NAMED ENTITY DISAMBIGUATION: A HYBRID APPROACH

Hien T. Nguyen
Ton Duc Thang University, Viet Nam
E-mail: hien@tdt.edu.vn

Tru H. Cao
Ho Chi Minh City University of Technology, Viet Nam
E-mail: tru@cse.hcmut.edu.vn

Received 2 January 2012

Accepted 27 July 2012

Abstract

Semantic annotation of named entities for enriching unstructured content is a critical step in development of Semantic Web and many Natural Language Processing applications. To this end, this paper addresses the named entity disambiguation problem that aims at detecting entity mentions in a text and then linking them to entries in a knowledge base. In this paper, we propose a hybrid method, combining heuristics and statistics, for named entity disambiguation. The novelty is that the disambiguation process is incremental and includes several rounds that filter the candidate referents, by exploiting previously identified entities and extending the text by those entity attributes every time they are successfully resolved in a round. Experiments are conducted to evaluate and show the advantages of the proposed method. The experiment results show that our approach achieves high accuracy and can be used to construct a robust entity disambiguation system.

Keywords: *Entity disambiguation. Entity linking. Named entity. Knowledge base. Wikipedia.*

1. Introduction

In Information Extraction (IE) and Natural Language Processing (NLP) areas, named entities (NE) are people, organizations, locations, and others that are referred to by proper names. Having been raised from research in those areas, named entities have also become key issue in development of the Semantic Web [37]. That is because, in many domains, in particular news articles, the information and semantics of the article texts center around the named entities and their relations mentioned therein. In 2001, Berners-Lee *et al*. [37] described the evolution of a Web of documents for human to read to a Web of data where information is given well-defined meaning for computers to manipulate. The Semantic Web is an extension of the current Web that adds new data and metadata to existing Web documents so that computers can automatically integrate and re-use data across various applications. In that spirit, extracting named entities in texts and adding semantics, metadata about those entities in the texts themselves with supporting of some ontologies or knowledge bases (KB) such as KIM [45], Wikipedia[a], etc. have been increasingly attracting researchers' attention.

For the past decade, Named Entity Recognition (NER) has become an interesting topic, attracting much research effort, with various approaches introduced for different domains, scopes, and purposes [35, 36, 38, 39]. Some work on NER address the task of classification of NEs into broad categories such as Person, Organization, or Location [34, 36, 38], while others classify NEs into more fine-grained categories that are specified by a given ontology [35, 39]. In recent years, some well-known systems such as SemTag [46] and KIM have been attempted in not only fine-grained categorization but also identification of NEs with respect to a given ontology.

One great challenge in dealing with named entities is that one name may refer to different entities in differ-

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

Published by Atlantis Press

ent occurrences and one entity may have different names that may be written in different ways and with spelling errors. For example, the name "John McCarthy" in different occurrences may refer to different NEs such as a computer scientist from Stanford University, a linguist from University of Massachusetts Amherst, an Australian ambassador, a British journalist who was kidnapped by Iranian terrorists in Lebanon in April 1986, etc. Such ambiguity makes identification of NEs more difficult and raises NE disambiguation problem (NED) as one of the main challenges to research not only in the Semantic Web but also in areas of natural language processing in general.

Indeed, for the past five years, many approaches have been proposed for NED [1-23, 27, 28]. And, since 2009, Entity Linking (EL) shared task held at Text Analysis Conference (TAC) [1, 9] has attracted more and more attentions in linking entity mentions to knowledge base entries [1, 3, 4, 6, 7, 8, 9, 12, 15]. In EL task, given a query consists of a named entity (PER, ORG, or geo-graphical entity) and a background document containing that named entity, the system is required to provide the ID of the KB entry describing that named entity; or NIL if there is no such KB entry [9]. The used KB is Wikipedia. Even though those approaches to EL exploited diverse features and employed many learning models [1, 8, 9, 12, 15], a hybrid approach that combines rules and statistics have not been proposed.

In this paper, we present our work that aims at detecting named entities in a text, disambiguating and linking them to the right ones in Wikipedia. The proposed method is rule-based and statistical-based. It utilizes NEs and related terms co-occurring with the target entity in a text and Wikipedia for disambiguation because the intuition is that these respectively convey its relationship and attributes. For example, suppose that in a KB there are two entities named "Jim Clark", one of which has a relation with the Formula One car racing championship and the other with Netscape. Then, if in a text where the name appears there are occurrences of Netscape or web-related referents and terms, then it is more likely that the name refers to the one with Netscape in the KB.

The contribution of this paper is three-fold. First, we propose a hybrid method that combines heuristics and a learning model for disambiguation and identification of NEs in a text with respect to Wikipedia. Second, the proposed disambiguation process is iterative and incremental, each round of which exploits the previously identified entities and extends the text by the attributes of those identified entities in order to disambiguate the remaining named entities. Third, our method makes use of *disambiguation texts* in article titles of Wikipedia as an important feature for resolving the right entities for some mentions in a text, and then the identifiers of those entities are exploited as anchors to disambiguate the others. Note that this work is based on [21], [22], and [23].

The rest of the paper is organized as follows. Section 2 presents Wikipedia and related works. Section 3 presents in details the disambiguation method. Section 4 presents experiments and evaluation. Finally, we draw a conclusion in Section 5. Note that in the rest of this paper we use *mention* in the sense that is a reference to an entity. An entity of a reference is called *referent*. Therefore, we use the terms *name* and *mention* interchangeably, as well as for the terms *entity* and *referent*.

2. Background

NED can be considered as an importantly special case of Word Sense Disambiguation (WSD) [26]. The aim of WSD is to identify which sense of a word is used in a given context when several possible senses of that word exist. In WSD, words to be disambiguated may appear in either a plain text or an existing knowledge base. Techniques for the latter use a dictionary, thesaurus, or an ontology as a sense inventory that defines possible senses of words. Having been emerging recently as the largest and widely-used encyclopedia in existence, Wikipedia is used as a knowledge source for not only WSD [25], but also IE, NLP, Ontology Building, Information Retrieval, and so on [24].

This paper proposes a method that also makes use of available knowledge sources of entities for NED besides exploiting the context of a text where mentions of named entities occur. Exploiting the external source of knowledge for NED is natural and reasonable as the same as the way humans do. Indeed, when we ask a person to identify which entities a name in a text refers to, he may rely on his knowledge accumulated from diverse sources of knowledge, experiences, etc.

In literature, the knowledge sources used for NED can be divided into two kinds: close ontologies and open ontologies. Close ontologies are built by experts following a top-down approach, with a hierarchy of concepts based on a controlled vocabulary and strict constraints, e.g., KIM, WordNet. These knowledge sources are generally of high reliability, but their size and coverage are restricted. Furthermore, not only is the building of the sources labor-intensive and costly, but also they are not kept updated of new discoveries and topics that arise daily. Meanwhile, open ontologies are built by collaborations of volunteers following a bottom-up approach, with concepts formed by a free vocabulary and community agreements, e.g. Wikipedia. Many open ontologies are fast growth with wide coverage of diverse topics and keeping update daily by volunteers, but someone has doubt about quality of their information contents. Wikipedia is considered as an open ontology where contents of its articles have high quality. Indeed, in [47], Giles investigated the accuracy of content of articles in Wikipedia in comparison to those of articles in Encyclopedia Britannica, and showed that both sources were equally prone to significant errors.

2.1. *Wikipedia*

Wikipedia is a free encyclopedia written by a collaborative effort of a large number of volunteer contributors. We describe here some of its resources of information for disambiguation. A basic entry in Wikipedia is a *page* (or *article*) that defines and describes a single entity or concept. It is uniquely identified by its title. When the name is ambiguous, the title may contain further information that we call *disambiguation text* to distinguish the entity described from others. The disambiguation text is separated from the name by parentheses e.g. `John McCarthy (computer scientist)`, or a comma, e.g., `Columbia, South Carolina`.

