Personalized EntityRank-based Entity Linking with DBpedia

Huiying Li
School of Computer Science and Engineering
Southeast University
Nanjing, P.R.China
huiyingli@seu.edu.cn

Jing Shi
School of Computer Science and Engineering
Southeast University
Nanjing, P.R.China
220151530@seu.edu.cn

ABSTRACT

Entity linking is the task of linking the textual mention in a document to the referent entity in the existing knowledge base. This step is important for many tasks, such as text understanding, semantic search, and question answering over knowledge base. This task is challenging due to name ambiguity and word polysemy. In this paper, we propose an approach to link the entity mention in the document to the DBpedia knowledge base entity. We measure the semantic similarity between a document and a candidate entity by comparing their personalized EntityRank vectors, and then score the candidate entities for each mention combined with other local features. Evaluation on three different datasets shows that our approach outperforms state-of-the-art methods.

Keywords

entity linking, Linked Data, Personalized PageRank, knowledge base, DBpedia

1. INTRODUCTION

Linked Data aims to publish structured data to enable the interlinking of such data and thus enhance their utility[1]. It shares information that can be read automatically by computers and allows data from different sources to be connected and queried. In recent years, many large scale publicly available knowledge bases including DBpedia[2], YAGO[3] and Freebase[4] have emerged.

As the world evolves, digital systems are producing increasing amounts of data daily. Therefore, maintaining and improving the existing knowledge bases become increasingly important. The need for the development of automatic methods for text aggregation, text understanding, and, semantic search is also increasing. Entity linking is a key step towards these goals as it maps the entity mention in the textual document to the corresponding real world entity in the knowledge base. Entity linking task is challenging due to name variations and entity ambiguity. Figure 1 illustrates the difficulty of this task when dealing with real-world data. In reality, an entity may present multiple surface forms. For example, the entity "Michael_Jordan" has its abbreviation "Jordan" and its nickname "Air Jordan". On the contrary, one entity mention may also be shared by multiple entities. For example, the entity mention "Jordan" can refer to the famous basketball player "Michael_Jordan", country "Jordan" in Western Asia.

In this paper, we describe and evaluate a novel approach for document-level entity linking with DBpedia knowledge base. As a prerequisite, we assume that a given input set of mentions is already detected via named entity recognition procedure. First, we collect a dictionary about the surface forms of entities from entity pages, redirect pages and disambiguation pages in Wikipedia. We can then generate candidate entities for the given mention based on this dictionary, and map these candidate entities in Wikipedia to the entities in DBpedia knowledge base. Second, the lexical features (such as Literal Similarity and Context Similarity) of the candidate entity and the popularity of the candidate entity in DBpedia knowledge base are measured. Third, we construct two kinds of entity association graph among the candidate entities on the basis of DBpedia knowledge base. On the basis of the constructed graph, the personalized EntityRank vector for the document and for each candidate entity is computed. Specifically, the anchor entities in the document are applied to compute the personalized EntityRank vector for the document. In addition, we define the global feature for each candidate entity to measure its semantic similarity with the document. Finally, our iterative entity linking algorithm assigns a rank to the candidate entity list for each entity mention with the combination of these four features, namely, Literal Similarity, Context Similarity, Entity Popularity, and Semantic Similarity. The referent entity is added to the preference set when computing the personalized EntityRank vector for the document in the next iteration. We evaluate our approach on three datasets to validate its effectiveness. The experimental results show that our approach outperforms previous methods. The main contributions of this paper are summarized as follows.

- We propose an approach that combines four features,

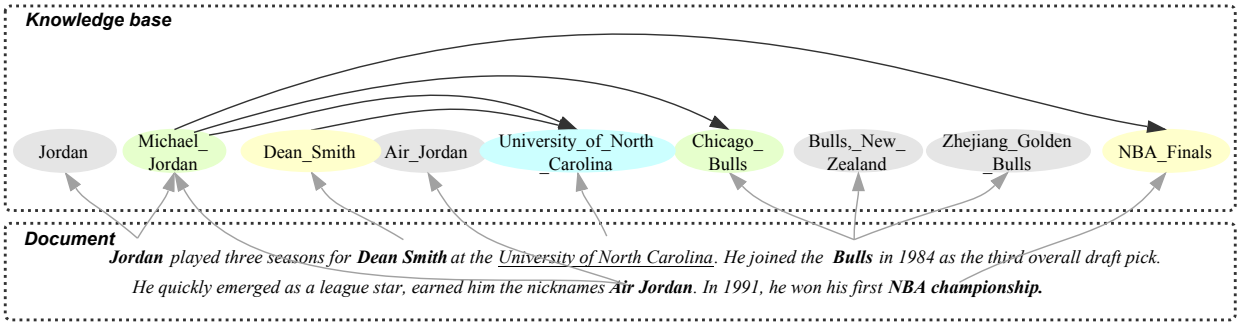

Figure 1: An entity linking problem with five given mentions and their candidate entities.

namely, Literal Similarity, Context Similarity, Entity Popularity, and Semantic Similarity to deal with the entity linking task.

