and sparse information of entity lists, we design a set of simple but efficient string matching rules for entity generation, as is shown in Table 1. In the examples, the left are mentions while the right are candidate entities.

TABLE 1. String matching rules.

Rules	Examples
Containment	Chicago → Chicago Bulls
Partial Matching	President Trump → Donald Trump LA → Los Angeles
Alternative Names	National Capital → Washington, D.C. Smiley → Miley Cyrus

The generated candidate entities for mention m_{ij} are represented by $Can(m_{ij})$. Noteworthily, we adopt candidate-pruning policy to ensure that a mention will not have two or more candidates from the same list, since entity list is utilized to help candidate entity within it to compete with entities from other lists, and choosing among candidate entities from the same list will render coherence within entity list useless.

2) ENTITY INFORMATION ENRICHMENT

Solely relying on co-occurrences between entities is not enough to establish relations among entities, let alone semantically bridge mentions with candidate entities. Therefore, we enrich information on entity side by selecting representatives derived from input documents.

Given input documents $D = \{d_1, \ldots d_n\}$, the mentions $M_i = \{m_{i1}, \ldots m_{is}\}$ in each d_i, a set of entity lists $\mathcal{E} = \{E_1, \ldots E_l\}$, the enrichment process should collect *a set of highly relevant and representative texts* $T^r = \{t_1^r, \ldots t_h^r\}$ around mentions for E_r, which can be achieved by harnessing co-occurrences of entities in the same entity list.

Specifically, the idea is that, since a document is not only composed of mentions, but also a lot of other irrelevant information, we merely extract the texts around all mentions in all documents as *candidate representatives* τ to avoid noisy information. If a candidate representative $t_p \in \tau$ contains many entity names from the same entity list E_r, chances are that it indeed shares the same category or topic with entity list E_r, and the mention m_p in candidate representative t_p is thus much more likely to refer to the candidate entity from E_r. Consequently, t_p is a representative of E_r and the text in t_p can be used to enrich the textual descriptions of entities in E_r.

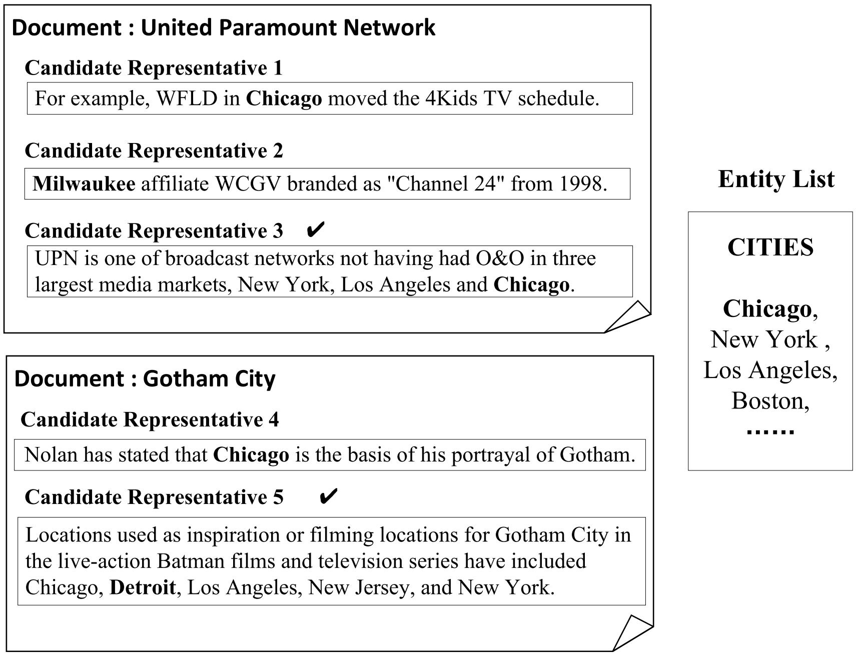
We further illustrate the method in Fig. 3. Note that in each candidate representative, the bold text represents a mention, and the rest texts are its surroundings. Given an entity list `Cities` and the entity *Chicago*, the goal is to collect relevant representatives for *Chicago* from documents, which are then used to enrich representatives of entity list `Cities`. In *Document: United Paramount Network*, there are three candidate representatives, two of them contain name string Chicago. However, Candidate Representative 1 includes no extra name strings of other entities from the entity list, thus might not refer to Entity List `Cities`. In contrary, both *New York* and *Los Angeles* co-occur with *Chicago* in Candidate Representative 3, indicating the high possibility that it is a true representative for Entity List `Cities`. Switching to *Document: Gotham City*, both Candidate Representatives contain name string *Chicago*. Despite the fact that Candidate Representative 4 is derived from mention *Chicago*, we cannot consider it as a representative due to lack of co-occurrences information. Conversely, containing several name strings from entity list `Cities`, Candidate Representative 5 is chosen as a representative, even though it is built surrounding mention *Detroit*.

B. GRAPH CONSTRUCTION AND DISAMBIGUATION

In this subsection, we illustrate the construction of candidate entity graph, followed by the description of our proposed algorithm Gloel, which takes advantage of Personalized PageRank so as to determine target entities.