In Wikipedia, every entity page is associated with one or more categories, each of which can have subcategories expressing meronymic or hyponymic relations. Each page may have several incoming links (henceforth *inlinks*), outgoing links (henceforth *outlinks*), and *redirect* pages. A redirect page typically contains only a reference to an entity or a concept page. Title of the redirect page is an alternative name of that entity or concept. For example, from redirect pages of the United States, we extract alternative names of the United States such as “US”, “USA”, “United States of America”, etc. Other resources are disambiguation pages. They are created for ambiguous names, each of which denotes two or more entities in Wikipedia. Based on disambiguation pages one may detect all entities that have the same name in Wikipedia.

Note that when searching an entity by its name using the search tool of Wikipedia, if this name occurs in Wikipedia, it appears that Wikipedia ranks pages whose titles contain the name, and returns either the most relevant entity page or the disambiguation page for that name. For those cases when the returned page describes an entity, we set this entity as the *default referent* for that name. For example, when one queries “Oxford” from Wikipedia, it returns the page that describes the city Oxford in South East England. Therefore, in this case, for the name “Oxford”, we set its default referent the city `Oxford` in South East England. For another example, when one queries “John McCarthy” from Wikipedia, the disambiguation page of the name “John McCarthy” is returned. In the case of “John McCarthy”, we do not set any default referent for this name.

2.2. *Related Problems*

In this section, we review related works on Entity Disambiguation. We are interested in locating in a KB the entity that a name in a text refers to. However, we start out by summarizing work on Record Linkage, which aims at detecting records intra- or inter-database or file that refer to the same entity, and then links or merges them together. We then describe and summarize work on Cross-Document Co-reference Resolution, which aims at grouping mentions of entities in different documents into equivalence classes by determining whether any two mentions refer to the same entity. Next, we focus on both simplified cases of NED that are Toponym Resolution and Person Disambiguation. Finally, we survey disambiguation solutions for NED.

Record Linkage

Record Linkage (RL) is a means of combining information from different sources such as databases or structured files in general. It has been known for more than five decades across research communities (i.e. AI and databases) with multiple names such as *entity matching* [51], *entity resolution* [53], *duplicate detection* [54], *name disambiguation* [56, 57], etc. The basic method to RL is to compare values of fields to identify whether any pair of records associated with the same entity. NED is different from RL in that it analyses free texts to capture entity mentions and then link them to KB entries other than link entity mentions from structured data sources.

A typical method proposed for RL involves two main phases, namely *data preparation* and *matching* [52]. The former is to improve the data quality, as well as make them comparable and more usable such as transforming those data from different sources into a common form or standardizing the information represented in certain fields to a specific content format. The latter is to match records to identify whether they refer to the same real-world entity. Conventional matching approaches to RL focused on discovering independent pair-wise matches of records using a variety of attribute-similarity measures such as [54]. State-of-the-art matching methods are collective matches [51, 53, 55] that rely on sophisticated machine learning model such as Latent Dirichlet Allocation topic model or Markov Logic Networks.

Cross-Document Co-reference Resolution

Cross-Document Co-reference Resolution (CDC) aims at grouping mentions of entities across documents into clusters, each of which consists of mentions that refer to the same real-world entity, rather than identifying what actual entities are. Most approaches to this problem use clustering techniques. This paper addresses the NED problem that aims at locating in a KB the entity that a mention in a document refers to. NED is different from CDC in that it does a further step that links each mention in a document to a KB entry. If ignoring this step, one can consider NED as CDC. Motivated from finding information about persons on World Wide Web, Web People task, emerged as a challenge topic and attracted attention of researchers recently years, is a simplified case of CDC [44].

A typical solution to CDC usually contains three basic steps: (i) exploiting textual contexts where mentions of entities occur to extract contextual features for creating the profiles of those entities; (ii) then, calculating the similarity between profiles using similarity metrics; (iii) and finally, applying clustering algorithms to group mentions of the same entities together. The profiles contain a mixture of collocation and other information that may denote attributes (personal information) and relations of those entities.

In general, two main types of information that often used for CDC are personal and relational information [43]. Personal information gives biographical information about each entity such as birthday, career, occupation, alias and so on. Relational information specifies relations between entities such as the membership relation between Barack Obama and the Democratic Party of the United States. The relational information can be expressed explicit or implicit in documents. The explicitly relational information of an entity may be captured by exploiting the local contexts where the mentions occur, whereas the implicitly relational information is far away the local ones.

In particular, some solutions to CDC exploit features, which denote attributes of target entities to be disambiguated, in local contexts such as token features [40, 50], bigrams [42], biographical information [48], or co-occurrence NE phrases and NE relationships [50]. Whereas others try to extract information related to NEs in consideration beyond local contexts [41, 43, 49]. After that, clustering algorithms are employed to cluster mentions of the same entities based some similarity metric such as cosine, gain ratio, likelihood ratio, Kullback-Leibler Divergence, etc. In general, the most popular clustering algorithm used by those methods is the Hierarchical Agglomerative Clustering (HAC) algorithm, although the choice of linkage varies such as single-link or complete-link, etc.

When applying clustering techniques to group mentions of entities together, since the number of clusters is not known in advance, cluster-stopping criteria is a challenge issue. To deal with this issue in cases when using the techniques like HAC, the number of clusters in the output is determined by a fixed similarity threshold. Besides HAC, some works employ other models such as classifiers in [49].

Toponym Resolution

Toponym Resolution (TR) is a task of identifying whether an entity mention refer to a place and mapping it to a geographic latitude/longitude footprint or a unique identifier in a KB. A conventional approach to TR typically involves two main sub-tasks: place name extraction and place name disambiguation. The former is to identify geographical mentions in a text. The latter firstly looks up candidate referents of a mention from an external source such as a constructed gazetteer or a particular ontology; then disambiguates it by examining the context where the mention appears to choose the most contextually similar candidate referent as the right one.

In literature, many methods are proposed to TR, most of which fit into the rule-based and machine learning methods. A completely survey of rule-based methods are in [32]. Machine learning methods employed for TR consist of bootstrapping learning [30], unsupervised learning [31], or supervised learning [29].

In summary, although various methods have been introduced since 1999, an important issue of TR is that those methods are usually evaluated in different corpora, under different conditions. The shortcoming of the methods proposed to TR is that it omits relationships between named entities with different classes, such as between persons and organizations, or organizations and locations, etc. Therefore, they are not suitable to NED where entities belong to different types.

2.3. *Related Work*

Many approaches have proposed for NED. All of them can fit into three disambiguating strategies: local, global, and collective. Local methods disambiguate each mention independently based on local context compatibility between the mention and its candidate entities using some contextual features. Global and collective methods assume that disambiguation decisions are interdependence and there is coherence between co-occurrence entities in a text, enabling the use of measures of semantic relatedness for disambiguation. While collective methods simultaneously perform disambiguation decisions, global methods in turn disambiguate each mention.