- We propose a method to measure the semantic similarity between the document and the candidate entity by comparing their personalized EntityRank vectors.
- We construct two kinds of entity association graph for computing the personalized EntityRank vector. We specially consider the anchor entities when computing the personalized EntityRank vector for the document.
- We evaluate our approach on three datasets. The experimental results show that our approach can achieve higher accuracy and $F1$ value than state-of-the-art methods do.

The remainder of this article is organized as follows: Section 2 discusses the relevant entity linking literature. Section 3 formally introduces the preliminaries of entity linking task. Section 4 presents the local and global feature applied in our approach. Section 5 empirically demonstrates the merits of the proposed approach on multiple standard collections of manually annotated documents. Finally, Section 6 summarizes our approach and elaborates the conclusions of the study.

2. RELATED WORK

Many existing work focus on the task of entity linking with Wikipedia (Wikification) and can be divided into three main groups, which are discussed below.

Local methods consider the individual context of each entity separately in order to reduce the size of the decision space. Early entity linking systems focus mainly on lexical features, such as the literal of mention or the contextual words surrounding each mention in the document [5]. Bunescu et al. [6] also consider Wikipedia as an information source for named entity disambiguation. The disambiguation is performed using an SVM kernel that compares the lexical context around the ambiguous named entity to the context of the candidate Wikipedia page. Wikify [7] proposes an entity linking method based on similarity statistics between the mention context and the entity's Wikipedia page. Cucerzan [8] addresses the entity linking problem through maximizing the agreement between the text of the mention document and the context of the Wikipedia entity, as well as the agreement among the categories associated with the candidate entities. A supervised entity linking approach is adopted by the authors in [9]. They use a machine learning ranker to score each candidate, and select the candidate with the highest similar score.

The above-mentioned approaches can work when the context is sufficiently rich to uniquely identify a mention, which is not always the case. The subsequent work on Wikification recognize the global document-level topical coherence of the entities.

Global methods attempt to jointly disambiguate all mentions in a document with the assumption that the underlying entities are correlated and consistent with the main topic of the document. These approaches result in superior accuracy, but enlarge the space of possible entity assignments. As a consequence, many approaches in this group of methods rely on approximate inference mechanisms or heuristics. Milne&Witten [10] define the semantic context as a set of unambiguous surface forms in the text, and used the Normalized Google Distance (NGD) to compute the relatedness. Chakrabarti et al. [11] introduce an annotator based on 12 local features and a global feature to the entire document (involving all the other mentions and a relatedness function inspired by [12]). TagMe [13] focuses on short documents such as tweets or search engine snippets. On the basis of evidence across all mentions, they employ a voting scheme for entity disambiguation. The authors in[14] formalize the disambiguation to Wikipedia task as an optimization problem with local and global variants, and analyzes the strengths and weaknesses. Luo et al. [15] jointly model named entity recognition and linking tasks, and captures the mutual dependency between them. It helps to leverage mutual dependency of the two tasks, and to predict coherent outputs. A probabilistic approach is proposed in [16], in which entity mentions are disambiguated jointly across an entire document by combining a document-level prior of entity co-occurrences with local information captured from mentions

and their surrounding context.

Some global methods use the graph method to obtain global coherence (e.g., based on semantic relatedness).

Graph-based methods establish relationships between candidate entities and mentions by use of structural models. Hughes et al. [17] introduce a measure of lexical relatedness based on the divergence of the stationary distributions computed from random walks over graphs extracted from WordNet. Rel-RW [18] adopts a semantic signature for the candidate entity and the document, the semantic signature represents the probability distribution obtained from a random walk on a subgraph of the knowledge base. The iterative disambiguation algorithm selects the candidate entity with the highest total score above a threshold every round. Han et al. [19] propose a referent graph to model the global interdependence among different entity linking decisions. An algorithm is proposed to infer the referent entities of all name mentions by utilizing the interdependence captured in the referent graph. Babelfy [20] creates a graph-based semantic interpretation of an entire document by linking the candidate meanings of the extracted fragments based on the semantic signatures. Hulpuş et al. [21] discuss different measures of semantic relatedness based on the path in the knowledge graph.