1) GRAPH CONSTRUCTION

Through the pre-processing step, mentions and their candidate entities are obtained. Then after enriching textual descriptions in the entity side, the *compatibility score* between each mention and corresponding candidate entity can be calculated in terms of text similarity. Previous list-only EL ranked the candidate entities for each mention merely based on mention-entity compatibility scores, thereby producing the results accordingly. We argue that the judgement simply depending on compatibility score is not convincing enough because the coherence among entities is ignored, which plays an indispensable role in the linking process. For instance, as is shown in Fig. 1, it is easy to map mentions *Chicago* and *Los Angeles* to the `Cities` entities *Chicago* and *Los Angeles* due to the short text information and high name string similarity. Provided that the candidate entity coherence is considered, the high interdependence among



Document : United Paramount Network

Candidate Representative 1

For example, WFLD in **Chicago** moved the 4Kids TV schedule.

Candidate Representative 2

Milwaukee affiliate WCGV branded as "Channel 24" from 1998.

Candidate Representative 3 ✔

UPN is one of broadcast networks not having had O&O in three largest media markets, New York, Los Angeles and **Chicago**.

Document : Gotham City

Candidate Representative 4

Nolan has stated that **Chicago** is the basis of his portrayal of Gotham.

Candidate Representative 5 ✔

Locations used as inspiration or filming locations for Gotham City in the live-action Batman films and television series have included Chicago, **Detroit**, Los Angeles, New Jersey, and New York.

Entity List

CITIES

Chicago,
New York ,
Los Angeles,
Boston,
……

FIGURE 3. Example of entity information enrichment.

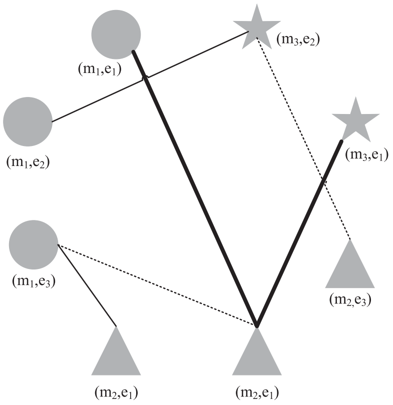


FIGURE 4. Entity graph.

`Teams` entities *New York Knicks, Chicago Bulls, Los Angeles Lakers* would lead to the correct answers for mentions *Chicago* and *Los Angeles*.

To better capture the correlations among entities, similar to many existing KB-based EL methods, we construct an entity graph, which is depicted in Fig. 4. The definition of entity graph is defined as follows:

Definition 2 [Entity Graph]: An *entity graph* $G = \{V,\ E\}$ is a weighed graph, in which the nodes V represent all candidate entities, with their source mentions specified, and edges E include relations between entities.

It is noteworthy that we differentiate the mentions with identical name strings even though they might appear in the same document, and similarly, by specifying the source mention of nodes, the candidate entities with the same name but generated from different mentions are also treated differently. In this way, the situations where there are duplicate nodes, either caused by mentions or entities, can be avoided. In the mathematical form, we represent the r-th candidate entity for mention m_{ij} in document d_i as $e_{ij,r}$, which clearly shows the source mention m_{ij} of candidate entity $e_{ij,r}$.

With reference to edges, following the tradition in KB-based EL and adapting it to list-only problem, we connect two nodes with an edge under three circumstances: (1) The name strings of the two entities are in the same entity list $E \in \mathcal{E}$, and in this case, the edge weight is defined as 1. (2) The name strings of the two entities simultaneously appear in at least one candidate representative $t \in \tau$. (3) The name strings of the other entities in the entity lists these two entities separately belong to, simultaneously appear in at least one candidate representative $t \in \tau$. The first two kinds of relations are termed as *explicit relations*, while the third method of adding edges among entities, named *implicit relations* mining, leverages the unique characteristic of entity list — that the rest entities $E_i^{'} = E_i \backslash \{e_j\}$ in the same entity list E_i can help mine more correlations for entity e_j even if e_j is in the long tail. As for edge weight, which is defined below, takes into account both explicit and implicit relations between entities. Furthermore, the edges among candidate entities with the same source mention are pruned so as to eliminate the influence generated by competitors themselves.

We further assign *initial node weight* $ini(v)$ and *edge weight* on the graph. The *initial node weight* $ini(v)$ is defined as the *compatibility score* between candidate entity and its source mention, while *edge weight* is determined by *relation score* between the entities on the two sides of the edge. The specific approaches to calculate *compatibility score* and *relation score* are:

a: COMPATIBILITY SCORE

Given a document d_i, and m_{ij}, a mention contained in d_i, suppose $e_{ij},_r \in E_r$ is a candidate entity for m_{ij} and $T^r = \{t_1^r, \dots t_h^r\}$ is the set of representative texts for E_r. The compatibility score $\phi(m_{ij}, e_{ij,r})$ can be measured by the following equation

$$ini(m_{ij}, e_{ij,r}) = \phi(m_{ij}, e_{ij,r}) = \frac{1}{|T^r|}\sum_{p=1}^{|T^r|} Sim(m_{ij}, t_p^r). \qquad (1)$$

Since entities in the same entity list share the same representative texts, which are collected according to the method proposed in former section, calculating compatibility between a pair of mention m_{ij} and candidate entity $e_{ij},_r \in E_r$ can be converted to computing the average text similarity *Sim* between texts surrounding mention m_{ij} and all the text representatives T^r of candidate entity $e_{ij},_r$.