Local approaches

A typical local approach to NED focused on local context compatibility between a mention and its candidate entities. Firstly, contextual features of entities were extracted from their text descriptions. Then those extracted features were weighted and represented in a vector model. Finally, each mention in a text was linked to the candidate entity having the highest contextual similarity with it. Bunescu and Paşca [19] proposed a method that uses an SVM kernel to compare the lexical context around the ambiguous mention to that of its candidate entities, in combination with estimating correlation of the contextual word with the categories of the candidate entities. Each candidate entity is a Wikipedia article and its lexical context is the content of the article. Mihalcea and Csomai [27] implemented and evaluated two different disambiguation algorithms. The first one based on the measure of contextual overlap between the local context of the ambiguous mention and the contents of candidate Wikipedia articles to identify the most likely candidate entity. The second one trains a Naïve Bayes classifier for each ambiguous mention using three words to the left and the right of outlinks in Wikipedia articles, with their parts-of-speech, as contextual features. Zhang *et al.* [13] employed classification algorithms to learn context compatibility for disambiguation. Zheng *et al.* [14], Dredze *et al.* [15] and Zhou *et al.* [16] employed learning-to-rank techniques to rank all candidate entities and link the mention to the most likely one. Zhang *et al.* [7, 8] improve their approach in [13] by a learning model for automatically generating a very-large training set and training a statistical classifier to detect name variants. The main drawback of the local approaches is that they do not take into account the interdependence between disambiguation decisions. Han and Sun [6] proposed a generative probabilistic model that combines three evidences: the distribution of entities in document, the distribution of possible names of a specific entity, and the distribution of possible contexts of a specific entity.

Global approaches

Global approaches assumed interdependence between disambiguation decisions and exploited two main kinds of information that are disambiguation context and semantic relatedness. Cucerzan [20] was the first to model interdependence among disambiguation decisions. In [20] disambiguation context are all Wikipedia contexts that occur in the text and semantic relatedness is based on overlap in categories of entities that may be referred to in the text. Wikipedia contexts are comprised of inlink labels, outlink labels, and appositives in titles of all Wikipedia articles.

Milne and Witten [28] proposed a learning-based method that ranks each candidate based on three factors: the candidate's semantic relatedness to contextual entities, the candidate's commonness - defined as the number of times it is used as a destination in Wikipedia, and a measure of overall quality of contextual entities. A contextual entity is identified based on a disambiguation context, which is the set of unambiguous mentions having only one candidate in Wikipedia. Guo *et al.* [4] built a directed graph $G = (E, V)$, where V contains name mentions and all of their candidates. Each edge connects from an entity to a mention or vice versa; and, there is not any edge connecting two mentions or two entities. Then the approach ranks candidates of a certain mention based on their in-degree and out-degree. Hachey *et al.* [5] firstly built a seed graph $G = (E, V)$ where V contains candidates of all unambiguously mentions. The graph was then expanded by traversing length-limited paths via links in both entity and category pages in Wikipedia, and adding nodes as well as establishing edges as required. Finally, the approach ranks candidate entities using cosine and degree centrality. Ratinov *et al.* [10] proposed an approach that combines both local and global approaches by extending methods proposed in [19] and [28]. Kataria *et al.* [11] proposed a weakly semi-supervised LDA to model correlations among words and among topics for disambiguation.

Collective approaches

Kulkarni et al. [17] proposed the first collective entity disambiguation approach that can simultaneously link entity mentions in a text to corresponding KB entries and introduced the collective optimization problem to this end. The approach combines local compatibility between mentions and their candidate entities and semantic relatedness between entities. Since jointly optimization of overall linking is NP-hard, the authors proposed two approximation solutions to resolve it. Kbleb and Abecker [18] proposed an approach that exploits an RDF(s)-graph structure and co-occurrence among entities in a text for disambiguation. The approach applies Spreading Activation method to rank and generate the most optimal Steiner graph based on activation values. The result graph contains KB entities that actually are referred to in the text.

Some research works [2, 3] built a referent graph for a text and proposed a collective inference method to entity disambiguation. A referent graph is a weighted and undirected graph $G = (E, V)$ where V contains all mentions in the text and all possible candidates of these mentions. Each node represents a mention or an entity. The graph has two kinds of edges:

- A mention-entity edge is established between a mention and an entity, and weighted based on context similarity, or a combination of popularity and context similarity;
- An entity-entity edge is established between two entities and weighted using semantic relatedness between them.

Based on a referent graph, one can proposed a method that performs collective inference KB entities referred to in a text. Han and Sun [3] and Hoffart *et al.* [2] proposed approaches that exploit local context compatibility and coherence among entities to build a referent graph and then proposed a collective reference based on the graph in combination with popularity measures of mentions or entities for simultaneously identifying KB entries of all mentions in the text. Note that exploiting the popularity of mentions is based on a popular assumption that some mentions or entities in a text are more important than others, which was used in previous work [27, 28].

Hoffart *et al.* [2] proposed a method for collective disambiguation based on a close ontology - YAGO ontology. The authors calculated the weight of each mention-entity edge based on popularity of entities and context similarity, which is comprised of keyphrase-based and syntax-based similarity; calculated the weight of each entity-entity edge based on Wikipedia-inlinks overlap between entities. Then they proposed a graph-based algorithm to find a dense-subgraph, which is a graph where each mention node has only one edge connecting it with an entity.

Han and Sun [3] firstly built a referent graph where the local context compatibility was calculated base on a bag-of-words model as in [19] and semantic relatedness was adopted the formula presented in [28]. Second, the authors proposed a collective algorithm for disambiguation. The collective algorithm collects initial evidence for each mention and then reinforces the evidence by propagating them via edges of the referent graph. The initial evidence of each mention shows its popularity over the other mentions and its value is TF-IDF score normalized by the sum over TF-IDF scores of all mentions in the text.

In our method, we exploit not only tokens around mentions, but also their co-occurring named entities in a text. Especially, for those named entities that are already disambiguated, we use their identifiers, which are more informative and precise than entity names, as essential disambiguation features of co-occurring mentions. We also introduce a rule-based method and combine it with a statistical one. The experimental results show that the rule-based phase enhances the disambiguation precision and recall significantly. Both of the statistical and rule-based phases in our algorithm are iterative, exploiting the identifiers of the resolved named entities in a round for disambiguation of the remaining mentions in the next round.

In fact, the incremental mechanism of our method is similar to the way humans do when disambiguating mentions based on previously known ones. That is, the proposed method exploits both the flow of information as it progresses in a news article and the way humans read and understand what entities that the mentions in the news article refers to. Indeed, an entity occurring first in a news article is usually introduced in an unambiguous way, except when it occurs in the headline of the news article. Like humans, our method disambiguates named entities in a text in turn from the top to the bottom of the text. When the referent of a mention in a text is identified, it is considered as an anchor and its identifier and own features are used to disambiguate others. Also, when encountering an ambiguous mention in a text, a reader usually links it to the previously resolved named entities and his/her background knowledge to identify what entity that mention refers to. Similarly, our method exploits the coreference chain of mentions in a text and information from an encyclopedic knowledge base like Wikipedia for resolving ambiguous mentions. Furthermore, both humans and our method explore contexts in several levels, from a local one to the whole text, where diverse clues are used for the disambiguation task.

3. Proposed method

In a news article, co-occurring entities are usually related to the same context. Furthermore, the identity of a named entity is inferable from nearby and previously identified NEs in the text. For example, when the name "Georgia" occurs with "Atlanta" in a text and "Atlanta" is already recognized as a city in the United States, it is more likely that "Georgia" refers to a state of the United States than the country Georgia. Meanwhile, if "Georgia" occurs with "Tbilisi" capital as in the text "*TBILISI (CNN) -- Most Russian troops have withdrawn from eastern and western Georgia*", it is "Tbilisi" that helps to identify "Georgia" referring to the country next to Russia. In addition, the words surrounding ambiguous mentions may denote attributes of the NEs they refer to. If those words are automatically recognized, the ambiguous mentions may be disambiguated. For example, in the text "*John McCarthy, an American computer scientist pioneer and inventor, was known as the father of Artificial Intelligence (AI)*", the word "computer scientist" can help to discriminate John McCarthy who invented the Lisp programming language from other ones.