Several Linked Data knowledge bases such as DBpedia, Freebase, and YAGO are publicly available, thus, researchers have shown a great interest in mapping the textual entity mention to its corresponding entity in these knowledge bases.

DBpedia Spotlight[22] is a system for automatically annotating text documents with DBpedia URIs. The prominence, topical relevance, and contextual ambiguity score can be configured by users according to their task-specific requirements. LINDEN[23] is a framework to link named entities in text with YAGO knowledge base, by leveraging the rich semantic knowledge embedded in the Wikipedia and the taxonomy of the knowledge base. The semantic associativity and semantic similarity are considered based on the construction of the semantic network. AGDISTIS [24] achieves entity disambiguation by use of only DBpedia knowledge base. After finding the candidate sets for all the ambiguous named entities, AGDISTIS extracts a subgraph of DBpedia that contains all the candidate senses of all the targeted entities and their n-hop neighbors and the relationships between them. Then, the HITS algorithm is run over the extracted subgraph and for each targeted entity, DBpedia concept that has the highest authority score is selected. Cheng et al. [25] propose a method to select features to generate summary for candidate entities, it helps human users carry out entity linking tasks.

Contrary to previous works on this problem, our approach measures the semantic similarity between the document and the candidate entity by comparing their personalized EntityRank vectors. We introduce two kinds of entity association graph for computing the personalized EntityRank vector. Specifically, we consider the anchor entities when computing the personalized EntityRank vector for the document. We analyze the performance of DBpedia Spotlight, AGDISTIS, Rel-RW and our approach in Section 5.

3. PRELIMINARIES

The entity linking task is formalized as follows. Given document d with a set of mentions $M = \{m_1, m_2, ..., m_N\}$, and a knowledge base with an entity set E, entity linking is conducted to produce a mapping from M to $E \cup \{\text{NIL}\}$ [14]. Usually, NIL is used for mentions outside of the knowledge base. We denote the output matching as an N-tuple $\Gamma = (e_1, e_2, ..., e_N)$ where e_i is the output disambiguation for mention m_i.

A local entity linking approach disambiguates each mention separately. Specifically, we let $\phi(m_i, e_j)$ be a score function reflecting the likelihood that the candidate entity $e_j \in E$ is the correct disambiguation for m_i. A local approach solves the following optimization problem:

$$\Gamma^*_{local} = \arg\max_{\Gamma} \sum_{i=1}^{N} \phi(m_i, e_i) \tag{1}$$

We expect that the correct disambiguations will form a "coherent" set of related entities. Global approach defines a coherent function ψ, which solves the following disambiguation problem:

$$\Gamma^*_{local+global} = \arg\max_{\Gamma}(\sum_{i=1}^{N} \phi(m_i, e_i) + \psi(\Gamma)) \tag{2}$$

The above-mentioned global optimization problem above is NP-hard and thus requires approximations.

In the following sections, we introduce the local features and global feature proposed in our entity linking approach.

4. GLOBAL ENTITY LINKING APPROACH

In this section, we introduce our personalized EntityRank-based entity linking approach with DBpedia. First, we introduce the local features used in our approach. Then, we describe two kinds of entity association graph, and present the basic idea of personalized EntityRank. Finally, we introduce the method for combining the semantic similarity with other local features to rank the candidate entities for each mention.

4.1 Local Features

We use three local features in our approach.

Literal Similarity is computed to evaluate the literal similarity between the mention m and the entity candidate e. Given that $surf(m)$ denotes the the surface of mention m, $e.L$ denotes the literal name of entity e. The function to quantify how similar literal mention m is to the entity candidate e can be written as a sigmoid function of t

$$LS(m, e) = \frac{1}{1 + e^{-t}} \tag{3}$$

where $t = ed(surf(m), e.L)$ denotes the edit distance between $surf(m)$ and $e.L$.

Context Similarity is computed to evaluate the context similarity between the mention m and the entity candidate e. The set $m.T$ contains all words occurring in a limit length window centered on m. The window size is set to 50 in the

experiment. The set $e.T$ contains all words occurring in the DBpedia description property of entity e with window length of 100. The context similarity is computed using a Jaccard distance between $m.T$ and $e.T$

$$CS(m,e) = \frac{|m.T \cap e.T|}{|m.T \cup e.T|} \tag{4}$$

Entity Popularity is computed to evaluate the popularity for each candidate entity. Given an entity mention m, $CE(m)$ is the set of candidate entities. Each $e \in CE(m)$ containing the same surface form m shows different popularity, some entities are rare whereas some entities are popular for the given surface form. An entity with many entities pointing to it is a popular entity. Accordingly, we define the popularity score $PS(e)$ for entity e as

$$PS(e) = \frac{num(e)}{\sum_{ce \in CE(m)} num(ce)} \tag{5}$$

where $num(e)$ is the number of DBpedia triples that contains entity e as an object.