There are many ways to measure text similarity *Sim* and in this paper, we choose to compute the similarity between *embedding vectors* of two texts, which is represented as $E(m_{ij}, t_p^r)$. Additionally, we also regard the name string similarity between mention and candidate entity as an appropriate indicator, and it is denoted as $N(m_{ij}, e_{ij},_r)$. Thus, the *Compatibility score* equation is converted to

$$\phi(m_{ij}, e_{ij,r}) = \alpha N(m_{ij}, e_{ij},_r) + \beta\frac{1}{|T^r|}\sum_{p=1}^{|T^r|} E(m_{ij}, t_p^r). \qquad (2)$$

In the equation above, α and β are the weight coefficients balancing the importance of text similarity and name string similarity.

b: RELATION SCORE

Given two entities $e_i^p \in E_p, e_j^q \in E_q$ (We merely consider relationships among entities when calculating Relation Score, which is mention-irrelevant, thus we neglect the mention here), the *Relation Score* is denoted in the following equation

$$Rel(e_i^p, e_j^q) = \begin{cases} \eta O(e_i^p, e_j^q) + \frac{\theta}{M}\sum\limits_{u}^{E_p-i}\sum\limits_{v}^{E_q-j} O(e_u^p, e_v^q), & p \neq q; \\ 1, & p = q, \end{cases} \qquad (3)$$

where

$$O(e_i, e_j) = \frac{|Occur(e_i) \cap Occur(e_j)|}{|Occur(e_i) \cup Occur(e_j)|},$$

$Occur(e) = \{t | e \in t, t \in \tau\}$, and $M = (|E_p| - 1)(|E_q| - 1)$.

We illustrate equations above as follows: $Occur(e)$ denotes the occurrences of entity e in *all* candidate representatives τ, since compared with noisy textual information contained in the whole documents, merely considering texts around mentions (candidate representatives) can improve the accuracy. The Co-occurrence Frequency $O(e_i, e_j)$ of two entities e_i and e_j is defined as the number of candidate representatives they both occur in, divided by all the candidate representatives they either occur in together, or separately. As for the *Relation Score* $Rel(e_i^p, e_j^q)$ of two entities $e_i^p \in E_p, e_j^q \in E_q$, if p equals q, which means e_i^p and e_j^q are from the same entity list, we set the relation score as 1. Otherwise, the score is composed of two parts. The first component is the direct Co-occurrence Frequency of these two entities, multiplied by a weight factor η, which indicates explicit relations. The implicit relations are represented by indirect Co-occurrence Frequency, which is the second component with a coefficient θ, and it takes into account the co-occurrences of the rest entities in E_p and E_q in a pair-wise fashion.

Furthermore, as is shown in Fig. 4, there are three kinds of lines. The bold line represents that entities on the two sides are in the same entity list, and the *Relation Score* is 1. The dotted line denotes that two entities merely have implicit relations, while the normal line requires that there are explicit relations between entities.

It is noteworthy that, different from traditional KB-oriented EL problem which merely considers the direct relations between two entities, we extend the definition by taking into account the contribution made by relations between two entity lists as well, and represent them as implicit relations of two entities. The detailed approach to quantitatively describe the implicit relations is embodied in the equations above.

2) RANKING MENTION-ENTITY PAIRS

Given a weighed entity graph G_i of document d_i, the target is to find the most likely entity $e_{ij,k}$ from a group of entities for each mention m_{ij} in document d_i. In line with popular methods proposed in KB-oriented EL [3], we propose graph-based list-only entity linking algorithm, namely Gloel, which utilizes Personalized PageRank to depict the coherence among candidate entities.

Specifically, we assign a vector $p(v_s)$ with length n to each node v_s to represent the results of a PageRank process starting from v_s. To better capture the coherence among entities within the same document, instead of regarding the similarity between the vectors of nodes as the *coherence score*, we define it as how a candidate entity fits in the document. To enable the definition, a n-length vector $p(d_i)$ is also assigned to document d_i, representing the results of the PageRank process initiating from a group of unambiguous nodes. Consequently, the *coherence score* of a candidate entity $e_{ij,r}$ for mention m_{ij} in document d_i is defined as

$$\psi(e_{ij,r}, d_i) = \frac{p(v_{ij,r})p(d_i)}{|p(v_{ij,r})||p(d_i)|}. \qquad (4)$$

We first elaborate the random walk process initiating from a single node, then extend it to calculating document PageRank vector. The PageRank algorithm, based on random walk theory, is firstly proposed to measure the importance of web pages by counting the number and quality of links to this page. It has been applied to EL problems in recent years, and has achieved great performance [3]–[6]. The basic elements of PageRank include initial vector r^0, transition matrix A, and preference vector s. Note that in our method, $r^0 = s$.

Transition Matrix A is the same in both individual and collective processes, the value at ith row and jth column is defined as

$$A_{ij} = \frac{Rel(e_i, e_j)}{\sum_{e_k \in Edges(e_i)} Rel(e_i, e_k)}. \tag{5}$$

where $Edges(e_i)$ represents the edges connected to entity e_i.