When analyzing the structure of news articles, we observe that when first referring to a named entity, except in the headline, journalists usually either implicit or explicit introduce it in an unambiguous way by using its main alias or giving more information for readers to understand clearly about the entity they mean. For instance, in the news article with the headline "*U.S. on Palestinian government: Hamas is sticking point*" on CNN (March 04, 2009) has the lead "*JERUSALEM (CNN) -- U.S. Secretary of State Hillary Clinton on Tuesday ruled out working with any Palestinian unity government that includes Hamas if Hamas does not agree to recognize Israel*" in which the journalist refers to the wife of the 42nd President of the United States clearly by the phrase "U.S. Secretary of State Hillary Clinton". Then in the body of the story, s/he writes "*Clinton said Hamas must do what the Palestine Liberation Organization has done*" where "Clinton" mentions the Hillary Clinton without introducing more information to differentiate with the former president Bill Clinton of the United States. Especially, for a well-known location entity, although its name may be ambiguous, a journalist can still leave the name alone. However, for other cases, s/he may clarify an ambiguous location name by mentioning some related locations in the text. For instance, when using "Oxford" to refer to a city in Mississippi of the United States, a journalist may write "Oxford, Mississippi" whereas, when using this name to refer to the well-known city Oxford in South East England, s/he may just write "Oxford".

From those observations, we propose a method with the following essential points. Firstly, it is a hybrid method containing two phases. The first phase is a rule-based phase that filters candidates and, if possible, it disambiguates named entities with high reliability. The second phase employs a statistical learning model to rank the candidates of each remaining mention and choose the one with the highest ranking as the right referent of that mention. Secondly, each phase is an iterative and incremental process that makes use of the identifiers of the previously resolved named entities to disambiguate others. Finally, it exploits both entity identifiers and keywords for named entity disambiguation in two phases. The specific steps in the two phases of our disambiguation process are presented below.

- **Step 1:** identifies if there exist entities in Wikipedia that a mention in a text may refer to and then retrieves those entities as candidate referents of the mention.
- **Step 2:** applies some heuristics to filter candidates of each mention and, if possible, choose the right one for the mention. The earlier a mention is resolved in this step, the more reliable the identified entity is. As a result, when an entity in Wikipedia is identified as the actual entity that a mention in a text refer to, its identifier will be considered as an anchor that the method exploits to resolve others.

- **Step 3:** employs the vector space model in which the cosine similarity is used as a scoring function to ranks the candidates of the mention and chooses the one with the highest score as the right entity that the mention refers to.

As mentioned above, the disambiguation process involves two stages. The first stage is rule-based and includes Step 1 and Step 2. The second stage is statistical and includes Step 3.

3.1. *Heuristic*

In this section, we propose some heuristics used in the first stage and based on local contexts of mentions to identify their correct referents. The local context of a location mention is its preceding and succeeding mentions in the text. For example, if "Paris" is a location mention and followed by "France", then the country France is in the local context of this "Paris". The local context of a person or an organization mention comprises the keywords and unambiguous mentions occurring in the same sentence where the mention occurs. We exploit such a local context of a mention to narrow down its candidates and disambiguate its referents if possible, using the following heuristics in the sequence as listed.

H_1. Disambiguation text following

For a location mention, its right referent is the candidate whose disambiguation text is identical to the succeeding mention. For example, in the text *"Columbia, South Carolina"*, for the mention "Columbia", the candidate Columbia, South Carolina, the largest city of South Carolina, in Wikipedia is chosen because the disambiguation text of the candidate is "South Carolina" and identical to the succeeding mention of "Columbia".

H_2. Next to disambiguation text

For a location mention, its right referent is the candidate whose name is identical to the disambiguation text of the referent of its preceding unambiguous mention. For example, in the text *"Atlanta, Georgia"*, assuming that the referent of "Atlanta" has already been resolved as Atlanta, Georgia, a major city of state Georgia of United States. Then, for the mention "Georgia", the candidate Georgia (U.S. state) is chosen because the referent of its preceding mention "Atlanta" is Atlanta, Georgia whose disambiguation text is identical to "Georgia".

H_3. Disambiguation text in the same window

For a person or an organization mention, the chosen candidate referent is the one whose disambiguation text occurs in the local context of that mention, or the local contexts of the mentions in its coreference chain. After this step, if there is only one candidate in the result, the referent is considered being resolved. For example, in the text "*Veteran referee (Big) John McCarthy, one of the most recognizable faces of mixed martial arts*", the word "referee" helps to choose the candidate John McCarthy (referee) as the right one instead of John McCarthy (computer scientist) or John McCarthy (linguist) in Wikipedia.

To show more detail about the way that our method exploits the local contexts in the coreference chain of a mention, we describe here the example "*Sen. John McCain said Monday that Rep. John Lewis controversial remarks were "so disturbing" that they "stopped me in my tracks." [...] Lewis, a Georgia representative and veteran of the civil rights movement, on Saturday compared the feeling at recent Republican rallies to those of segregationist George Wallace.*" In this example, "John Lewis" and "Lewis" are actually co-referent and, in the local context of the mention "Lewis", there occurs the word "Georgia" that is the disambiguation text of the entity John Lewis (Georgia) in Wikipedia. Therefore, in this context, after applying heuristic H_3, our method identifies both mentions "John Lewis" and "Lewis" refer to the same entity John Lewis(Georgia) in Wikipedia.

H_4. Coreference relation

For each coreference chain, we propagate the resolved referent of a mention in it to others. For example, assume that in a text there are occurrences of coreferent mentions "Denny Hillis" and "Hillis", where "Hillis" may refer to Ali Hillis, American actress, Horace Hillis, American politician, or W. Daniel Hillis, American inventor. If "Denny Hillis" is recognized as referring to W. Daniel Hillis in Wikipedia, then "Hillis" also refers to W. Daniel Hillis. As another example, for the text *"About three-quarters of white, college-educated men age over 65 use the Internet, says Susannah Fox, [...] John McCain is an outlier when you compare him to his peers, Fox says.*", there are 164 entities in the Wikipedia version used with the same name "Fox". However, "Susannah Fox" does not exist in Wikipedia yet and is coreferent with "Fox" in the text, so our method recognizes "Fox" as referring to an out-of-Wikipedia entity.

We note that a coreference chain might not be correctly constructed in the pre-processing steps due to the employed NE coreference resolution module. Moreover, for a correct coreference chain, if there is more than one mention already resolved, then it does matter to choose the right one to be propagated. Therefore, for a high reliability, before propagating the referent of a mention that has already been resolved to other mentions in its coreference chain, our method checks whether that mention satisfies one of the following criteria:

(i) The mention occurs in the text prior to all the others in its coreference chain and is one of the longest mentions in its coreference chain (except for those mentions occurring in the headline of the text), or

(ii) The mention occurs in the text prior to all the others in its coreference chain and is the main alias of the corresponding referent in Wikipedia (except for those mentions occurring in the headline of the text). A mention is considered as the main alias of a referent if it occurs in the title of the entity page that describes the corresponding entity in Wikipedia. For example, "United States" is the main alias of the referent the `United States` because it is the title of the entity page describing the United States.

H_5. Default referents

After applying all the above heuristics, for location mentions that have not been resolved yet, our method chooses its default referent as the right one. For instance, in the context, "*McCain's willingness to disassociate himself with Bush is not a new strategy. The two men are not close and right now McCain is fighting for the support of undecided, independent voters in states such as Pennsylvania, Ohio and Florida.*", `Pennsylvania`, `Ohio`, `Florida` state of the United States in Wikipedia are chosen because these entities respectively are default referents of those underlined mentions.

