LINDEN: Linking named entities with knowledge base via semantic knowledge

Article · April 2012

DOI: 10.1145/2187836.2187898

4 authors:

Wei Shen
Nankai University
11 PUBLICATIONS **601** CITATIONS
SEE PROFILE

Jianyong Wang
Xiamen University
27 PUBLICATIONS **990** CITATIONS
SEE PROFILE

Ping Luo
HP Inc.
79 PUBLICATIONS **1,370** CITATIONS
SEE PROFILE

Min Wang
Fudan University
121 PUBLICATIONS **3,020** CITATIONS
SEE PROFILE

LINDEN: Linking Named Entities with Knowledge Base via Semantic Knowledge

Wei Shen[1], Jianyong Wang[1], Ping Luo[2], Min Wang[2]
[1]Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[2]HP Labs China, Beijing 100084, China
[1]chen-wei09@mails.tsinghua.edu.cn, jianyong@tsinghua.edu.cn
[2]{ping.luo, min.wang6}@hp.com

ABSTRACT

Integrating the extracted facts with an existing knowledge base has raised an urgent need to address the problem of entity linking. Specifically, entity linking is the task to link the entity mention in text with the corresponding real world entity in the existing knowledge base. However, this task is challenging due to name ambiguity, textual inconsistency, and lack of world knowledge in the knowledge base. Several methods have been proposed to tackle this problem, but they are largely based on the co-occurrence statistics of terms between the text around the entity mention and the document associated with the entity. In this paper, we propose LINDEN[1], a novel framework to link named entities in text with a knowledge base unifying Wikipedia and WordNet, by leveraging the rich semantic knowledge embedded in the Wikipedia and the taxonomy of the knowledge base. We extensively evaluate the performance of our proposed LINDEN over two public data sets and empirical results show that LINDEN significantly outperforms the state-of-the-art methods in terms of accuracy.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*

General Terms

Algorithms, Experimentation

Keywords

Entity linking, Knowledge base, Fact integration, Semantic knowledge, Wikipedia

1. INTRODUCTION

Search engine has become the most convenient way for people to find their information on the Web, which is the world's largest encyclopedic source. Unfortunately, in response to the query for the facts or specific attributes about certain named entity, search engine always returns a flat, long list of Web pages containing the name of that entity. The users are then forced either to refine their queries by adding new keywords or to browse through every returned Web page which is quite time consuming. Therefore, the trend to advance the functionality of search engine to a more expressive semantic level has attracted a lot of attention in recent years. To achieve this goal, it is a vital step to construct a comprehensive machine-readable knowledge base about the world's entities, their semantic classes and their mutual relationships. Recently, many large scale publicly available knowledge bases including DBpedia [1], YAGO [27, 26] and KOG [28, 29] have emerged.

As world evolves, new facts come into existence and are digitally expressed on the Web. Therefore, maintaining and growing the existing knowledge bases become more and more important. However, inserting new extracted knowledge derived from the information extraction systems into an existing knowledge base inevitably needs a system to map the entity mention associated with the extracted knowledge to the corresponding real world entity in the knowledge base. This entity linking task is challenging due to name variations and entity ambiguity. In reality, an entity may have multiple surface forms. For example, the entity of "National Basketball Association" has its abbreviation "NBA" and the entity of "New York City" has its nickname "Big Apple". On the contrary, one entity mention may also refer to several different real world entities. For instance, the entity mention of "Michael Jordan" can refer to the famous basketball player, the computer science professor or some other persons.

Entity linking is the task to link a textual entity mention, possibly identified by a named entity recognizer in the unstructured text, with the corresponding real world entity in the existing knowledge base. If the matching entity of certain entity mention does not exist in the knowledge base, NIL (denoting an unlinkable mention) should be returned for this entity mention. This task is also known as entity resolution, record linkage, or entity reconciliation. Entity linking is beneficial for many information extraction applications. For example, relation extraction is the process of discovering useful relationships between named entities mentioned in the text [11, 30, 9], and the extracted relations require the process of mapping entities associated with the relations to the knowledge base before they can be populated into the knowledge base. Besides, a large number of question answering systems rely on their supported knowledge bases to

[1]LINDEN stands for a framework for Linking named entIties with kNowleDge basE via semaNtic knowledge.

WWW 2012, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1229-5/12/04.

give the answer to the user's query. To answer the birth date of the famous basketball player Michael Jordan, the system should firstly leverage the entity linking approach to map the queried "Michael Jordan" to the NBA player, not to the Berkeley's professor; and then it retrieves the birth date of the NBA player named "Michael Jordan" from the knowledge base directly.

The emergence of large scale knowledge bases has spurred great interests in the entity linking task. Several methods [4, 6, 7] have been proposed to address this problem and they all aim to map the entity mention to its corresponding entity page in Wikipedia. Generally speaking, the essential step of entity linking is to define a similarity measure between the text around the entity mention and the document associated with the entity. Previous proposed methods [4, 6, 7] all use the bag of words model to measure the context similarity and consider this kind of similarity as an important feature to make the final decision. The bag of words model represents the context as a term vector consisting of the terms occurring in the window of text and their associated weights. Here, "terms" means words, phrases, named entities or Wikipedia concepts depending on the different methods. Anyway, in the bag of words model, similarity is measured by the co-occurrence statistics of terms and cannot capture various semantic relations existing between concepts. The entity mention would be mapped to the corresponding entity in knowledge base only if the compared texts contain some identical contextual terms. However, by leveraging the semantic relation existing between concepts, the similarity can also be bridged by the semantically related concepts. For instance, we assume the knowledge base contains the following two entities which could be referred by the same name "Michael Jordan":

- Entity name: **Michael J. Jordan**
  Description text: **American basketball player**
- Entity name: **Michael I. Jordan**
  Description text: **Berkeley professor in AI**

When the entity mention appears in the text "Michael Jordan wins NBA champion.", we should map this occurrence of "Michael Jordan" to the American basketball player, because the concept "NBA" around the entity mention is highly semantically related to "American" and "Basketball" which are the concepts appearing in the description text associated with the entity "Michael J. Jordan". While in this situation, the bag of words model cannot work well.