When computing the vector $p(v_t)$ for a single node v_t , $r^0 = s = (0 \dots 0, 1(t\text{th}), 0 \dots)_n$, which means that r^0 and s are identical n-length vectors, the position t of the vector is assigned with 1 and the rest are endowed with 0.

The situation is slightly more complicated as for document PageRank vector $p(d_i)$. Firstly, we regard a candidate entity $e_{ij,r}$ as a unambiguous one if it satisfies one of the following conditions:

1) $e_{ij,r}$ is the only candidate entity of mention m_{ij} and $ini(e_{ij,r})$ is above threshold μ. The unambiguous entities of this kind is endowed with initial weight λ.
2) When there are more than one candidate entities and $e_{ij,r}$ is the candidate entity with the largest initial value, suppose $e'_{ij,r}$ is the candidate entity with the second largest initial value. It suffices that $ini(e_{ij,r}) - ini(e'_{ij,r}) \geq \nu$. The initial weight of this kind is κ.
3) If there are no candidate entities meeting the conditions, all the candidate entities will be added to the unambiguous entities set, with the same weight endowments.

After obtaining unambiguous entities set, the actual weight can be assigned via normalization of initial weight values. Note that in graph, unambiguous entities are presented as equivalent nodes, and by placing the actual weight of unambiguous nodes in the corresponding positions of the n-length vector, we can attain r^0 and s for document accordingly. Furthermore, we adopt an iterative disambiguation approach. In other words, after erasing ambiguity for each mention, the chosen result entity will be regarded as unambiguous and added to the unambiguous entities set, with initial weight of ι. Afterwards, the document PageRank vector will be re-computed by utilizing the new unambiguous entities set.

With initial vector r^0, transition matrix A, and preference vector s defined as above, the Personalized PageRank is presented as following

$$r^{t+1} = (1 - \rho) \times A \times r^t + \rho \times s. \tag{6}$$

In the equation above, t represents the tth iteration, and ρ denotes the probability that the random walk process jumps out of the original iteration and starts from a new vector, which is usually set at 0.15. Normally, the restarting nodes are all nodes in the graph, and the weights in vector s are the same, which equal to $\frac{1}{|V|}$. Nonetheless, in this work, vector s is personalized and set as the same with initial vector, which means that the random walk merely restarts from the initial nodes, eliminating the effect from other nodes. When the iterative calculation reaches to a stage where r^k does not change any more or the variation is within a minimal range, we consider that it converges and $p(v_s)$, $p(d_i)$ are thereby attained. At last, we formalize the list-only EL problem in a mathematical way:

Definition 3 [List-Only Entity Linking in Mathematical Form]: Given a set of documents $D = \{d_1, \dots d_n\}$, each of which contains a set of *mentions* $M_i = \{m_{i1}, \dots m_{is}\}$, an ambiguous set of entity lists $\mathcal{E} = \{E_1, \dots E_l\}$, the task is to determine the most possible entity $e_{ij,k} \in E_k$ for each mention m_{ij}, and

$$e_{ij,k} = \underset{e_{ij,r} \in Can(m_{ij})}{\arg\max} \; (\gamma \phi(m_{ij}, e_{ij,r}) + \delta \psi(e_{ij,r}, d_i)). \tag{7}$$

where γ and δ are two weight coefficients balancing the weight between mention-entity compatibility score and entity coherence score. *NIL* will be returned if there is no corresponding entity.

III. EXPERIMENTS AND RESULTS

Considering the deficiency in current list-only EL dataset, we propose a similar but more comprehensive approach for dataset construction. Then our method is validated via experiments on this dataset and the merits are highlighted through comparison with state-of-the-art method.

A. DATASET

The current list-only EL dataset [1] contains 11065 documents and 7 groups of entity lists, with 139 entities in total. Each document merely includes a single mention to be disambiguated. In addition, the entity lists cover the categories of President, Company, University, State, Character, Brand, Restaurant, and the entities in different entity lists are disparate both in terms of surface forms and true meanings.

There are two shortcomings in current dataset. For one thing, each document merely contains a single mention to be disambiguated, which does not fit in most real-life occasions. For another, the target entity lists are not ambiguous enough, giving rise to the situation that most mentions merely have one candidate entity, and the disambiguation problem is converted to judging whether this sole candidate entity is true or not. Take entity *Apple* in `Company` entity list shown in the dataset of [1], there is no other similar entities in the set of entity lists. As a result, when given a mention *Apple*, the candidate entity for it will only be *Apple* in `Company` entity list, and the problem is transformed into deciding whether the mention can be mapped to entity lists or not.

In order to overcome the deficiencies, we propose to mine target entity lists and collect documents. The entity lists can be constructed both manually and automatically, but the ambiguity must be ensured. Given two entity lists $E_m = \{e_{1,m}, \dots e_{i,m}\}$ and $E_n = \{e_{1,n}, \dots e_{j,n}\}$, for E_m, the ambiguity caused by the existence of E_n is defined as

$$Amb(E_m, E_n) = \frac{1}{|E_m|} \sum_{e_{i,m} \in E_m} \operatorname*{argmax}_{e_{j,n} \in E_n} amb(e_{i,m}, e_{j,n}). \quad (8)$$

Note that $amb(e_{i,m}, e_{j,n})$ represents the ambiguity between two entities in different entity lists. Many approaches can be utilized to measure it, and in this paper, we harness the matching rules defined in the candidate entities retrieval section. If matching rules are satisfied, we endow 1 to $amb(e_{i,m}, e_{j,n})$. Otherwise, the value is determined by name string similarity. Furthermore, the reason why only the highest ambiguity value for $e_{i,m}$ is chosen lies in the fact that we merely need to assure $e_{i,m}$ has one ambiguous competitor to avoid the situation as the example above.