4.2 Entity Association Graph

Entity association graph $G_e = (V_e, E, \rho)$ is a basic component for our global feature. To compute the personalized EntityRank vector in the next subsection, we need to use the transition matrix which depends on the edge weights of the entity association graph. We use two kinds of entity association graph to compute the EntityRank vector.

We construct the entity association graph on the basis of the knowledge base. A knowledge graph is defined as follows:

Definition 1. (Knowledge Graph) An RDF triple is a $(subject, predicate, object) \in (U \cup B) \times U \times (U \cup B \cup L)$, where U, B, and L represent the set of URI references (denotes entity), blank nodes, and literals, respectively. A knowledge graph is a set of RDF triples, and is denoted by T.

Let $G_{ce} = (V_e, E, \rho_c)$ denote the Co-occurrence Entity Association Graph, where $V_e = U$ is the set of all entities, $E = \{< e_i, e_j > | (e_i, p, e_j) \in T, e_i, e_j \in V_e\}$, and $\rho_c(< e_i, e_j >) = |\{p|(e_i, p, e_j) \in T, e_i, e_j \in V_e\}|$ denotes the co-occurrence frequency of two entities.

We consider another method to measure the association between two entities. Determining the length of the path between two nodes is a common way of measure the association between nodes in a graph. A path in the knowledge graph is defined as follows:

Definition 2. (Path) A path L from e_i to e_j in the knowledge graph means there is a sequence of triples (e_i, p_i, e_{i+1}), $(e_{i+1}, p_{i+1}, e_{i+2})$,..., (e_{j-1}, p_{j-1}, e_j) in T, denoted as $e_i \xrightarrow{L} e_j$.

We adapt Katz's [27] centrality measure that is commonly used in social network analysis. The idea is that the effectiveness of a path between two nodes is governed by a constant probability α. In case of a path made up of n nodes, the probability of the path is n^α. We use this idea to measure the association between two entities, where the association between two entities is the accumulated score over the top-k shortest paths between them.

Let $G_{ke} = (V_e, E, \rho_k)$ denote the Katz Relatedness Entity Association Graph, where $V_e = U$, $E = \{< e_i, e_j > | \exists L(e_i \xrightarrow{L} e_j), e_i, e_j \in V_e\}$, and $\rho_k(< e_i, e_j >) = rel_{Katz}(e_i, e_j)$ denotes the relatedness between two entities, which can be written as

$$rel_{Katz}(e_i, e_j) = \frac{\sum_{L \in SL(e_i, e_j)} \alpha^{length(L)}}{k} \tag{6}$$

where $SL(e_i, e_j)$ denotes the set of the top-k shortest paths between e_i and e_j.

4.3 Personalized PageRank

After the construction of entity association graph, we can then compute the personalized EntityRank vectors for the candidate entity and for the document.

Let $G_w = (V_w, E)$ denote the Web Page Graph, where V_w is the set of all web pages and E contains a directed edge $< w_i, w_j >$ iff page w_i links to page w_j. For a page w_i, $I(w_i)$ and $O(w_i)$ denote its set of in-neighbors and out-neighbors.

The personalized PageRank vector can be formalized by matrix-vector equations. Let A be the matrix corresponding to the web graph G_w, where $A_{ij} = \frac{1}{O(w_j)}$ if page w_j links to page w_i, and $A_{ij} = 0$ otherwise. For a given *preference vector* u, the personalized PageRank equation can be written as

$$v = (1-c)Av + cu \tag{7}$$

where $c \in (0,1)$ is a constant that is usually set at 0.15, and experiments have shown that small changes have little effect in practice [26]. A solution v to Equation (7) is a steady-state distribution of random surfers, where at each step a surfer jumps to a random preferred page with probability c, and with probability $1-c$ continues forth along a hyperlink [26].

Let $G_e = (V_e, E, \rho)$ denote the Entity Association Graph, where V_e is the set of all entities, E contains a directed edge $< e_i, e_j >$ iff entity e_i is associated to entity e_j, and $\rho : E \to R$ denote the semantic association from entity e_i to e_j.

The personalized EntityRank vector can also be formalized using similar matrix-vector equations. Matrix B denotes the transition matrix of entity association graph G_e, where $B_{ij} = \frac{\rho(<e_j, e_i>)}{\sum_{e_k \in O(e_j)} \rho(<e_j, e_k>)}$. Given a preference set P, the *preference vector* u can be set. The ith component of u is $\frac{1}{|P|}$ if $e_i \in P$, and the ith component of u is 0 if $e_i \notin P$.