In this paper, we propose LINDEN, a novel framework to link named entities in text with a knowledge base unifying Wikipedia and WordNet by leveraging the semantic knowledge derived from Wikipedia and the taxonomy of the knowledge base. It is assumed that the named entity recognition process has been completed, and we focus on the task of linking the detected named entity mention with the knowledge base. Specifically, we collect a dictionary about the surface forms of entities from four sources in Wikipedia (i.e., entity pages, redirect pages, disambiguation pages and hyperlinks in Wikipedia article), and record the count information for each target entity in the dictionary. Using this dictionary, we can generate a candidate entity list for each entity mention and try to include all the possible corresponding entities of that mention in the generated list. Furthermore, we leverage the count information to define the *link probability* for each candidate entity. Subsequently, we recognize all the Wikipedia concepts in the document where the entity mention appears. By leveraging the link structure of the Wikipedia pages and the taxonomy included in the ontology, we start by constructing a semantic network among the recognized Wikipedia concepts and candidate entities. Via this constructed semantic network, *semantic associativity* which is derived from the Wikipedia link structure and *semantic similarity* measured from the taxonomy of the knowledge base can be calculated among Wikipedia concepts and candidate entities. In addition, we define the *global coherence* for each candidate entity to measure the global document-level topical coherence among the mapping entities in the document. And then we can give a rank to the candidate entity list for each entity mention with the combination of these four measures, *link probability*, *semantic associativity*, *semantic similarity* and *global coherence*. Furthermore, LINDEN learns how to return NIL for the entity mention which has no matching entity in the knowledge base. To validate the effectiveness of LINDEN, we empirically evaluate it over two public data sets (i.e., Cucerzan's ground truth data [6] and the standard TAC[2] data set). The experimental results show that LINDEN greatly outperforms the previous methods in terms of accuracy. The main contributions of this paper are summarized as follows.

- We present LINDEN, a novel framework which leverages the rich semantic information derived from Wikipedia and the taxonomy of the knowledge base to deal with the entity linking task.
- We propose a novel method to measure the *semantic similarity* between Wikipedia concepts based on the taxonomy of the knowledge base.
- We extensively evaluate LINDEN for the entity linking task over two public data sets. The experimental results show that LINDEN can achieve significantly higher accuracy on both data sets compared with the state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 discusses related work and Section 3 introduces the LINDEN framework and some notations. Next, Section 4 describes how to generate candidate entities for the entity mention. Section 5 presents the approach for entity disambiguation, and NIL mention prediction is introduced in Section 6. Section 7 presents our empirical results and Section 8 draws conclusions.

2. RELATED WORK AND DISCUSSION

Name ambiguity is very common on the Web and has raised serious problems in many different areas such as Web people search, question answering and knowledge base population. Before the emergence of large scale publicly available knowledge bases, named entity disambiguation is called coreference resolution and is regarded as a clustering task. Entity mentions of a particular name either within one document or across multiple documents are clustered together, and each resulting cluster represents one specific real world entity. This problem has been addressed by many researchers starting from Bagga and Baldwin [2], who used the bag of

[2]http://www.nist.gov/tac/

words model to represent the context of the entity mention and applied the agglomerative clustering technique based on the vector cosine similarity. Mann and Yarowsky [16] extended the work by adding a rich feature space of biographic facts. Pedersen et al. [25] employed the statistically significant bigrams to represent the context of a name observation. After that, several methods [14, 15, 3] tried to capture the semantic relation between terms via constructing social networks to add the background knowledge for disambiguation. The work in [22, 10] adopted the graph based framework to extend the similarity metric to disambiguate the entity mentions effectively. However, all these studies focus on clustering all mentions of an entity within a given corpus, which are insufficient for the entity linking task.

As several knowledge bases like DBpedia [1] and YAGO [27, 26] are available publicly, researchers have shown a great interest in mapping the textual entity mention to its corresponding entity in the knowledge base. Bunescu and Pasca [4] firstly tackled this problem by exploiting a set of useful features derived from Wikipedia for entity detection and disambiguation. They leveraged the bag of words model to measure the cosine similarity between the context of the mention and the text of the Wikipedia article. Besides, to overcome the deficiency of the bag of words model, they used a disambiguation SVM kernel which models the magnitude of each word-category correlation based on the Wikipedia taxonomy. The work proposed by Cucerzan [6] is the first system to recognize the global document-level topical coherence of the entities. The system 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. This work assumes that all entity mentions have the corresponding entities in the knowledge base, however, this assumption fails for a large number of mentions in reality. The learning based solution in [7] focuses on the classification framework to resolve entity linking. It develops a rich set of features based on the entity mention, the source document and the knowledge base entry, and then uses a SVM ranker to score each candidate entity. Moreover, this solution incorporates NIL prediction into the ranker, which obviates hand tuning. However, the performance of these previous methods is largely based on the feature of context similarity which depends on the term co-occurrence between the text around the entity mention and the document associated with the entity. Therefore, they ignore the semantic knowledge existing between concepts. Furthermore, the knowledge bases used in these methods are directly derived from the Wikipedia, and the categories in Wikipedia are not clean and well-formed enough for the ontological purpose although they are indeed arranged in a hierarchy. Hence, the semantic knowledge embedded in the taxonomy of concepts cannot be well taken advantage of by these methods.

The task of entity linking is similar to the lexical task of word sense disambiguation (WSD) in some aspects. The task of WSD aims to assign dictionary meanings to all instances of a predefined set of polysemous words in a corpora [23, 18, 24]. For instance, it has to choose whether the word "tree" in some specific context refers to the meaning of plant or data structure in the field of computer science. Recently, people start to use Wikipedia as a resource for word sense disambiguation. Given an input document, these systems are able to automatically enrich the input text with links to Wikipedia pages [19, 21, 12]. However, this task is different from our entity linking task in several respects: firstly, these systems have to decide whether the detected terms or phrases are important enough in the document to be linked to Wikipedia due to considering the system users' experience, which raises the problem of tradeoff between precision and recall. On the contrary, entity linking is the task to just map every detected entity mention in the text to the knowledge base to pursue high accuracy. Secondly, the named entity mentions like common person or place names have much higher average ambiguity compared with the keywords or concepts in the task of word sense disambiguation. Therefore, the entity linking task has much more challenges in comparison with the WSD task. Thirdly, the entity linking task has to encounter the problem that some entity mentions have no matching entities in the knowledge base. Consequently, it must learn how to predict NIL for the unlinkable mentions, while the word sense disambiguation task has no such problem.