For the whole entity lists set $\mathcal{E} = \{E_1, \ldots E_l\}$, the ambiguity is denoted as

$$A(\mathcal{E}) = \frac{1}{|\mathcal{E}|} \sum_{E_p \in \mathcal{E}} \operatorname*{argmax}_{E_q \in \mathcal{E} \backslash E_p} Amb(E_p, E_q). \quad (9)$$

Again, for each entity list E_p, we only consider the highest ambiguity it has with the rest entity lists in $\mathcal{E}$, since constructing a entity lists set with high ambiguity between each pair of entity lists is nearly impossible.

Referring to [1], we generated raw entity lists by utilizing NeedleSeek,[2] which were then filtered and processed according to the definition of ambiguity. At last, seven entity lists with 70 entities in total were generated. The ambiguity of newly-constructed entity lists set is 0.965, calculated according to the equations given above, while the value [3] for entity lists set in [1] is 0.267. Part of the newly constructed entity lists are presented in Table 2.

TABLE 2. Part of entity lists.

E_i	Entities
1	Atlanta, Chicago, Boston, Houston, New York, Detroit...
2	Atlanta Hawks, Chicago Bulls, Boston Celtics, Houston Rockets...
3	Atlanta Braves Chicago Cubs, Boston Red Sox, Houston Astros...
4	Toyota Camry, Ford Ikon, Tata Indica, Honda Accord...
5	Toyota, Ford, Tata Motors, Honda, Hyundai, Chevrolet...
6	Cambridge, Oxford, St Andrews, Warwick, London...
7	University of Cambridge, University of Oxford...

As for building the documents dataset, we emphasize that there have to be at least two mentions in the same document to enable the construction of candidate entity graph. Otherwise there will be no difference between independent linking method and the proposed collective linking method based on graph.

To be specific, we utilized *wikilinks* in Wikipedia to obtain the documents. For each entity $e_{i,k}$ in entity list E_k, its referent Wikipedia page was determined in the first place. For instance, the Wikipedia page of entity *Atlanta* is *en.wikipedia.org/wiki/Atlanta*. Then we randomly retrieved 1,000 Wikipedia pages directing at $e_{i,k}$ via the *WhatLinksHere* page. As for *Atlanta*, the url of its *WhatLinksHere* page is *en.wikipedia.org/wiki/Special:WhatLinksHere/Atlanta*. After conducting the same operation for all the entities in entity list E_k, the links appearing in at least three entities' 1,000 Wikipedia pages were selected and the web pages texts they refer to were considered as documents. In this way, we can affirm that each document involves at least three mentions. Table 3 describes the specific information of documents and mentions.

TABLE 3. Dataset statistics.

Target E_i	#documents	#mentions
1	156	731
2	535	3,450
3	528	3,054
4	41	108
5	151	742
6	97	472
7	115	564
Total	1,623	9,121

B. RESULTS AND ANALYSES

We compare Gloel and the method utilized in [1] (denoted as *Independent*) on the dataset we create. The results are shown in Table 4 and the settings of parameters are listed as follows: $\alpha = 0.4, \beta = 0.6, \eta = 0.7, \theta = 0.3, \gamma = 0.5, \delta = 0.5, \lambda = 0.5, \kappa = 0.4, \iota = 0.3$.

The measurements we adopt are the same with the metrics in [7], namely *Precision*, *Recall* and *F1*. *Precision* takes into account all entity mentions that are linked by the system and determines the correctness. *Recall* on the other hand, considers all the mentions should be linked, and reflects the fraction of correctly linked mentions. *F1* is a balanced indicator of *Precision* and *Recall*.

We first report the results on original dataset. As is depicted in Table 4, Gloel outperforms independent EL method in all occasions, with a overall F1 gain at 1.1%. Nevertheless, it is evident that both methods achieve high Precision, Recall and F1 scores. This can be justified that most mentions in the documents appear in the same name string form as the entity name strings. For instance, in documents containing mention referring to entity *University of Cambridge*, the name form of the mention is also *University of Cambridge*, thus the high name string similarity basically guarantees the correct matching and rules out the possibility of other candidate entities. Plus, this does not fit in situations of most text sources other than Wikipedia. In news reports concerning *University of Cambridge*, it constantly goes by the name *Cambridge* as in sentence *Cambridge beats Oxford in terms of computer science*. In these cases, the probability of generating result entity *Cambridge* is enhanced significantly and the disambiguation difficulty also rises up.

As a consequence, we corrupted the dataset to observe the corresponding results produced by these two methods.

[2] http://needleseek.msra.cn

[3] Since the author did not offer the full entity lists information, we compute the ambiguity of the segmental entity lists presented in the previous work.

TABLE 4. Experimental results on original dataset.