3.2. *Statistical Ranking Model*

To maximize accuracy of mapping NEs referred to in a text to the right ones in a given KB poses a significant question that how contexts in which the mentions of the NEs occur are exploited and how the corresponding NEs in the KB can be represented. In our case, we represent NEs in the KB by their attributes and relations. For NEs referred to in a text, we extract those features that likely represent their attributes and relations in contexts where those NEs occur. The attributes are birthday, career, occupation, alias, first name, last name, and so on. The relations of an entity represent its relations to others such as part-of, located-in, for instances. The way we exploit a context is based on Harris' Distributional Hypothesis [58] stating that words occurring in similar contexts tend to have similar senses. We adapt that hypothesis to NE instead of word sense disambiguation. After exploring meaningful features for representing NEs in texts and a KB, our method assigns each NE referred to in a text to the most contextually similar referent in the KB.

In this section, we present a statistical ranking model where we employ the Vector Space Model (VSM) to represent entity mentions in a text and entities in Wikipedia by their features. The VSM considers the set of features of entities as a bag-of-words. Firstly, we present what contextual features are extracted and how we normalize them. Then we present how to weight words in the VSM and calculate the similarity between feature vectors of mentions and entities. Based on the calculated similarity, our disambiguation method ranks the candidate entities of each mention and chooses the best one. The quality of ranking depends on used features.

Text features

To construct the feature vector of a mention in a text, we extract all mentions co-occurring with it in the whole text, local words in a context window, and words in the context windows of those mentions that are coreferent with the mention to be disambiguated. Those features are presented below.

- *Entity mentions* (EM). After named entity recognition, mentions referring to named entities are detected. We extract these mentions in the whole text. After extracting the mentions, for the ones that are identical, we keep only one and remove the others. For instance, if "U.S" occurs twice in a text, we remove one.
- *Local words* (LW). All the words found inside a specified context window around the mention to be disambiguated. The window size is set to 55 words, not including special tokens such as $, #, ?, etc., which is the value that was observed to give opti-

mum performance in the related task of cross-document coreference resolution [40]. Then we remove those local words that are part of mentions occurring in the window context to avoid extracting duplicate features.

- *Coreferential words* (CW). All the words found inside the context windows around those mentions that are co-referent with the mention to be disambiguated in the text. For instance, if "John McCarthy" and "McCarthy" co-occur in the same text and are co-referent, we extract words not only around "John McCarthy" but also those around "McCarthy". The size of those context windows are also set to 55 words. Note that, when the context windows of mentions that are co-referent are overlapped, the words in the overlapped areas are extracted only once. We also remove those extracted words that are part of mentions occurring in the context windows to avoid extracting duplicate features.

Wikipedia features

For each entity in Wikipedia, serving as a candidate entity for an ambiguous mention in a text, we extract the following information to construct its feature vector.

- *Entity title* (ET). Each entity in Wikipedia has a title. For instance, "John McCarthy (computer scientist)" is the title of the page describing Prof. John McCarthy who is the inventor of Lisp programming language. We extract "John McCarthy (computer scientist)" for the corresponding entity.
- *Titles of redirect pages* (RT). Each entity in Wikipedia may have some redirect pages whose titles contain different names, i.e. aliases, of that entity. To illustrate, from the redirect pages of an entity John Williams in Wikipedia, we extract their titles: Williams, John Towner; Johnny Williams; Williams, John; John Williams (composer); etc.
- *Category labels* (CAT). Each entity in Wikipedia belongs to one or more categories. We extract labels of all its categories. For instance, from the categories of the entity `John McCarthy (computer scientist)` in Wikipedia, we extract the following category labels as follows: Turing Award laureates; Computer pioneers; Stanford University faculty; Lisp programming language; Artificial intelligence researchers; etc.
- *Outlink labels* (OL). In the page describing an entity in Wikipedia there are some links pointing to other Wikipedia entities. We extract labels (anchor texts) of those outlinks as features of that entity.

Note that infoboxes of pages in Wikipedia are meaningful resources for disambiguation. However, these resources of information may be missed in many pages or information in many infoboxes is quite poor. Moreover, the information in infobox of each page can be distilled from the content of the page. Therefore, our disambiguation method does not extract information from infoboxes for disambiguation.

Normalization

After extracting features for a mention in a text or an entity, we put them into a 'bag of words'. Then we normalize the bag of words as follows: (i) removing special characters in some tokens such as normalizing U.S to US, D.C (in "Washington, D.C" for instance) to DC, and so on; (ii) removing punctuation mark and special tokens such as commas, periods, question mark, $, @, etc.; and (iii) removing stop words such as *a*, *an*, *the*, etc., and stemming words using Porter stemming algorithm. After normalizing the bag of words, we are already to convert it in to a token-based feature vector.

Term weighting

For a mention in a text, suppose there are N candidate entities for it in Wikipedia. We use the *tf-idf* weighting schema viewing each 'bag of words' as a document and using cosine similarity to calculate the similarity between the bag of words of the mention and the bag of words of each of the candidate entities respectively. Given two vector S_1 and S_2 for two bags of words, the similarity of the two bags of words is computed as:

$$Sim(S_1, S_2) = \sum_{\textit{common word } t_j} w_{1j} \times w_{2j} \qquad (1)$$

where t_j is a term present in both S_1 and S_2, w_{1j} is the weight of the term t_j in S_1 and w_{2j} is the weight of the term t_j in S_2.

The weight of a term t_j in vector S_i is given by:

$$w_{ij} = log(tf_j+1) \times log(N/df_j)/\sqrt{s_{i1}^2 + s_{i2}^2 + ... + s_{iN}^2} \qquad (2)$$

where tf_j is the frequency of the term t_j in vector S_i, N is the total number of candidate entities, df_j is the number of bags of words representing candidate entities in which the term t_j occurs, $s_{ij} = log(tf_j+1) \times log(N/df_j)$.

Algorithm

For a mention *m* that we want to disambiguate, let *C* be the set of its candidate entities. We cast the named entity disambiguation problem as a ranking problem with the assumption that there is an appropriate scoring function to calculate semantic similarity between feature vectors of an entity $c \in C$ and the mention *m*. We build a ranking function that takes as input the feature vectors of the entities in *C* and the feature vector of the mention *m*, then based on the scoring function to return the entity $c \in C$ with the highest score. We use *Sim* function as given in Eq.1 as the scoring function.

What we have just described is implemented in Algorithm 1. *Sim* is used at Line 5 of the algorithm. The *FVector* function in the algorithms returns the feature vector of a mention.

Algorithm 1 Statistical-Based Entity Ranking

```
1:  let C a set of candidate entities of m
2:  for each candidate c do
3:      score[c] ← Sim(FVector(c), FVector (m))
4:  end for
5:  $c^* \leftarrow \arg\max_{c_i \in C} score[c_i]$
6:  if score[c*] > τ then return c*
7:  return NIL
```

3.3. *Disambiguating process*

Prior to looking up candidates in Wikipedia, we perform some pre-processing steps. In particular, we perform NE recognition and NE coreference resolution using natural language processing resources of an Information Extraction engine based on GATE [34], a general architecture for developing natural language processing applications. The NE recognition applies pattern-matching rules written in JAPE's grammar of GATE, in order to identify the class of an entity in the text. After performing NE recognition and detecting all mentions of entities occurring in the text, we perform NE co-reference resolution using the method presented in [33] and implemented in GATE system. After these pre-processing steps, for each name in the text, we send it as a query to Wikipedia to retrieve its candidate referents. Finally, we run our disambiguating algorithm, namely Algorithm 2. Algorithm 2 takes as an input a set of mentions and return a set of mention-entity mappings. During the disambiguation process, if a mention is disambiguated, the entity corresponding with it is immediately used to disambiguate the others. The function *revised* (.) makes use of coreference relations among mentions of named entities to adjust the disambiguated results. Line 2 to Line 17 shows the first stage using the heuristics presented above and Line 18 to Line 31 shows the second stage employing the statistical ranking model for disambiguation.