Table 1: Notations

d	A document to be processed
M_0	All named entity mentions in d
$m \in M_0$	A named entity mention required to be linked
E	All entities in KB
$e \in E$	An entity label, here, the entity name in KB
E_m	The set of candidate entities for mention m
E_0	All candidate entities for all mentions in M_0
NIL	The label for the unlinkable mention
Γ_d	The set of context concepts in d
$F_m(e)$	The feature vector for entity $e \in E_m$
$\overrightarrow{w}$	Weight vector
$Score_m(e)$	Score of entity $e \in E_m$
τ	Threshold for returning NIL
$LP(e\|m)$	The *link probability* of entity e, given m
$SA(e)$	*Semantic associativity* of entity e with Γ_d
$SS(e)$	*Semantic similarity* of entity e with Γ_d
$GC(e)$	*Global coherence* of entity e in d

3. THE LINDEN FRAMEWORK AND NOTATIONS

In this paper, entity linking is defined as the task to map a textual named entity mention m, already recognized in the unstructured text, to the corresponding real world entity e in the knowledge base. If the matching entity e for entity mention m does not exist in the knowledge base, we should return NIL for m. The knowledge base we adopt in this work is YAGO [27, 26], an open-domain ontology combining Wikipedia and WordNet with high coverage and quality. The reasons why we choose YAGO as the knowledge base are as follows. On one hand, YAGO has the vast amount of entities in the same order of magnitude as Wikipedia. On the other hand, it adopts the clean taxonomy of concepts from WordNet [8] which can be made fully use of by our LINDEN. Currently, YAGO contains over one million entities and five million facts about them.

We process one document at a time, so we consider the entity mentions appearing in one document together. Given an input document d, M_0 is the set of named entity mentions

which need to be mapped in d. A named entity mention $m \in M_0$ is a token sequence of a named entity that is potentially linked with an entity in the knowledge base, which has been detected beforehand. E is the set of all entities in the knowledge base, and an entity is expressed as the entity name in the knowledge base and denoted as e. Since some mentions' mapping entities do not exist in the knowledge base, we define this kind of mentions as unlinkable mentions and give NIL as a special label denoting "unlinkable". In this paper, we propose LINDEN, a framework to address this entity linking task with three modules as follows:

- **Candidate Entity Generation**
  For each named entity mention $m \in M_0$, we retrieve the set of candidate entities E_m in this module. Using a dictionary collected from four sources in Wikipedia (i.e., entity pages, redirect pages, disambiguation pages and hyperlinks in Wikipedia article), we try to include all the possible candidate entities for each $m \in M_0$ in E_m. E_0 is the set of all candidate entities for all mentions in M_0.

- **Named Entity Disambiguation**
  In most cases, the size of E_m is larger than one, so we define a scoring measure for each $e \in E_m$ and give a rank to E_m to find which entity $e \in E_m$ is the mostly likely link for m. We firstly recognize all the Wikipedia concepts Γ_d in the context of d and regard them as *context concepts* to represent the context of d. And then we define a rich set of features and generate a feature vector $F_m(e)$ for each $e \in E_m$. The features used in LINDEN are mainly based on the *link probability* $LP(e|m)$, *semantic associativity* $SA(e)$ of entity e with the context concepts in Γ_d derived from the Wikipedia link structure, *semantic similarity* $SS(e)$ of entity e with the context concepts in Γ_d measured from the taxonomy of YAGO, and *global coherence* $GC(e)$ of entity e with the other mapping entities associated with the mentions $m' \neq m \in M_0$. We also learn a weight vector $\overrightarrow{w}$ which gives different weights for each feature element in $F_m(e)$. Then we can calculate a score $\overrightarrow{w} \cdot F_m(e)$ for each $e \in E_m$ and rank the candidates according to their $Score_m(e)$.

- **Unlinkable Mention Prediction**
  To deal with the problem of predicting unlinkable mentions, we learn a threshold τ in this module to validate whether the entity e_{top} which has the highest score in E_m is the target entity for mention m. If $Score_m(e_{top})$ is smaller than the learned threshold τ, we return NIL for mention m.

Those three modules are introduced in the following sections in details and some notations used in this paper are summarized in Table 1.

4. CANDIDATE ENTITY GENERATION

Given an entity mention $m \in M_0$, we generate the set of candidate entities E_m in this module. Intuitively, the candidates in E_m should have the name of the surface form of m. To solve this problem, we need to build a dictionary that contains vast amount of information about the surface forms of entities, like name variations, abbreviations, confusable names, spelling variations, nicknames, etc. We take advantage of the huge amount of knowledge available in Wikipedia, a free online encyclopedia created through decentralized, collective efforts of thousands of users[3]. Wikipedia is the largest encyclopedia in the world and is also a very dynamic and quickly growing resource. English Wikipedia contains over 3,500,000 articles and new articles are added within days after their occurrence. The structure of Wikipedia provides a set of useful features for the construction of the dictionary we need, such as redirect pages, disambiguation pages and hyperlinks in Wikipedia article. Besides, Wikipedia has high coverage of named entities [31], which is profitable for constructing our dictionary. We use the following four structures of Wikipedia to build the dictionary about the surface forms of entities:

- **Entity pages**: Each entity page in Wikipedia describes a single entity and contains the information focusing on this entity. Generally, the title of each page is the most common name for the entity described in this page, e.g., the page title "Microsoft" for that giant software company headquartered in Redmond. When the name of the entity is ambiguous, it is further qualified with a parenthetical expression. For example, the article for the English goalkeeper Michael Jordan has the title "Michael Jordan (footballer)". Therefore, we store not only the exact article title but also the surface form from which we eliminate appositives, i.e., "Michael Jordan" in this example.

- **Redirect pages**: A redirect page exists for each alternative name which can be used to refer to an existing entity in Wikipedia. For example, the article titled "Microsoft Corporation" which is the full name of "Microsoft" contains a pointer to the article titled "Microsoft". Redirect pages often indicate synonym terms, abbreviations or other variations of the pointed entities.

- **Disambiguation pages**: When multiple entities in Wikipedia could be given the same name, a disambiguation page is created to separate them and contains a list of references to those entities. For example, the disambiguation page for the name "Michael Jordan" lists eight associated entities having the same name of "Michael Jordan" including the famous NBA player and the Berkeley professor. These disambiguation pages are very useful in extracting abbreviations or other aliases of entities.

- **Hyperlinks in Wikipedia article**: The article in Wikipedia often contains hyperlinks which link to the pages of entities mentioned in this article. The anchor text of a link pointing to an entity page provides a very useful source of synonyms and other variations of the entity, and can be regarded as the surface form of that linked entity.