For the given *preference vector* u, the EntityRank equation can be written as

$$v = (1-c)Bv + cu \tag{8}$$

In the situation of entity linking problem, when m is an

entity mention in document d, set $CE(m)$ contains the candidate entities for m.

For every entity $e \in CE(m)$, a personalized EntityRank vector v_e can be computed by equation (8) with a preference vector u_e. We can set the ith component of u_e is 1 if $e_i = e$, and the ith component of u_e is 0 if $e_i = e$. In other words, we suppose that the preference set P_e for entity e contains only e itself. Informally, the computed personalized EntityRank vector v_e for entity e is its view of the importance of all the other entities in the entity association graph.

Similarly, a personalized EntityRank vector v_d for document d can also be computed with a preference vector u_d. Ideally, we can set the preference vector u_d if preference set P_d that represents document d is given. The computed vector v_d for document d is its view of the importance of all the other entities.

Among the candidate entities in $CE(m)$, we can measure the global similarity between the candidate entity and the document by comparing their personalized EntityRank vector. An accurate personalized EntityRank vector for the document will improve the entity linking greatly. However, the true preference entity set P_d that represents the document is unknown.

To solve the above-mentioned problem, we use the unambiguous entity and anchor entity to set the preference set for the document. For document d, we can set the preference set $P_d = U_d \cup A_d$ to compute the personalized EntityRank vector v_d from d's view. Set U_d contains the referent entities of all unambiguous mentions in d. If all mentions in the document are ambiguous, then $U_d = \emptyset$. Set A_d contains all the anchor entities in d. In the document, we find that some phrases or words are not annotated as mentions, but can be linked to entities unambiguously, the referent entity is regarded as an anchor entity if its popularity is larger than a threshold. As shown in Figure 1, the phrase "*University of North Carolina*" is not annotated as a mention, but can be linked to a popular entity "University_of_North_Carolina" unambiguously, the entity "University_of_North_Carolina" is an anchor entity (blue node in Figure 1) in this document. To find the anchor entity, we use the Stanford NER (Named Entity Recognizer) tool[1] to detect entity mentions except the annotated mention. These anchor entities can help compute an accurate personalized EntityRank vector for the document.

4.4 Global Entity Linking

We propose an entity linking approach that combines local and global features to perform entity linking iteratively.

First, the personalized EntityRank vectors v_e and v_d are computed for the candidate entity and the document. The KL divergence [17] is used to compute the semantic similarity between candidate entity e and document d.

$$SS(e,d) = \frac{1}{KL(v_e||v_d)} \tag{9}$$

[1]https://nlp.stanford.edu/software/CRF-NER.shtml

$$KL(v_e||v_d) = \sum_i v_{e_i} log \frac{v_{e_i}}{v_{d_i}} \tag{10}$$

Second, the local feature scores $LS(m,e)$, $CS(m,e)$, and $PS(e)$ are also computed for entity e. Based on the four score factors introduced above, we compute the final score for each candidate entity $e \in \bigcup_{m \in M} CE(m)$ in document d. We use a linear function which is expressed as follows:

$$S_{m,d}(e) = W \bullet X_{m,d}(e) \tag{11}$$

The feature vector $X_{m,d}(e)$ is generated for each e, where $X_{m,d}(e) =< LS(m,e), CS(m,e), PS(e), SS(e,d) >$. These features affect the final score, and different features exhibit different degrees of importance. The weight vector W, namely, $W =< w_1, w_2, w_3, w_4 >$, assigns different weights for each feature in $X_{m,d}(e)$.

Third, the candidate entity e with the highest score is selected as the referent entity for its mention. Then, the preference set for document d is updated with $U_d \cup A_d \cup \{e\}$. The personalized EntityRank vector v_d is computed again with this new preference set. This new v_d is used to compute the semantic similarity for other candidate entities on the subsequent round.

The process continues until all mentions in d are linked to the referent entity.

We perform the entity linking iteratively, in which the least ambiguous mention is performed first. In each iteration, the results of previous iterations are used to update the preference set and the personalized EntityRank vector v_d for the document, it represents the document better than the preference set in previous iteration.

5. EXPERIMENTAL STUDY

In this section, we introduce the datasets, evaluation metrics, and evaluation results. The experiments are performed within the Eclipse environment and on a 64 bit quad Core ThinkStation with 3.20 MHz and 16 GB RAM (14 GB of RAM are assigned to the Java virtual machine).