Algorithm 2 Iterative and Incremental NED

```
1: let 𝒩 be a set of mentions and E be an empty set
2: E ← ∅
3: flag ← false
4: loop until 𝒩 empty or flag is true
5:     𝒩' ← 𝒩
6 :    for each n ∈ 𝒩' do
7:         C ← a set of candidate entities of n
8:         apply $H_1$, $H_2$, $H_3$ respectively for n
9:         if sizeof(C) = 1 then
10:            map n to γ* ∈ C
11:            E ← revised(E ∪ {<n → γ*>})
12:            remove n from 𝒩
13:        end if
14:    end for
15:    if E no change then flag = true
16: end loop
17: apply $H_5$
18: flag ← false
19: loop until 𝒩 empty or flag is true
20:    𝒩' ← 𝒩
21:    for each n ∈ 𝒩' do
22:        C ← a set of candidate entities of n
23:        γ* ← run Algorithm 1 for n
24:        if γ* is not NIL then
25:            map n to γ*
26:            E ← revised(E ∪ {<n → γ*>})
27:            remove n from 𝒩
28:        end if
29:    end for
30:    if E no change then flag = true
31: end loop
```

4. Experiments and evaluation

For evaluating the performance of our disambiguation method, we have built a corpus in which named entities of the types Person, Location, and Organization are manually annotated with their attributes using Wikipedia data. We first downloaded the top two or three articles in each of the eleven CNN news categories, namely, Top Stories, Politics, Entertainment, Tech, Travel, Africa, World, World Sport, World Business, Middle East, and Americas on July 22, 2008. Then we downloaded 10 articles on Oct 17, 2008 in the Top Stories category of the CNN news agency to build a dataset *D* with 40 articles for evaluation.

We divide entity names in the dataset into the four categories as follows:

- **Category 1**: names that occur in Wikipedia, and they refer to entities in Wikipedia.
- **Category 2**: names that occur in Wikipedia, but they refer to entities that are not in Wikipedia.
- **Category 3**: names that do not occur in Wikipedia, and they refer to entities that are not in Wikipedia.
- **Category 4**: names that do not occur in Wikipedia, but they refer to entities in Wikipedia.

The annotation process focuses on named entities of three types – Person, Location, and Organization. Finally, we obtain a golden standard corpus in which each named entity is annotated with the four following information:

- *TYPE*: represents the type of the named entity, which is Person, Location, or Organization.
- *ID***:** uniquely identifies the corresponding referent, if existing, in Wikipedia. If the name of the entity belongs to Category 1 or Category 4, the *ID* is the title of the corresponding referent in Wikipedia. For instance*,* if the entity name "John McCarthy" in a text actually refers to John McCarthy who is the inventor of Lisp programming language, then its *ID* is `John McCarthy (computer scientist).` Otherwise, if the name of the entity belongs to Category 2 or Category 3, then *ID* receives the *NIL* value.
- *CAT:* represents the category of the name of the entity. That is, *CAT* is either Category 2 or Category 3 when the entity name actually refers to an out-of-Wikipedia entity, or it is either Category 1 or Category 4 when the entity name refers to an entity in Wikipedia.
- *POS:* represents the position where the named entity occurs by characters. For instance, in the text "*Sen. Barack Obama says Sen. John McCain will not bring the change the country needs*", the position where "John McCain" occurs is 28.

The corpus size is 30,699 in tokens. There are totally 1,852 mentions of named entities in the corpus that refer to totally 526 distinct entities in the real-world, among which there are totally 664 distinct names. There are 1,706 mentions having the corresponding entities in Wikipedia, among which there are 967 mentions having two or more candidates, adding up to 6,885 as the total number of matched candidates. Therefore, the average number of candidates per a name in those 664 distinct names is 6885/664 = 10.36 candidates.

In more details, Table 1 shows the statistics of the named entities for each entity type in the golden standard corpus. The Column 1# represents the number of mentions for each entity type in the corpus. The Column 2# represents the number of mentions in the corpus that actually refer to entities in Wikipedia. The Column 3# represents the number of mentions in the corpus that refer to out-of-Wikipedia entities. The Column 4# represents the number of mentions that have two or more candidate referents in Wikipedia.

Table 1. Statistics of mentions in the datasets.

Entity type	#1	#2	#3	#4
Person	863	736	127	409 (out of 736)
Location	665	655	10	402 (out of 655)
Organization	324	315	9	156 (out of 315)
Total	1852	1706	146	967 (out of 1706)

To evaluate, we first define the measures to evaluate the performance of the proposed method, whose outcome is a mapping from the mentions in a text to entities in Wikipedia or to *NIL*. Table 2 defines if a mapping for a mention is correct or not, depending on the category of that mention. Specifically, for a mention of Category 1 or Category 4, which actually refers to an entity in Wikipedia, it is correct if and only if the mention is mapped to the right entity in Wikipedia. For a mention of Category 2 or Category 3, which does not refer to any entity in Wikipedia, it is correct if and only if the mention is mapped to *NIL*.

Table 2. Correct and incorrect mention-entity mappings with respect to mention categories

	Correct mapping	Incorrect mapping
Category 1	to the right entity in Wikipedia	to a wrong entity in Wikipedia or NIL
Category 2	to NIL	to an entity in Wikipedia
Category 3	to NIL	to an entity in Wikipedia
Category 4	to the right entity in Wikipedia	to a wrong entity in Wikipedia or NIL

We evaluate our method in two scenarios. In the first scenario, we use GATE 3.0[b] to detect and tag boundaries of names occurring in the dataset and then categorize corresponding referents as Person, Location and Organization. After that, we gain D_1 dataset. We found some wrong cases in D_1 as follows:

- GATE fails to detect boundaries of some names. For example, "African National Congress" is rec-

[b] http://gate.ac.uk/download/

ognized as “African National”, “Andersen Air Force Base” as “Air Force”, and “Luis Moreno-Ocampo” as “Luis Moreno-”.

- GATE detects some names (12 cases in our constructed corpus) as two different names. For example, “Omar al-Bashir” is recognized as separate names “Omar” and “al-Bashir”, “Sony Ericsson” as “Sony” and “Ericsson”.
- There are many names (145 cases) that GATE misses recognizing them. For example, “Darfur”, “Qunu”, “Soweto”, “Interfax”, “Rosoboronexport” are not recognized as entity names.
- GATE fails to identify types of named entities, e.g. “Robben Island Prison” is recognized as a person.
- GATE wrongly recognizes mentions such as Iranian, Young (meaning the young people), and Christian, as named entities.
- GATE wrongly produces some coreference chains.

Then we manually fix all such errors in the dataset D_1, obtaining the dataset D_2 with no error. Table 3 presents the statistics of mentions recognized by GATE in the dataset D_1. We note that the figures in Table 3 are not necessarily the same as those in Table 1 for the ground-truth corpus, due to GATE’s errors as pointed above.

Table 3. Statistics of mentions in the dataset D_1

Entity type	#1	#2	#3	#4
Person	794	613	180	403 (out of 613)
Location	625	597	28	373 (out of 597)
Organization	297	253	44	140 (out of 253)
Total	1716	1463	252	916 (out of 1463)

Due to the aforementioned possible error of a named entity recognition module splitting a name into two separate ones, we introduce the notion of *partially correct* mappings. That is, if a mention is correctly disambiguated but it is only part of a full name in a text, then the mapping is only partially correct. For example, if “Barack Obama” (meaning the current President of the United States) in a text is recognized as two separate mentions “Barack” and “Obama”, and the mention “Barack” is mapped to the entity `Barack Obama` (the same President) in Wikipedia, then the mapping is partially correct. A mention-entity mapping is said to be *fully* correct if the mention coincides with its full name in a text.