Using the above mentioned structures in Wikipedia, we can construct the dictionary containing all surface forms for each entity. In the mean time, we record the count information for each target entity which is linked by some surface forms as well. An example of the dictionary is shown in Table 2. For each mention $m \in M_0$, we look up the dictionary

[3]http://www.wikipedia.org/

Table 2: An example of the dictionary

Surface form	Target entity	Count
Microsoft Corporation	Microsoft	16
Michael Jordan	Michael Jordan	65
	Michael I. Jordan	10
	Michael Jordan (mycologist)	7
	Michael Jordan (footballer)	3
	...	...
New York	New York City	121
	New York (magazine)	12
	New York (film)	7
	"New York" (Eskimo Joe song)	5
	...	...

and search for the mention m directly in the field of surface forms. If a hit is found, we add all target entities of that surface form m to the set of candidate entities E_m.

It can be seen from the count information in Table 2 that each $e \in E_m$ having the same surface form m has different commonness and some entities are very obscure and rare for the given surface form m. For example, for the surface form "New York", the entity "New York (film)" is much rarer than "New York City", and in most cases when people mention "New York", they mean the city of New York rather than the film whose name is also "New York". Hence, we take advantage of this count information and define the *link probability* $LP(e|m)$ for entity e as:

$$LP(e|m) = \frac{count_m(e)}{\sum_{e_i \in E_m} count_m(e_i)} \quad (1)$$

where $count_m(e)$ is the number of links which point to entity e and have the surface form m. The candidate entities with very low *link probability* will be discarded.

5. NAMED ENTITY DISAMBIGUATION

In this section, we describe how to give a rank to E_m when the size of E_m generated in Section 4 is larger than one. Our guiding premise is that a document largely refers to coherent entities or concepts from one or a few related topics, and we exploit this "topical coherence" for named entity disambiguation. To achieve this goal, we firstly recognize all the Wikipedia concepts Γ_d in the document d, and by leveraging the rich semantic knowledge embedded in Wikipedia and YAGO, we construct a semantic network among the recognized Wikipedia concepts Γ_d and candidate entities E_0, which will be described in Section 5.1. From the semantic network, we can see the rich semantic relations existing among Wikipedia concepts Γ_d and candidate entities E_0, however, it does not explicitly provide the value of the semantic relation's strength. In order to measure the semantic relation's strength, we show how to compute the *semantic associativity* $SA(e)$ of entity e based on the Wikipedia link structure and *semantic similarity* $SS(e)$ of entity e derived from the taxonomy of YAGO in Section 5.2 and Section 5.3, respectively. Besides the semantic relation existing between Γ_d and E_0, we exploit the global document-level topical coherence among entities which are chosen to be mapped to by the mentions in M_0. The *global coherence* $GC(e)$ of entity e is measured as the average semantic associativity of entity e to the other mapping entities associated with the mentions $m' \neq m \in M_0$, which will be introduced in details in Section 5.4). Combining those features introduced above, we generate a feature vector $F_m(e)$ for each $e \in E_m$ and learn a weight vector $\overrightarrow{w}$ which gives different weights for each feature element in $F_m(e)$. Then we calculate a score $\overrightarrow{w} \cdot F_m(e)$ for each $e \in E_m$ and rank the candidates according to their $Score_m(e)$, which will be introduced in Section 5.5.

5.1 Semantic Network Construction

To construct the semantic network, we start by recognizing the Wikipedia concepts Γ_d in the context of the document d, and regard them as context concepts to represent the context of d. For the general textual document, we utilize the open source toolkit Wikipedia-Miner[4] to detect the Wikipedia concepts appearing in the context. The Wikipedia-Miner toolkit takes the general unstructured text as input and uses the machine learning approach to detect the Wikipedia concepts in the input document [21]. For instance, the entity mention of "Michael Jordan" occurs in a document containing such a sentence, "The Chicago Bulls' player Michael Jordan won his first NBA championship in 1991.". For this sentence, we firstly remove the entity mention, and then utilize this Wikipedia-Miner toolkit to obtain four Wikipedia concepts, i.e., Chicago Bulls, National Basketball Association, NBA Finals and Chicago. Therefore, it can be seen that these detected Wikipedia concepts are highly semantically related to the NBA player Michael Jordan, and we can leverage this semantic information to link this entity mention "Michael Jordan" with the corresponding real world entity (i.e., the NBA player Michael Jordan) in the knowledge base effectively.

As we know, for the document from the Wikipedia, it has its special layout to organize its content, i.e., *Wiki markup*[5]. The references to other Wikipedia concepts in the Wikipedia document are within pairs of double square brackets, which can be exploited to identify the Wikipedia concepts easily. Illustratively, the Wikipedia article for the entity *Bill Gates*, the billionaire, contains the following *Wikitext*:

Gates was born in [[Seattle]], Washington, of [[English people|English]], [[Germans|German]], and Scotch-Irish descent.

In this *Wikitext*, there are three references which surround with double square brackets. If a reference contains a vertical bar (e.g., "English people|English"), then the text at the left of the bar is the name of the referred Wikipedia concept (e.g., "English people"), while the text at the right of the bar (e.g., "English") is the anchor text of this link. Otherwise, the anchor text is identical to the title of the Wikipedia concept referred (e.g., "Seattle"). Henceforth, for the Wikipedia document, we can identify the Wikipedia concepts appearing in it directly by leveraging the characteristic of *Wiki markup*.

Wikipedia contains rich semantic information between concepts and the hyperlink structure of Wikipedia articles is one important form of expressing semantics. Therefore, we add all the link relations and the associated Wikipedia articles to our constructed semantic network. Moreover, the taxonomy of concepts in YAGO also expresses the semantic relation between Wikipedia concepts which we call *semantic similarity*. Hence, we add the taxonomic relations among these detected context concepts and candidate entities as well. Figure 1 shows an example of the constructed seman-

[4]http://wikipedia-miner.sourceforge.net/index.htm
[5]http://en.wikipedia.org/wiki/Wiki_markup