5.1 Datasets

The real-world RDF dataset DBpedia is selected as the knowledge base to evaluate our entity linking approach. DBpedia is extracted based on the hand-generated mappings of Wikipedia infoboxes. All reported experiments are run on DBpedia 2016-04 version[2]. Table 1 shows the statistics of the DBpedia knowledge base.

Table 1: Statistics of the knowledge base.

KB	#entities	#statements	#relationships
DBpedia	12,301,672	42,325,764	11,478,362

We evaluate our approach on three public entity linking datasets(i.e., MSNBC [8], AQUAINT [10], and ACE2004

[2]http://downloads.dbpedia.org/2016-04/

Table 2: Statistics on tested datasets.

Dataset	# non-NIL mentions	# NIL mentions	# documents	mention/document
MSNBC-DBpedia	657	82	20	36.95
AQUAINT-DBpedia	727	0	50	14.54
ACE2004-DBpedia	260	46	36	8.5

Table 3: Effectiveness of local features.

features	Datasets								
	MSNBC-DBpedia			AQUAINT-DBpedia			ACE2004-DBpedia		
	Accuracy	*F1@MI*	*F1@MA*	*Accuracy*	*F1@MI*	*F1@MA*	*Accuracy*	*F1@MI*	*F1@MA*
LS	0.54	0.52	0.55	0.63	0.63	0.64	0.76	0.74	0.75
CS	0.60	0.58	0.58	0.53	0.54	0.55	0.59	0.55	0.56
PS	0.70	0.71	0.70	0.60	0.61	0.62	0.81	0.80	0.80

[14]) built from various sources. These datasets are developed for Wikipedia knowledge base; thus, we annotate new versions[3] of these datasets for DBpedia knowledge base. The statistics of these datasets are shown in Table 2, and their descriptions are provided below.

- MSNBC-DBpedia: MSNBC is a dataset of news documents with 36.95 mentions per document; this dataset contains many mentions that do not easily map to knowledge base entities because of their rare surface forms or distinctive lexicalization. The version for DBpedia knowledge base is called MSNBC-DBpedia, which assigns an unique DBpedia entity to each mention in the dataset.
- AQUAINT-DBpedia: AQUAINT is a dataset of news documents with 14.54 mentions per document. This dataset is compiled from a news corpus from the Xinhua News Service, the New York Times, and the Associated Press. The version for DBpedia knowledge base is called AQUAINT-DBpedia.
- ACE2004-DBpedia: ACE2004 is a subset of ACE2004 Coreference documents with 8.5 mentions per document; this dataset is annotated using Amazon Mechanical Turk. The version for DBpedia knowledge base is called ACE2004-DBpedia.

5.2 Evaluation Metrics

We quantify the quality of an entity linking system by measuring common metrics such as accuracy, precision, recall and F_1 scores. Let M^*_{en} be the ground truth of mentions linking to entities, M^*_{NIL} be the ground truth of mentions linking to NIL. Let M_{en} and M_{NIL} be the output of an entity disambiguation system. Then, our quality metrics are computed as follows:

- Accuracy: $Acc. = \frac{|M^*_{en} \cap M_{en}| + |M^*_{NIL} \cap M_{NIL}|}{|M^*_{en} \cup M^*_{NIL}|}$
- Precision: $P = \frac{|M^*_{en} \cap M_{en}|}{|M_{en}|}$
- Recall: $R = \frac{|M^*_{en} \cap M_{en}|}{|M^*_{en}|}$
- F_1 score: $F_1 = \frac{2 \cdot P \cdot R}{P+R}$

We mostly report results in terms of F_1 scores, namely macro-averaged F1@MA (aggregated across documents), and micro-averaged F1@MI (aggregated across mentions).

[3]The entity linking datasets for DBpedia knowledge base are available at http://cse.seu.edu.cn/PersonalPage/huiyingli/data/dataset.zip

5.3 Evaluation Results

To empirically assess the accuracy gain introduced by each incremental step of our approach, we first conduct experiments for evaluating the proposed local features, namely, *LS*-Literal Similarity; *CS*-Context Similarity; *PS*-Entity Popularity.

Table 3 lists the results of these local approaches on three datasets in terms of accuracy and F1. It is shown that *PS* feature performs better than the two other features when applying to MSNBC-DBpedia and ACE2004-DBpedia datasets. *LS* feature performs well when applying to AQUAINT-DBpedia dataset.

To evaluate the accuracy gain introduced by the global feature, we implement the global feature method with four settings:

***G*$_{coocur}$**: implements the global feature method based on the co-occurrence entity association graph.

***GA*$_{coocur}$**: implements the global feature method based on the co-occurrence entity association graph, and the anchor entities are applied to the preference set when computing the personalized EntityRank vector.