Let T_{all} be the number of all ground-truth mention-entity mappings in a dataset, T_C be the number of fully correct mappings, T_P be the number of partially correct mappings, and T_I be the number of incorrect mappings by a named entity recognition and disambiguation system. Each fully correct mapping is counted as one point, while each partially correct mapping is counted as only a half. Then the *precision* and *recall* of the system on the dataset are defined as follows:

- *Precision (P)*: the ratio of the number of correct mention-entity mappings and the number all returned mappings by the system.

$$P = \frac{T_C + \frac{1}{2}T_P}{T_C + \frac{1}{2}T_P + T_I} \tag{3}$$

- *Recall (R):* the ratio of the number of correct mention-entity mappings and the number of all ground-truth mappings.

$$R = \frac{T_C + \frac{1}{2}T_P}{T_{all}} \tag{4}$$

Table 4. Precision and Recall after running Algorithm 2 in the three modes on the dataset D_2

	P&R	PER	LOC	ORG	ALL
Random	*P*=*R*	52.65%	38.34%	55,75%	48,09%
Rule-based	*P*	97,48%	97,07%	93,42%	**96,78%**
	R	85,10%	89,92%	60,30%	82,42%
Statistical	*P*=*R*	89,95%	65,86%	83,03%	80,11%
Hybrid	*P*=*R*	95,38%	92,78%	87,27%	**93,01%**

In order to evaluate the affect of each phase in our proposed method, we run Algorithm 2 in three modes. The first mode, named `Rule-based`, only employs heuristics presented in Section 3.1 to disambiguate named entities, i.e., running the algorithm from Line 1 to Line 17. The second mode, named `Statistical`, only employs the vector space model for raking candidates as presented in Section 3.2, i.e., running Line 1, 2, and Line 18 to Line 31 in the algorithm. The last mode, named *Hybrid*, runs the whole algorithm. Also, for separately evaluating performance of the system with and without incurred errors of the preceding named entity recognition module, we run the three modes on both datasets D_1 and D_2. All the results are matched against the golden standard corpus.

Table 4 presents the precision and recall calculated when we randomly assign a KB entry for each entity mention in D_2 and run the Algorithm 2 on D_2 in the three modes. Since our disambiguation method maps all available mentions in an input dataset, i.e., D_2 in this case, the number of returned mappings is equal to the number of mappings in the corresponding gold standard corpus. Therefore, *P* and *R* are the same for each running mode on D_2 except in the `Rule-based` mode. When running the Algorithm 2 in the `Rule-based` mode, because there are not any heuristics that fire for some entity mentions, *P* and *R* are different.

Table 4 also shows that the proposed heuristics give high precision. So one can adopt these heuristics to improve performance of related works such as [3], [10], or [28]. Indeed, disambiguation context in [28] are the only candidate entities of *unambiguous* mentions and disambiguation context in [10] are candidate entities having highest local compatibility with context of their mentions. In our opinion, these disambiguation context are not really reliable due to low performance of disambiguation systems based on local compatibility and the fact that the only candidate entity of an unambiguous mention may not be the one to which the mention actually refer. Therefore, our proposed heuristics can produce more reliable disambiguation context than those proposed in [10] and [28]. These heuristics can also be employed to reduce the size of referent graph proposed in [3], which lead to reduce calculation cost of the collective inference algorithm.

Table 5. Precision and Recall after running Algorithm 2 in the three modes on the dataset D_1

	PER	LOC	ORG	ALL
P	76,58%	88,00%	73,06%	80,12%
R	70,85%	82,70%	66,97%	74,43%
F	73,60%	85,26%	69,88%	77,17%

Table 5 presents the precision and recall calculated when we run the Algorithm 2 on D_1. One can observe that, due to the errors of the preceding named entity recognition and coreference resolution phases by GATE, all the precision and recall measures are decreased as compared to those on D_2.

In summary, there are different sources of failures in the results. First, it is due to errors of the employed named entity recognition and coreference resolution modules, i.e., ones of GATE in this experiment. Second, it is due to the incompleteness of Wikipedia, such as shortage of entity aliases and real-world entities, and poor descriptions of some entities, which cause failures in the looking up and ranking steps. Third, it is due to our method itself. We isolated and evaluated our method on the dataset D_2 without errors from pre-processing phases and evaluated the method on D_1 with errors accumulated from pre-processing phases. The experiment results presented respectively in Table 4 and Table 5 show that our method achieves good performance.

We note that although we utilize information from Wikipedia for named entity disambiguation, our method can be adapted for an ontology or a knowledge base in general. In particular, one can generate a profile for each of KB entities by making use of ontology concepts and properties of the entities. For instance, one can extract the direct class and parent classes of an entity as ones of its features from the given hierarchy of classes. Also, values of properties of entities are exploited. For attributes, their values are directly extracted. For relation properties, one can utilize the names and identifiers of the corresponding entities. All the extracted features of an entity will be concatenated into a text snippet, which can be considered as a profile of that entity for further processing.

5. Conclusion

We have proposed a method to named entity disambiguation. It is a hybrid and incremental process that utilizes previously identified named entities and related terms co-occurring with ambiguous names in a text for the disambiguation task. Our method is robust to free texts without well-defined structures or templates. It can also be adapted for other languages using freely available various language versions of Wikipedia, as well as for any ontology and knowledge base in general. Moreover, the proposed method can map a name to an entity that is missing that name in the knowledge base of discourse. As such, it helps to discover from texts, and automatically enrich the knowledge base with, new aliases of named entities. The experiment results have shown that our method achieves good performance in terms of the precision and recall measures.

This work focuses on named entities in texts. However, general concepts also play an important role in forming the meaning of those texts. Therefore, in the future work, we will investigate new features and disambiguation methods that are suitable for both named entities and general concepts. For this line of research, we find it possible to adapt the Latent Dirichlet Allocation Category Language Model proposed in [59] for disambiguating both named entities and general concepts, in combination with the method proposed in this paper.