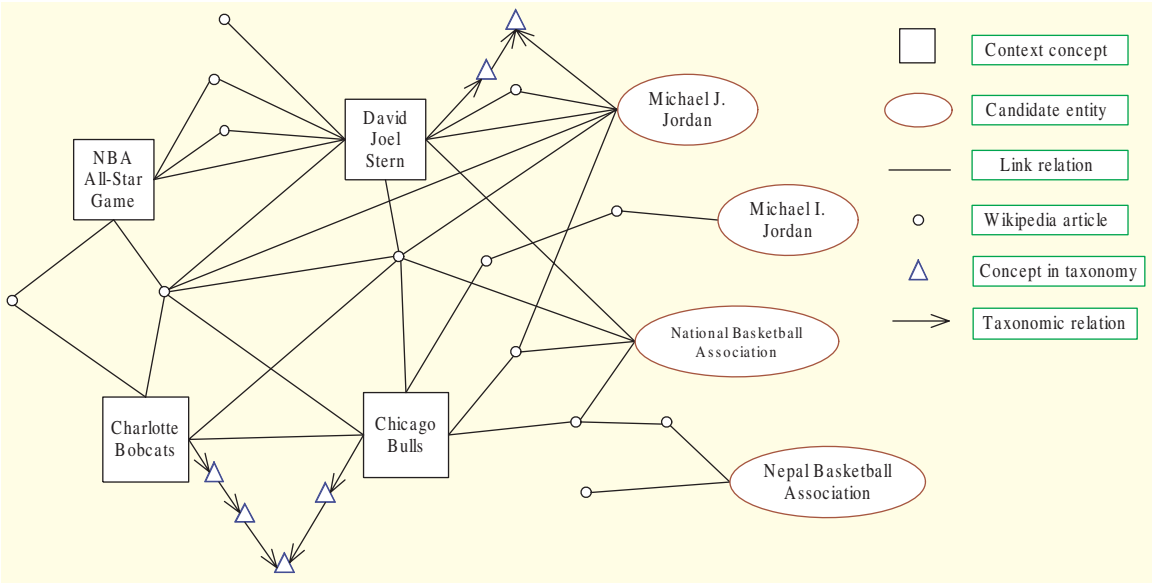

Figure 1: An example of the constructed semantic network

tic network. The four candidate entities in Figure 1 are generated from two entity mentions (i.e., "Michael Jordan" and "NBA"), and each of the entity mentions has two candidate entities respectively. From the constructed semantic network, we can see that the candidate entities "Michael J. Jordan" and "National Basketball Association" are more semantically related to the four context concepts compared with the other two candidate entities. Moreover, the semantic relations between "Michael J. Jordan" and "National Basketball Association" also show the highly global topical coherence. Therefore, we can predict that "Michael J. Jordan" and "National Basketball Association" are the mapping entities for the entity mentions "Michael Jordan" and "NBA", respectively.

5.2 Semantic Associativity

Though the link relations among the context concepts Γ_d and candidate entities E_0 in Figure 1 express high semantic relations, this structure does not explicitly provide the exact value of the semantic relation's strength. In order to measure the strength of the link relation, we adopt the Wikipedia Link-based Measure (WLM) described in [20] to calculate the *semantic associativity* between Wikipedia concepts. Since all the context concepts Γ_d and candidate entities E_0 in our work are Wikipedia concepts, we can leverage this measure of WLM directly. The WLM which is modeled from the Normalized Google Distance [5] is based on Wikipedia's hyperlink structure. Given two Wikipedia concepts e_1 and e_2, we define the *semantic associativity* between them as follows:

$$SmtAss(e_1, e_2) = 1 - \frac{log(max(|E_1|, |E_2|)) - log(|E_1 \bigcap E_2|)}{log(|W|) - log(min(|E_1|, |E_2|))} \quad (2)$$

where E_1 and E_2 are the sets of Wikipedia concepts that link to e_1 and e_2 respectively, and W is the set of all concepts in Wikipedia. The numerator is a slight variation of Jaccard similarity and the denominator is inversely related to $min(|E_1|, |E_2|)$. Therefore, this definition gives higher value to more related concept pair. The feature value of *semantic associativity* $SA(e)$ for each entity e is defined as the average of its *semantic associativity* to each context concept in Γ_d:

$$SA(e) = \frac{\sum_{cc \in \Gamma_d} SmtAss(cc, e)}{|\Gamma_d|} \quad (3)$$

5.3 Semantic Similarity

In this subsection, we propose a novel method to measure the *semantic similarity* between Wikipedia concepts based on the taxonomy of the knowledge base. According to the rules of constructing YAGO ontology in [27], each Wikipedia concept may have multiple super classes in the taxonomy. Given two Wikipedia concepts e_1 and e_2, we assume the sets of their super classes are Φ_{e_1} and Φ_{e_2}, respectively. To measure the *semantic similarity* between Wikipedia concepts, we firstly define how to calculate the *semantic similarity* between the sets of their super classes. Since the sizes of Φ_{e_1} and Φ_{e_2}, and the elements in Φ_{e_1} and Φ_{e_2} are likely to be different, we start by defining the correspondence between the elements of classes from one set to another set. For each class C_1 in the set Φ_{e_1}, we assign a target class $\varepsilon(C_1)$ in another set Φ_{e_2} as follows:

$$\varepsilon(C_1) = \arg\max_{C_2 \in \Phi_{e_2}} sim(C_1, C_2) \quad (4)$$

where $sim(C_1, C_2)$ is the *semantic similarity* between two classes C_1 and C_2, and $\varepsilon(C_1)$ is the class in Φ_{e_2} which maximizes the *semantic similarity* between these two classes. To compute $sim(C_1, C_2)$, we adopt the approach introduced in [13] which is an information-theoretic method. Assuming the taxonomy is a tree and C is a class in the taxonomy, the amount of information contained in the statement "$x \in C$" is $-log(P(C))$, where $P(C)$ is the probability that a randomly selected object belongs to the subtree with the root of C in the taxonomy. We assume that C_0 is the class which is the most specific class that subsumes both C_1 and C_2 in the taxonomy, in other words, C_0 is the root of the smallest

subtree that contains both C_1 and C_2 in the taxonomy. The following is the definition of the *semantic similarity* between two classes C_1 and C_2 in the taxonomy:

$$sim(C_1, C_2) = \frac{2 \times log(P(C_0))}{log(P(C_1)) + log(P(C_2))} \quad (5)$$

Next, we can calculate the *semantic similarity* from one set of classes Φ_{e_1} to another set of classes Φ_{e_2}:

$$sim(\Phi_{e_1} \rightarrow \Phi_{e_2}) = \frac{\sum_{C_1 \in \Phi_{e_1}} sim(C_1, \varepsilon(C_1))}{|\Phi_{e_1}|} \quad (6)$$