Method		E_1	E_2	E_3	E_4	E_5	E_6	E_7	All
Independent	P	0.914	0.996	0.998	0.818	0.998	0.667	0.997	0.962
	R	0.990	0.984	0.988	0.998	0.959	0.989	0.585	0.959
	F1	0.950	0.990	0.993	0.900	0.979	0.797	0.734	0.961
Gloel	P	0.997	1.000	0.998	0.931	1.000	0.675	0.997	0.974
	R	0.997	0.998	0.997	1.000	0.981	0.992	0.598	0.971
	F1	**0.997**	**0.999**	**0.998**	**0.964**	**0.990**	**0.803**	**0.748**	**0.972**

To achieve corruption and increase ambiguity, we replaced mention names of the entities in lists 2,3,5,7 to the corresponding ambiguous names in lists 1,4,6. For instance, the mention names *Atlanta Hawks* and *Atlanta Braves* were substituted by *Atlanta*. Considering the fact that after corruption, the name string similarity (in [1] the NER results) might be of no use and possibly lead to negative contributions, which was unfair for *Independent* results, we merely took into account the embedding vectors similarity in terms of mention-entity similarity calculation and altered the corresponding parameter setting. It is noteworthy that, for each corruption degree, we generated five corrupted corpus and reported the average results so as to increase the stability and persuasiveness of the outcomes.

The results of 50% corruption in Table 5 are generated after half of the entities' corresponding mention names in lists 2,3,5,7 get replaced. For fair comparison, the parameters are optimized for separate methods. As can be seen, the gap between the results of Independent and Gloel widens. Gloel achieves better outcomes with overall F1 score at 90.9%, while the overall F1 value of previous method is 76.5%, hence validating the superiority of the proposed method.

We further conducted 25% and 75% corruption on the dataset and Fig. 5 dynamically depicts the F1 scores of these two methods under corruption. With input texts getting more difficult, the results of EL based solely on mention-entity compatibility decline rapidly, while Gloel, a method based on graph, still yields robust results with smaller decreases.

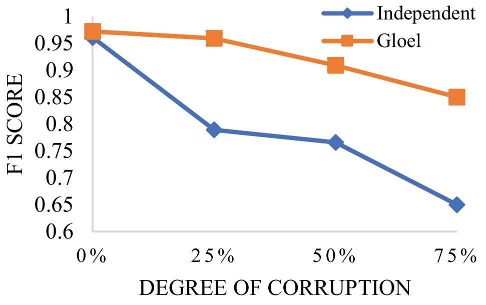


FIGURE 5. F1 score of Independent and Gloel over corrupted dataset.

The following instances are presented to show the improvement made by Gloel and the remaining errors. The improvements mostly take place in documents where half or more than half of the mentions are unambiguous, in which cases the merits of Gloel can be best embodied. A case in point is in Example III-B, whether the unambiguous mentions *California Institute of Technology* and *Stanford University* can help the disambiguation of mentions *Oxford* and *Cambridge*.

Example 2: But Oxford and Cambridge saw significant increases in their total institutional income - up 24% and 11% respectively while their nearest rivals, the California Institute of Technology and Stanford University saw falls in income.

Nevertheless, in scenarios where nearly all mentions in the document are highly ambiguous, Gloel seems to be unable to increase the accuracy, as is shown in Example 3.

Example 3: Boston Celtics won over Toronto by 95-94. Other results: Detroit vs Miami, 112-103. Indiana vs Houston 95-118.

The bad performance of Gloel on this type of text can be justified that, without sufficient texts, even human beings can get confused, let alone an algorithm relying on the text information.

IV. RELATED WORK

In this section, we brief related work, and discuss the differences and connections between list-only EL and traditional EL.

A. LIST-ONLY ENTITY LINKING

Over recent years, in accordance with the emergence of various text sources, EL tasks in new forms have been put forward. List-only EL task, first formally defined by Lin *et al.* [1], is the task of mapping mentions to a group of entity lists, rather than complete KBs. Lin it et al. selected seed mentions for each entity list to bridge the gap between mentions and non-informative target entities, and then conducted the independent linking process to determine final results. Noticing that they merely harnessed entity lists co-occurrences information for generating entity descriptions, in this work, we further utilize the co-occurrences information to model entity relatedness and integrate it in the entity graph, which yields a more robust and accurate EL framework when confronting difficult input texts.

There are other new forms of EL problems which are similar to the list-only task. One is the Target Entity Disambiguation problem [2], [8]. The main disparity is that the focus of Target Entity Disambiguation task lies in finding documents related to the entities given a entity list, whereas the starting point of list-only EL task is to eliminate the ambiguity in documents by using entity lists. Another similar task is the Named Entity Disambiguation with Linkless KBs [9].

TABLE 5. Experimental results on 50% corrupted dataset.

Method		E_1	E_2	E_3	E_4	E_5	E_6	E_7	All
Independent	P	0.968	0.735	0.987	0.167	0.968	0.867	0.639	0.766
	R	0.856	0.994	0.624	0.944	0.287	0.318	0.961	0.764
	F1	0.909	0.845	0.765	0.284	0.443	0.465	0.768	0.765
Gloel	P	0.641	1.000	0.998	0.943	0.967	0.555	0.995	0.910
	R	0.997	0.966	0.899	0.769	0.985	0.992	0.332	0.907
	F1	0.781	**0.982**	**0.946**	**0.847**	**0.976**	**0.712**	0.497	**0.909**

Different from the mere entity lists in our task, there are still textual descriptions for entities in Linkless KBs.