***G*$_{katz}$**: implements the global feature method based on the Katz relatedness entity association graph, with $\alpha = 0.25$ and top-5 most shortest paths.

***GA*$_{katz}$**: implements the global feature method based on the Katz relatedness entity association graph, with the same set above, and the anchor entities are applied to the preference set.

Table 4 lists the results of the global feature method. It is easily noticeable that GA_{coocur} and GA_{katz} methods perform generally better than those of other methods. In other words, global feature alone based on the entity association

Table 4: Effectiveness of global feature.

features	Datasets								
	MSNBC-DBpedia			AQUAINT-DBpedia			ACE2004-DBpedia		
	Accuracy	*F1@MI*	*F1@MA*	*Accuracy*	*F1@MI*	*F1@MA*	*Accuracy*	*F1@MI*	*F1@MA*
G_{coocur}	0.72	0.69	0.68	0.51	0.51	0.52	0.63	0.58	0.61
GA_{coocur}	0.76	0.74	0.74	0.60	0.60	0.61	0.84	0.82	0.80
G_{katz}	0.73	0.70	0.73	0.50	0.50	0.52	0.67	0.63	0.64
GA_{katz}	0.73	0.70	0.73	0.59	0.59	0.60	0.80	0.78	0.78

Table 5: Effectiveness of DBpedia Spotlight, AGDISTIS and our methods. L denotes the three local features.

methods	Datasets					
	MSNBC-DBpedia			AQUAINT-DBpedia		
	Precision	*Recall*	*F1@MI*	*Precision*	*Recall*	*F1@MI*
DBpedia Spotlight	0.317	0.347	0.331	0.178	0.48	0.26
AGDISTIS	0.796	0.729	0.761	0.777	0.422	0.547
$L + GA_{katz}$	0.811	0.842	0.826	0.778	0.778	0.778
$L + GA_{coocur}$	0.832	0.866	0.849	0.792	0.792	0.792

graph cannot solve the entity linking problem satisfactorily. Anchor entities should therefore be applied to the preference set when computing the personalized EntityRank vector to provide substantial gains.

We compare our approach to state-of-the-art entity linking methods to determine its performance. We choose Rel-RW [18], AGDISTIS [24], and DBpedia Spotlight [22] for comparison with our approach. Because Rel-RW is the outstanding approach by now, AGDISTIS and DBpedia Spotlight are well-known entity linking methods with DBpedia knowledge base.

We evaluate our $L + GA_{katz}$, $L + GA_{coocur}$ methods against DBpedia Spotlight and AGDISTIS on the proposed datasets. The two latter methods achieve entity linking by use of only DBpedia knowledge base similar to us. We obtain the performance of DBpedia Spotlight and AGDISTIS directly from [24]. The comparison results in Table 5 show only the *Precision*, *Recall*, and $F1@MI$ metrics on two datasets as in [24]. The results show that our $L + GA_{katz}$ and $L + GA_{coocur}$ methods outperform the compared methods.

We also evaluate our approach against Rel-RW on the proposed datasets. The experimental benchmarks (i.e., MSNBC, AQUAINT, and ACE2004) used in Rel-RW are based on Wikipedia knowledge base. For ease of comparison, we implement the Rel-RW method based on DBpedia knowledge base, and evaluate it on the DBpedia versions of these datasets. The comparison results in Table 6 show that the $L+GA_{coocur}$ method performs best on the three datasets.

All the compared methods are tested in the setting wherein a fixed set of mentions is given as input, without requiring the mention detection step.

6. CONCLUSION

Entity linking is an important task for many applications such as semantic search, question answering, and knowledge base population. In this paper, we propose a personalized EntityRank-based entity linking approach with DBpedia. We present a method to measure the semantic similarity between a document and a candidate entity by comparing their personalized EntityRank vectors. We conduct two kinds of entity association graph for computing the personalized EntityRank vector. We apply anchor entity when computing the personalized EntityRank vector for a document. The final approach combines the semantic similarity with other local features to rank the candidate entities for each mention. Several experiments are conducted over three datasets. The evaluation results show that our approach outperforms state-of-the-art methods.

7. ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China under grant No. 61502095 and the Natural Science Foundation of Jiangsu Province under Grant BK20140643.

8. REFERENCES

[1] C. Bizer, T. Heath, T. Berners-Lee. Linked Data - The Story So Far, IJSWIS 5(3) (2009) 1-22.

[2] J. Lehmann, C. Bizer, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - a crystallization point for the web of data. Journal of Web Semantics, 7(3) (2009) 154-165.