Identically, we can calculate the *semantic similarity* from Φ_{e_2} to Φ_{e_1}, i.e., $sim(\Phi_{e_2} \rightarrow \Phi_{e_1})$, in the similar way to that in Formula 6. Based on the definitions mentioned above, we can define the *semantic similarity* between Wikipedia concepts e_1 and e_2 as the average of the *semantic similarity* from Φ_{e_1} to Φ_{e_2} and that from Φ_{e_2} to Φ_{e_1}:

$$SmtSim(e_1, e_2) = \frac{sim(\Phi_{e_1} \rightarrow \Phi_{e_2}) + sim(\Phi_{e_2} \rightarrow \Phi_{e_1})}{2} \quad (7)$$

Intuitively, there might be some context concepts having similar types to entity e but it is unlikely that all the types of context concepts are similar to the entity e's type. Therefore, we define the set of k context concepts in Γ_d which have the highest *semantic similarity* with entity e as Θ_k and the parameter k is set empirically. We calculate the feature value *semantic similarity* $SS(e)$ for entity e as follows:

$$SS(e) = \frac{\sum_{cc \in \Theta_k} SmtSim(cc, e)}{k} \quad (8)$$

5.4 Global Coherence

In this subsection, we exploit the global document-level topical coherence among entities which should be linked with by the mentions in M_0. In this work, the *global coherence* $GC(e)$ of entity e is measured as the average *semantic associativity* of entity e to the mapping entities of the other mentions m', where $m' \neq m \in M_0$. If $e_{m'}$ is the mapping entity of mention m', then for entity e, the *global coherence* $GC(e)$ is defined as

$$GC(e) = \frac{\Sigma_{m' \neq m \in M_0}(SmtAss(e_{m'}, e))}{|M_0| - 1} \quad (9)$$

Unfortunately, $e_{m'}$, the mapping entity of mention m', is unknown to us and needs to be assigned in this task. It can be seen that the assignment of an entity to a mention depends on all the other assignments made for other mentions, which makes this a difficult optimization problem. In this paper, we adopt an arguably more robust strategy which is to calculate the average *semantic associativity* of entity e to the most likely assigned entities of the other mentions. The most likely assigned entity $e'_{m'}$ for mention m' is considered as the candidate entity which has the maximum *link probability* in $E_{m'}$.

$$e'_{m'} = \arg\max_{e' \in E_{m'}} LP(e'|m') \quad (10)$$

Now the computation of *global coherence* $GC(e)$ in Formula 9 is simplified as shown in Formula 11 which can be computed directly.

$$GC(e) = \frac{\Sigma_{m' \neq m \in M_0}(SmtAss(e'_{m'}, e))}{|M_0| - 1} \quad (11)$$

5.5 Candidates Ranking

Combining those features introduced in the subsections above, we can generate a feature vector $F_m(e)$ for each $e \in E_m$ where $F_m(e) =< LP(e|m), SA(e), SS(e), GC(e) >$. The different features in $F_m(e)$ have different degrees of importance for the entity disambiguation task. Therefore, we learn a weight vector $\overrightarrow{w}$ which gives different weights for each feature element in $F_m(e)$. Then we calculate $Score_m(e)$ for each $e \in E_m$,where $Score_m(e) = \overrightarrow{w} \cdot F_m(e)$. Finally, we rank the candidates according to their $Score_m(e)$ and pick $e_{top} = \arg\max_{e \in E_m} Score_m(e)$ as the predicted mapping entity for mention m.

To learn $\overrightarrow{w}$, we use a max-margin technique based on the training data set. Given the ground truth mapping entity e^* for mention m, we assume that $Score_m(e^*)$ is larger than any other $Score_m(e)$ with a margin, where $e \in E_m$ and $e \neq e^*$. This gives us the usual SVM linear constraints for all linkable mentions:

$$\overrightarrow{w} \cdot F_m(e^*) - \overrightarrow{w} \cdot F_m(e) \geq 1 - \xi_m \quad (12)$$

and we minimize over $\xi_m \geq 0$ and the objective $||\overrightarrow{w}||_2^2 + \alpha \Sigma_m \xi_m$ where α is the usual balancing parameter.

6. UNLINKABLE MENTION PREDICTION

The approach discussed above implicitly assumes that the knowledge base contains all the matching entities of the mentions. But in practice, this assumption fails in many cases without a doubt. Therefore, we have to deal with the problem of predicting unlinkable mentions in LINDEN. Firstly, if the size of E_m generated in the Candidate Entities Generation module for mention m is equal to zero, we predict mention m as an unlinkable mention and return NIL for mention m undoubtedly. If the size of E_m is equal to one, we assume the only entity in E_m as e_{top} and regard it as the predicted mapping entity for mention m. When the size of E_m generated in Section 4 is larger than one, we give a score to each $e \in E_m$ in the Named Entity Disambiguation module and pick $e_{top} = \arg\max_{e \in E_m} Score_m(e)$ as the predicted mapping entity for mention m. In this module, our task is to validate whether the predicted entity e_{top} is the target entity for mention m. We adopt a simple method and learn a threshold τ to validate the predicted entity e_{top}. If $Score_m(e_{top})$ is greater than the learned threshold τ, we return e_{top} as the target entity for mention m, otherwise we return NIL for m.

7. EXPERIMENTAL RESULTS

7.1 Data sets

To evaluate the performance of LINDEN, we have to choose some test data sets available publicly. The experimental data used by Bunescu and Pasca [4] is not publicly available. The newswire data used by Cucerzan [6] (which we refer to "CZ") is available and we use it to test our LINDEN. The data set "IITB" built by Kulkarni et al. in [12] is unsuitable for our task since they annotated a broad set of types of entities rather than named entities due to their aggressive recall target, which is similar to the WSD task addressed in [19, 21]. In addition, entity linking is initiated as a task in the track of Knowledge Base Population (KBP) at the Text Analysis Conference (TAC) recently. The data

Table 3: Experimental results over the CZ data set

	# of total mentions	LINDEN		Cucerzan	
		#	Accu.	#	Accu.
All	614	**581**	**0.9463**	549	0.8941
Linkable	522	**493**	**0.9444**	466	0.8927
Unlinkable	92	**88**	**0.9565**	83	0.9022

set for TAC-KBP2009[6] is available for us so we use it as another test data set for LINDEN. We downloaded the October 2010 version of Wikipedia and YAGO(1)[7] of version 2009-w10-5 for our experiments.

In the following subsections, we will introduce the experimental results of LINDEN over the two test data sets, i.e., the CZ data set and the TAC-KBP2009 data set, respectively.