B. KNOWLEDGE BASE ORIENTED ENTITY LINKING

Earlier work on EL focus on the situation where abundant information exists on the entity side. Specifically, KBs such as YAGO, Freebase and Wikipedia, offer rich semantic structures among entities as well as detailed textual descriptions, thus resulting in robust and accurate linking procedure. KB-oriented EL work can generally be divided into independent and collective methods.

In the former approach, mentions are disambiguated merely according the similarity between mentions and entities, and the problem is transformed into candidate entities ranking so as to obtain the most possible result. The similarity is mainly measured by lexical features such as bag-of-words of surrounding texts and statistical features such as prior popularities of entities. Then as for ranking process, unsupervised methods [10] calculate cosine similarities of feature vectors and output the results, whereas supervised approaches [11], [12] construct classifiers by training on annotated dataset, and the linking process is in the charge of classifiers when inputs are given. Although methods of this kind can achieve good results, semantic coherences within entities are neglected, which prove to be essential in improving overall performances.

With respect to collective linking methods in conventional EL task, most of them assume mentions in the same document are semantically coherent, which also should fit in the textual topic of the whole document. Therefore, the resulting entities also are expected to have high relatedness and the problem is in turn converted to find matching pairs maximizing the coherence. Cucerzan [13] proposed to harness Wikipedia categories to model coherence among entities, while Milne and Witten [14] reckoned normalized Google Distance as another useful tool for measurement, which was utilized by Kulkarni *et al.* [15] to form integer linear programming problem so as to collectively obtain results. Hoffart *et al.* [16] defined keyphrase relatedness to capture entity coherence, and proposed to construct a mention-entity graph, on which dense sub-graph generation algorithm was put forward to determine the subgraph containing one-to-one mention-entity matches. The method of re-formalizing the linking problem by constructing mention-entity or entity-only graph distinguished itself among other works due to its capability to integrate both local similarity information between mentions and entities, along with the coherence information among entities. Based on this, several works [3]–[6] proposed and applied modified graph algorithm on the graph, which improved the disambiguation accuracy and the adaptability to difficult texts. Overall, the collective linking methods generally perform better than the independent counterparts in terms of conventional KB-oriented EL.

C. DISCUSSION ON DIFFERENCES AND CONNECTIONS

There are indeed many similarities between these two lines of works, despite of the evident differences. The disparity mainly lies in the information on the entity side. Regarding conventional KB oriented EL, entities have rich and well-structured descriptions offered by KBs, in terms of both text description and internal links among entities [14]–[16]. Thereafter, researchers merely need to filter valuable information to improve linking results. In stark contrast, with respect to list-only scenarios, the mere information existing on the entity side is the co-occurrences among entity name strings in the same entity list, which in turn requires information mining and enrichment. In this paper, to avoid help from structured or semi-structured knowledge source, the dataset itself is leveraged to harvest the relevant relations among entities, thus fulfilling the entity information enrichment task.

Nevertheless, aside from information mining process, the methods utilized in conventional research can be applied to this newly-defined problem and will achieve promising results. For instance, with disambiguation problem taking the form of graph, [3]–[6] can all be implemented.

Above all, the techniques developed in traditional EL also apply in list-only EL problem, and the extra work for the latter is to mine information on the entity side.

V. CONCLUSION

List-only entity linking task, as a new form of traditional EL problem, distinguishes itself by the sparse information on the entity side. In this work, on the one hand, we propose to utilize entity co-occurrences information to mine both textual description of entities and relations among entities, so as to enrich entity information. On the other hand, inspired by conventional EL methods, we construct an entity graph to capture relations among entities, on which the newly proposed algorithm Gloel is applied to obtain results. Similar to the situation in traditional EL, our approach, a collective EL method based on graph, outperforms independent EL on the dataset we create for fair comparison.

For future work, we plan to investigate two aspects. One is to consider the situation where an entity appears in more than one entity list. For instance, *Washington, D.C.* can appear in entity lists featured `American Cities` and `Country Capitals`.

Another possible research direction is utilizing word embedding techniques and deep neural networks to better model mention-entity compatibility and entity coherence. Specifically, leveraging well-trained word embedding vectors as inputs, Long Short-Term Memory (LSTM) [17] with attention mechanism [18] could be used to summarize semantic meanings of the contexts around mentions and the representative texts of entities, which can be further harnessed to calculate more accurate compatibility score.