[3] F. M. Suchanek, G. Kasneci, and G.Weikum. Yago: A Core of Senmantic Knowledge Unifying Wordnet and Wikipedia. In Proceedings of the International Conference on World Wide Web (WWW), (2007) 697-706.

[4] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), (2008) 1247-1250.

[5] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In Proceedings of the Joint Conference on

Table 6: Effectiveness of Rel-RW and our methods. L denotes the three local features.

methods	Datasets								
	MSNBC-DBpedia			AQUAINT-DBpedia			ACE2004-DBpedia		
	Accuracy	*F1@MI*	*F1@MA*	*Accuracy*	*F1@MI*	*F1@MA*	*Accuracy*	*F1@MI*	*F1@MA*
Rel-RW	0.83	0.83	0.83	0.78	0.78	0.79	0.86	0.86	0.83
$L + GA_{katz}$	0.83	0.83	0.83	0.78	0.78	0.79	0.88	0.88	0.84
$L + GA_{coocur}$	0.85	0.85	0.84	0.79	0.79	0.80	0.88	0.88	0.84

Computational Natural Language Learning and Association for Computational Linguistics (COLING-ACL), (1998) 79-85.

[6] R. C. Bunescu and M. Paşca. Using encyclopedic knowledge for named entity disambiguation. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL), (2006) 9-16.

[7] R. Mihalcea and A. Csomai. Wikify! linking documents to encyclopedic knowledge. In Proceedings of the Conference on Information and Knowledge Management (CIKM), (2007) 233-242.

[8] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), (2007) 708-716.

[9] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin. Entity disambiguation for knowledge base population. In Proceedings of the 23rd International Conference on the Association for Computational Linguistics (ACL), (2010) 277-285.

[10] D. Milne and I. H. Witten. Learning to link with wikipedia. In Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM), (2008) 509-518.

[11] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of Wikipedia entities in web text. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), (2009) 457-466.

[12] D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In Proceedings of AAAI Workshop on Wikipedia and Artificial Intelligence, (2008).

[13] P. Ferragina and U. Scaiella. Tagme: on-the-fly annotation of short test fragments (by wikipedia entities). In Proceedings of the 19th ACM Conference on Information and Knowledge Management, (2010) 1625-1628.

[14] L. Ratinov, D. Roth, D. Downey, M. Anderson. Local and global algorithms for disambiguation to wikipedia. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), (2011) 1375-1384.

[15] G. Luo, X. Huang, C. Lin, Z. Nie. Joint Named Entity Recognition and Disambiguation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, (2015) 879-888.

[16] O. E. Ganea, M. Ganea, A. Lucchi, et al. Probabilistic Bag-Of-Hyperlinks Model for Entity Linking. In Proceedings of the 25th international conference on World Wide Web (WWW), (2016) .

[17] T. Hughes and D. Ramage. Lexical semantic relatedness with random graph walks. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), (2007) 581-589.

[18] Z. Guo and D. Barbosa. Robust entity linking via random walks. In Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM), (2014) 499-508.

[19] X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: a graph-based method. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR), (2011) 765?774.

[20] A. Moro, A. Raganato, and R. Navigli. Entity Linking meets Word Sense Disambiguation: a Unified Approach. Transactions of the Association for Computational Linguistics (TACL), (2014:2) 231?244.

[21] I. Hulpuş, N. Prangnawarat, C. Hayes. Path-Based Semantic Relatedness on Linked Data and Its Use to Word and Entity Disambiguation. In Proceedings of the International Conference on the Semantic Web (ISWC), (2015) 442-457.

[22] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. DBpedia spotlight: shedding light on the web of documents. In Proceedings of the 7th International Conference on Semantic Systems, (2011) 1-8.

[23] W. Shen, J. Wang, P. Luo, et al. LINDEN: linking named entities with knowledge base via semantic knowledge. In Proceedings of the 21st international conference on World Wide Web (WWW), (2012) 449-458.

[24] R. Usbeck, AC N. Ngomo, M. Röder, et al. AGDISTIS - Graph-Based Disambiguation of Named Entities Using Linked Data. // In Proceedings of the 2014 International Semantic Web Conference (ISWC) (2014) 457-471.

[25] G. Cheng, D. Xu, Y. Qu. Summarizing entity descriptions for effective and efficient human-centered entity linking. In Proceedings of the 24th International Conference on World Wide Web (WWW), (2015) 184-194.

[26] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford University Database Group, 1998.

[27] L. Katz. A new status index derived from sociometric analysis. Psychometrika 18(1), (1953) 39-43.