7.2 Experimental results on the CZ data set

To evaluate the performance of LINDEN, in this paper we adopted the evaluation measure *Accuracy (Accu.)* which is used in most work about entity linking [4, 6, 7] and TAC-KBP2009[17]. The accuracy is calculated as the number of correctly linked entity mentions divided by the total number of all mentions. The weight vector $\overrightarrow{w}$ and all parameters are tuned using 10-fold cross validation over the CZ data set. Since in the CZ data set, it is fairly common for one of the mentions of an entity in the document to be a long and typical surface form of that entity (e.g., Bob Nardelli), while the other mentions of the same entity are shorter surface forms (e.g., Nardelli). To address this problem, we used a simple in-document coreference resolution method which is to map short surface form to longer surface form in the same document before generating candidate entities for mentions.

The original data set in [6] contains 20 news stories which include the top two stories in each of the ten MSNBC news categories of January 2, 2007. But unfortunately, one of the MSNBC news articles is no longer available, so we used the remaining 19 articles. Meanwhile, the contents of these articles have changed slightly compared with them at the time Cucerzan annotated, therefore, we removed the entity mentions which did not appear in the articles we downloaded. Lastly, we obtained 614 entity mentions in those articles to construct the CZ data set, in which there are 522 entity mentions which are manually linked to the knowledge base and another 92 mentions are unlinkable.

Since the CZ data set we used in this experiment is different from the original data set in [6], the accuracy of 0.914 achieved by Cucerzan's system reported in [6] is not comparable. In order to give a fair comparison, we implemented the algorithm of Cucerzan's system and evaluated it over the CZ data set. The experimental results of LINDEN and our implemented Cucerzan's system over the CZ data set are shown in Table 3. Besides the number of total mentions, we also show the number of correctly linked mentions and the accuracy for both LINDEN and our implemented Cucerzan's system with different types (i.e., all, linkable and unlinkable). From the results in Table 3 we can see that LINDEN achieves significantly higher accuracy compared with the Cucerzan's system in all aspects. Though the system reported in [7] obtained the accuracy of 0.9469 in CZ data set, this result is not comparative with the result of our system due to the following reasons: firstly, they removed 297 mentions not recognized as entities by SERIF from the test data set; Secondly, the knowledge base they used is a subset of Wikipedia, which enabled many mentions to be unlinkable. Consequently, the test data set they used to evaluate their method contains total 452 mentions in which there are 187 unlinkable mentions, which is greatly different from the original test data set in [6].

[6]http://apl.jhu.edu/~paulmac/kbp.html
[7]http://www.mpi-inf.mpg.de/yago-naga/yago/

Table 4: Feature set effectiveness over the CZ data set

Feature Set	All		Linkable		Unlinkable	
	#	Accu.	#	Accu.	#	Accu.
LP	541	0.8811	453	0.8678	88	0.9565
LP+SA	573	0.9332	486	0.9310	87	0.9457
LP+SS	557	0.9072	467	0.8946	90	0.9783
LP+GC	562	0.9153	474	0.9080	88	0.9565
LP+SA+SS	579	0.9430	492	0.9425	87	0.9457
LP+SA+SS+GC	**581**	**0.9463**	493	0.9444	88	0.9565

We also analyzed the effectiveness of different feature sets to LINDEN's performance. Table 4 shows the accuracy and the number of correctly linked mentions obtained by LINDEN with different feature sets. It can be seen from Table 4 that every feature has a positive impact on the performance of LINDEN, and with the combination of all features LINDEN can obtain the best result. The improvement achieved by adding *semantic associativity* (SA) feature to *link probability* (LP) feature is greater than what can be achieved by adding one of the other two features (i.e., *semantic similarity* (SS) feature and *global coherence* (GC) feature), which indicates that the feature of *semantic associativity* is quite useful to deal with entity linking problem. Since these features correlate with each other quite closely, we can just get slight improvement by adding SS feature to the feature set of LP and SA, and the same thing occurs when adding GC feature to the feature set of LP, SA and SS.

7.3 Experimental results on the TAC-KBP2009 data set

The TAC-KBP2009 test data set consists of 3904 entity mentions (which they call queries) in which 1675 entity mentions can be aligned to their knowledge base. There are 2229 entity mentions which cannot be mapped to their knowledge base and hold 57% of the total queries. The reason why most queries are unlinkable in the TAC-KBP2009 data set is that the knowledge base they used to annotate these queries is a subset of Wikipedia and only contains the set of entities that have infoboxes in Wikipedia. The weight vector $\overrightarrow{w}$ and all parameters are tuned using 10-fold cross validation over the TAC-KBP2009 data set.

Since in LINDEN we use the whole information in Wikipedia to generate candidate entities in the Candidate Entities Generation module, we have to add some unlinkable mentions prediction strategies to the module of Unlinkable Mentions Prediction described in Section 6 in order to directly use the ground truth annotation of the TAC-KBP2009 data set. Before we learn the threshold τ to validate the predicted entity e_{top} in the Unlinkable Mentions Prediction module, we firstly verify whether the predicted entity e_{top} exists in the knowledge base of TAC-KBP2009. If it exists in the knowledge base of TAC-KBP2009, we go on the following steps

Table 5: Experimental results over the TAC-KBP2009 data set compared with top 4 ranked systems in TAC-KBP2009

System	Accu. of all	Accu. of linkable	Accu. of unlinkable
Rank 1	0.8217	0.7654	0.8641
Rank 2	0.8033	0.7725	0.8241
Rank 3	0.7984	0.7063	0.8677
Rank 4	0.7884	0.7588	0.8107
LINDEN	**0.8432**	**0.7988**	**0.8782**

Table 6: Feature set effectiveness over the TAC-KBP2009 data set

Feature Set	All		Linkable		Unlinkable	
	#	Accu.	#	Accu.	#	Accu.
LP	3109	0.7964	1149	0.6860	1960	0.8793
LP+SA	3258	0.8345	1316	0.7857	1942	0.8728
LP+SS	3186	0.8161	1192	0.7116	1994	0.8962
LP+SA+SS	**3292**	**0.8432**	1338	0.7988	1954	0.8782

introduced in Section 6, otherwise, we return NIL for this mention directly.