REFERENCES

[1] Y. Lin, C.-Y. Lin, and H. Ji, "List-only entity linking," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, Vancouver, BC, Canada, 2017, pp. 536–541. [Online]. Available: https://doi.org/10.18653/v1/P17-2085

[2] Y. Cao, J. Li, X. Guo, S. Bai, H. Ji, and J. Tang, "Name list only? Target entity disambiguation in short texts," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Lisbon, Portugal, 2015, pp. 654–664. [Online]. Available: http://aclweb.org/anthology/D/D15/D15-1077.pdf

[3] Z. Guo and D. Barbosa, "Robust entity linking via random walks," in *Proc. 23rd ACM Int. Conf. Inf. Knowl. Manage.*, Shanghai, China, 2014, pp. 499–508. [Online]. Available: http://doi.acm.org/10.1145/2661829.2661887

[4] X. Han, L. Sun, and J. Zhao, "Collective entity linking in Web text: A graph-based method," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Beijing, China, 2011, pp. 765–774. [Online]. Available:http://doi.acm.org/10.1145/2009916.2010019

[5] A. Alhelbawy and R. J. Gaizauskas, "Graph ranking for collective named entity disambiguation," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, Baltimore, MD, USA, 2014, pp. 75–80. [Online]. Available: http://aclweb.org/anthology/P/P14/P14-2013.pdf

[6] M. Pershina, Y. He, and R. Grishman, "Personalized page rank for named entity disambiguation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, Denver, CO, USA, 2015, pp. 238–243. [Online]. Available: http://aclweb.org/anthology/N/N15/N15-1026.pdf

[7] W. Shen, J. Wang, and J. Han, "Entity linking with a knowledge base: issues, techniques, and solutions," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 443–460, Feb. 2015. [Online]. Available: https://doi.org/10.1109/TKDE.2014.2327028

[8] C. Wang, K. Chakrabarti, T. Cheng, and S. Chaudhuri, "Targeted disambiguation of ad-hoc, homogeneous sets of named entities," in *Proc. 21st World Wide Web Conf.*, Lyon, France, 2012, pp. 719–728. [Online]. Available: http://doi.acm.org/10.1145/2187836.2187934

[9] Y. Li, S. Tan, H. Sun, J. Han, D. Roth, and X. Yan, "Entity disambiguation with linkless knowledge bases," in *Proc. 25th Int. Conf. World Wide Web*, Montreal, QC, Canada, 2016, pp. 1261–1270. [Online]. Available: http://doi.acm.org/10.1145/2872427.2883068

[10] R. C. Bunescu and M. Paşca, "Using encyclopedic knowledge for named entity disambiguation," in *Proc. 11th Conf. Eur. Chapter Assoc. Comput. Linguistics*, Trento, Italy, 2006, pp. 1–8. [Online]. Available: http://aclweb.org/anthology/E/E06/E06-1002.pdf

[11] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin, "Entity disambiguation for knowledge base population," in *Proc. 23rd Int. Conf. Comput. Linguistics*, Beijing, China, 2010, pp. 277–285. [Online]. Available: http://aclweb.org/anthology/C10-1032

[12] R. Mihalcea and A. Csomai, "Wikify!: Linking documents to encyclopedic knowledge," in *Proc. 16th ACM Conf. Inf. Knowl. Manage.*, Lisbon, Portugal, 2007, pp. 233–242. [Online]. Available: http://doi.acm.org/10.1145/1321440.1321475

[13] S. Cucerzan, "Large-scale named entity disambiguation based on wikipedia data," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, 2007, pp. 708–716. [Online]. Available: http://www.aclweb.org/anthology/D07-1074

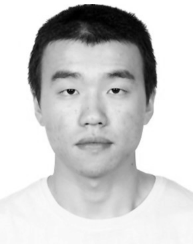
[14] D. N. Milne and I. H. Witten, "Learning to link with wikipedia," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, Napa Valley, CA, USA, 2008, pp. 509–518. [Online]. Available: http://doi.acm.org/10.1145/1458082.1458150

[15] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, "Collective annotation of Wikipedia entities in Web text," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Paris, France, 2009, pp. 457–466. [Online]. Available: http://doi.acm.org/10.1145/1557019.1557073

[16] J. Hoffart *et al.*, "Robust disambiguation of named entities in text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 782–792. [Online]. Available: http://www.aclweb.org/anthology/D11-1072

[17] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, 2015, pp. 1–9. [Online]. Available: http://jmlr.org/proceedings/papers/v37/jozefowicz15.html

[18] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for sentence summarization," in *Proc. 2015 Conf. Empirical Methods Natural Lang. Process.*, Lisbon, Portugal, 2015, pp. 379–389. [Online]. Available: http://aclweb.org/anthology/D/D15/D15-1044.pdf



WEIXIN ZENG received the bachelor's degree from the National University of Defense Technology, China, in 2017, where he is currently pursuing the master's degree. His research interests include knowledge graph and entity linking.

XIANG ZHAO was born in 1986. He received the Ph.D. degree from the University of New South Wales, Australia, in 2014. He is currently an Assistant Professor with the National University of Defense Technology, China. He has published several refereed papers in international conferences and journals, such as SIGMOD, SIGIR, VLDB Journal, and TKDE. His research interests include graph data management and knowledge graph construction.

JIUYANG TANG received the Ph.D. degree from the National University of Defense Technology (NUDT), China, in 2006. He is currently a Professor with NUDT. His research interests include knowledge graph and text analytics.

HAICHUAN SHANG received the B.S. degree from Tsinghua University, China, in 2007, and the Ph.D. degree from the University of New South Wales, Australia, in 2010. He is currently a Researcher with the National Institute of Information and Communications Technology and The University of Tokyo, Japan. He has published several refereed papers in international conferences and journals, such as SIGMOD, VLDB/PVLDB, ICDE, EDBT, and CIKM. His research interests include graph data management, distributed databases, and blockchain technology.