In addition, the track of TAC-KBP2009 requires the systems who participate in the track to process the queries independently from one to another, which means they require that systems cannot leverage the knowledge among the set of queries according to the task description of TAC-KBP2009[8]. Meanwhile, the total 3904 entity mentions exist in 3688 documents each of which has at most two mentions in its context according to the statistics of the TAC-KBP2009 data set. Therefore, we removed the feature of *global coherence* (GC) introduced in Subsection 5.4 in the following experiments for two reasons. On one hand, the systems in TAC-KBP2009 did not leverage the knowledge among the set of queries, so we want to give a relatively fair comparison of LINDEN with these systems. On the other hand, the *global coherence* feature can hardly have any positive impacts on the performance of LINDEN in this data set according to the data distribution mentioned above. In addition, due to many spelling errors existing in the set of queries, we also try to correct them using the query spelling correction supplied by Google.

The experimental results of LINDEN over the TAC-KBP2009 data set are shown in Table 5. The results of the top 4 systems which perform best in the track of TAC-KBP2009 [17] are also shown in Table 5 for the purpose of comparison. Moreover, the system introduced in [7] is the rank 3 system in TAC-KBP2009 track and obtains the overall accuracy of 0.7984 over this TAC-KBP2009 data set. The results in Table 5 show that LINDEN outperforms the best systems in TAC-KBP2009, which demonstrates the effectiveness of LINDEN.

We also show the effectiveness of different feature sets to LINDEN's performance over the TAC-KBP2009 data set in Table 6. The overall accuracy of LINDEN only using *link probability* (LP) feature is higher than the rank 4 system in TAC-KBP2009, which demonstrates that the *link probability* feature is also quite useful for this task. From the other results in Table 6 we can get the similar conclusion to what we get over the CZ data set. The impact of *semantic associativity* (SA) feature is greater than the *semantic similarity* (SS) feature when dealing with entity linking problem, and with the combination of all features LINDEN can obtain the best result.

[8]http://apl.jhu.edu/~paulmac/kbp/090601-KBPTaskGuidelines.pdf

8. CONCLUSION

Entity linking is a very important task for many applications such as Web people search, question answering and knowledge base population. In this paper, we propose LINDEN, a novel framework to link named entities in text with YAGO, a knowledge base unifying Wikipedia and WordNet. By leveraging the rich semantic knowledge derived from the Wikipedia and the taxonomy of YAGO, LINDEN can obtain great results on the entity linking task. A large number of experiments were conducted over two public data sets, i.e., the CZ data set and the TAC-KBP2009 data set. Empirical results show that LINDEN significantly outperforms the state-of-the-art methods in terms of accuracy. Moreover, all features adopted by LINDEN are quite effective for the entity linking task.

9. ACKNOWLEDGMENTS

This work was supported in part by National Basic Research Program of China (973 Program) under Grant No. 2011CB302206, National Natural Science Foundation of China under Grant No. 60833003, and an HP Labs Innovation Research Program award.

10. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of ISWC*, pages 11–15, 2007.

[2] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of COLING*, pages 79–85, 1998.

[3] R. Bekkerman and A. McCallum. Disambiguating web appearances of people in a social network. In *Proceedings of WWW*, pages 463–470, 2005.

[4] R. Bunescu and M. Pasca. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proceedings of EACL*, pages 9–16, 2006.

[5] R. L. Cilibrasi and P. M. B. Vitanyi. The google similarity distance. *IEEE Trans. on Knowl. and Data Eng.*, 19:370–383, March 2007.

[6] S. Cucerzan. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of EMNLP-CoNLL*, pages 708–716, 2007.

[7] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin. Entity disambiguation for knowledge base population. In *Proceedings of COLING*, pages 277–285, 2010.

[8] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[9] T. Hasegawa, S. Sekine, and R. Grishman. Discovering relations among named entities from large corpora. In *Proceedings of ACL*, pages 415–422, 2004.

[10] L. Jiang, J. Wang, N. An, S. Wang, J. Zhan, and L. Li. Grape: A graph-based framework for disambiguating people appearances in web search. In *Proceedings of ICDM*, pages 199–208, 2009.

[11] N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of ACL*, 2004.

[12] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of Wikipedia entities in web text. In *Proceedings of SIGKDD*, pages 457–466, 2009.

[13] D. Lin. An information-theoretic definition of similarity. In *Proceedings of ICML*, pages 296–304, 1998.

[14] B. Malin. Unsupervised name disambiguation via social network similarity. In *Proceedings of Workshop on Link Analysis, Counterterrorism, and Security*, 2005.

[15] B. Malin, E. Airoldi, and K. M. Carley. A network analysis model for disambiguation of names in lists. *Comput. Math. Organ. Theory*, 11:119–139, July 2005.

[16] G. S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of CONLL*, pages 33–40, 2003.

[17] P. McNamee, H. Simpson, and H. T. Dang. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, 2009.

[18] R. Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 411–418, 2005.

[19] R. Mihalcea and A. Csomai. Wikify! linking documents to encyclopedic knowledge. In *Proceedings of CIKM*, pages 233–242, 2007.

[20] D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceedings of WIKIAI*, 2008.

[21] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceeding of CIKM*, pages 509–518, 2008.

[22] E. Minkov, W. W. Cohen, and A. Y. Ng. Contextual search and name disambiguation in email using graphs. In *Proceedings of SIGIR*, pages 27–34, 2006.

[23] R. Navigli and M. Lapata. Graph connectivity measures for unsupervised word sense disambiguation. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1683–1688, 2007.

[24] R. Navigli and P. Velardi. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1075–1086, 2005.

[25] T. Pedersen, A. Purandare, and A. Kulkarni. Name Discrimination by Clustering Similar Contexts. In *Proceedings of CICLing*, pages 226–237, 2005.

[26] F. Suchanek, G. Kasneci, and G. Weikum. Yago: A Large Ontology from Wikipedia and WordNet. *Journal of Web Semantics*, 6(3):203–217, 2008.

[27] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge unifying wordnet and wikipedia. In *Proceedings of WWW*, pages 697–706, 2007.

[28] F. Wu and D. S. Weld. Autonomously semantifying wikipedia. In *Proceedings of CIKM*, pages 41–50, 2007.

[29] F. Wu and D. S. Weld. Automatically refining the wikipedia infobox ontology. In *Proceedings of WWW*, pages 635–644, 2008.

[30] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106, 2003.

[31] T. Zesch, I. Gurevych, and M. Mühlhäuser. Analyzing and Accessing Wikipedia as a Lexical Semantic Resource. In *Biannual Conference of the Society for Computational Linguistics and Language Technology*, 2007.