References

1. H. Ji, R. Grishman, and H. T. Dang, An Overview of the TAC2011 Knowledge Base Population Track, in *Proc.of Text Analysis Conference* (TAC2011).
2. Hoffart *et al.*, Robust Disambiguation of Named Entities in Text, in *Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing* (Edinburgh, Scotland, UK, July 27–31, 2011), pp. 782–792.
3. X. Han, L. Sun, and J. Zhao, Collective Entity Linking in Web Text: A Graph-Based Method, in *Proc. of the 34th Annual ACM SIGIR Conference* (Beijing, China, July 24-28, 2011), pp. 765-774.
4. Y. Guo, W. Che, T. Liu, S. Li, A Graph-based Method for Entity Linking, in *Proc. of the 5th International Joint Conference on Natural Language Processing* (IJCNLP-2011, Chiang Mai, Thailand, November 8-13, 2011), pp. 1010-1018.
5. B. Hachey, W. Radford, J. Curran, Graph-based Named Entity Linking with Wikipedia, in *Proc. of the 12th International Conference on Web Information System Engineering* (Sydney, NSW, Australia, 2011), pp. 213-226.
6. X. Han and L. Sun, A Generative Entity-Mention Model for Linking Entities with Knowledge Base, in *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Portland, Oregon, USA, June 19-24, 2011), pp. 945-954.
7. W. Zhang, J. Su, and C.-L. Tan, A Wikipedia-LDA Model for Entity Linking with Batch Size Changing Instance Selection, in *Proc. of International Joint Conference for Natural Language Processing* (IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011), pp. 562-570.
8. W. Zhang, Y. C. Sim, J. Su, and C.-L. Tan, Entity Linking with Effective Acronym Expansion, Instance Selection and Topic Modeling, in *Proc. of International Joint Conferences on Artificial Intelligence 2011* (IJCAI 2011, Barcelona, Spain, Jul 16-22, 2011), pp. 1909-1904.
9. H. Ji and R. Grishman, Knowledge Base Population Successful Approaches and Challenge, in *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Portland, Oregon, USA, June 19-24, 2011), pp. 1148-1158.
10. L. Ratinov, D. Roth, D. Downey, M. Anderson, Local and Global Algorithms for Disambiguation to Wikipedia, in *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologie*s (Portland, Oregon, USA, June 19-24, 2011), pp. 1375-1384.
11. S. Kataria, K. Kumar, R. Rastogi, P. Sen, and S. Sengamedu. Entity Disambiguation with Hierarchical Topic Models, in Proc. of *17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (KDD 2011, August 21-24, 2011, San Diego, CA), pp. 1037-1045.
12. S.Gottipati and J. Jiang, Linking Entities to a Knowledge Base with Query Expansion, in *Proc. of the 2011 Conference on Empirical Methods in Natural Language Processing* (Edinburgh, Scotland, UK, July 27–31, 2011), pp. 804-813.
13. W. Zhang, J. Su, C.-L. Tan, and W. Wang, Entity Linking Leveraging Automatically Generated Annotation, in *Proc. of 23rd International Conference on Computational Linguistics* (COLING 2010, Beijing, China, August 23-27, 2010), pp. 1290-1298.
14. Z. Zheng, F. Li, M. Huang, and X. Zhu, Learning to Link Entities with Knowledge Base, in *Human Language Technologies 2010: The Annual Conference of the North American Chapter of the Association for Computational Linguistics* (HLT/NAACL 2010).
15. M. Dredze, P. McNamee, D. Rao, A. Gerber, T. Finin, Entity Disambiguation for Knowledge Base Population, in *Proc. of 23rd International Conference on Computational Linguistics* (COLING 2010, Beijing, China, August 23-27, 2010), pp. 277-285.
16. Y. Zhou, L. Nie, O. Rouhani-Kalleh, F. Vasile, and, S. Gaffney, Resolving Surface Forms to Wikipedia Topics, in *Proc. of 23rd International Conference on Computational Linguistics* (COLING 2010, Beijing, China, August 23-27, 2010), pp. 1335-1343.
17. S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, Collective Annotation of Wikipedia Entities in Web Text, in *Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD 2009), pp. 457-466.
18. J. Kleb and A. Abecker, Entity Reference Resolution via Spreading Activation on RDF-Graphs, in *Proc. of the 2010 Extended Semantic Web Conference* (ESWC 2010).
19. R. Bunescu and M. Paşca, Using encyclopedic knowledge for named entity disambiguation, in *Proc. of the 11th Conference of EACL*, pp. 9–16, 2006.
20. S. Cucerzan, Large-Scale Named Entity Disambiguation Based on Wikipedia data, in *Proc. of EMNLP-CoNLL Joint Conference 2007*, pp. 708-716, 2007.
21. H.T. Nguyen and T.H. Cao, A Knowledge-Based Approach to Named Entity Disambiguation in News Articles, in *Proc. of the 20th Australian Joint Conference on Artificial Intelligence* (AI 2007)*;* LNAI, vol. 4830, Springer-Verlag, pp. 619–624.
22. H.T. Nguyen and T.H. Cao, Exploring Wikipedia and text features for named entity disambiguation*, in Proc. of the 2nd Asian Conference on Intelligent Information and Database Systems* (ACIIDS 2010); LNCS, vol. 5991, Springer-Verlag, pp. 11-20.
23. H.T. Nguyen and T.H. Cao, Named entity disambiguation on an ontology enriched by Wikipedia, *in Proc. of the 6th IEEE International Conference on Research, Innovation and Vision for the Future* (RIVF 2008, Ho Chi Minh City, Viet Nam), pp. 247-254.
24. O. Medelyan, D. Milne, C. Legg, I.H. Witten, Mining Meaning from Wikipedia, in *International Journal of Human-Computer Studies*, 67(9): 716-754.
25. R. Mihalcea, Using Wikipedia for Automatic Word Sense Disambiguation, in *Human Language Technologies 2007: The Annual Conference of the North American Chapter of the Association for Computational Linguistics* (HLT/NAACL 2007, Rochester, New York, April 2007).

26. R. Navigli, Word Sense Disambiguation: A Survey, in *ACM Computing Surveys*, 41(2):1-69.
27. R. Mihalcea, A. Csomai, Wikify!: Linking Documents to Encyclopedic Knowledge, in *Proc. of the 16th ACM Conference on Information and Knowledge Management* (CIKM 2007), pp. 233-242.
28. D. Milne and I.H. Witten, Learning to Link with Wikipedia, in *Proc. of the 17th ACM Conference on Information and Knowledge Management* (CIKM 2008), pp. 509-518.
29. S. Overell and S. Rüger, Using co-occurrence models for placename disambiguation, *International Journal of Geographical Information Science*, 22(3):265-287, 2008.
30. D. Smith and G. Mann, Bootstrapping Toponym Classifiers, in *HLT-NAACL Workshop on Analysis of Geographic References*, pp. 45–49.
31. E. Garbin and I. Mani, Disambiguating Toponyms in News, in *Proc. of the conference on Human Language Technology and Empirical Methods in Natural Language*, pp. 363-370.
32. J. Leidner, *Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names*, Ph.D. thesis, School of Informatics, University of Edinburgh, 2007.
33. K. Bontcheva, M. Dimitrov, D. Maynard, V. Tablan, H. Cunningham, Shallow Methods for Named Entity Coreference Resolution, in *Proc. of TALN 2002 Workshop*, Nancy, France, 2002.
34. H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications, in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.
35. P. Cimiano and J. Völker, Towards Large-scale, Open-domain and Ontology-based Named Entity Classification, in *Proc. of Recent Advances in Natural Language Processing - 2005*, pp. 166-172, 2005.
36. E.F. Tjong Kim Sang and F. De Meulder, Introduction to the CoNLL-2003 Shared Task: Language Independent Named Entity Recognition, in *Proc. of CoNLL-2003*, pp. 142–147, 2003.
37. T. Berners-Lee, J. Hendler, and O. Lassila, The Semantic Web, in *Scientific American*, pp. 34–43, 2001.
38. D.M. Bikel, R.L. Schwartz, and R.M. Weischedel, An Algorithm That Learns What's in a Name, in *Machine Learning*, 34(1-3):211–231, 1999.
39. M. Fleischman and E. Hovy, Fine grained Classification of Named Entities, in *Proc. of the 19th international conference on Computational linguistics*, pp.1-7, 2002.
40. C.H. Gooi and J. Allan, Cross-document Coreference on a Large-scale Corpus, in *Proc. of HLT-NAACL for Computational Linguistics Annual Meeting*, pp.9-16, 2004.
41. Y. Chen and J. Martin, Towards robust unsupervised personal name disambiguation, in *Proc. of EMNLP-CoNLL Joint Conference 2007*, pp. 190–198, 2007.
42. T. Pedersen, A. Purandare, and A. Kulkarni, Name Discrimination by Clustering Similar Contexts, in *Proc. of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 226-237, 2005.
43. B. Malin, Unsupervised Name Disambiguation via Social Network Similarity, in *Proc. of SIAM Conference on Data Mining 2005*.
44. J. Artiles, J. Gonzalo, and S. Sekine, WePS 2 Evaluation Campaign*:* overview of the Web People Search Clustering Task, in *Proc. of 2nd Web People Search Evaluation Workshop*, 18th WWW Conference.
45. A. Kiryakov, B. Popov, I. Terziev, D. Manov, D. Ognyanoff. Semantic Annotation, Indexing, and Retrieval, in *Journal of Web Semantics*, 2(1), 2005.
46. S. Dill, *et al.* Semtag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation, in *Proc. of 12th WWW Conference*, pp.178–186. 2003.
47. Jim Giles. Internet encyclopedias go head to head, in *Nature* 438 (7070), pp. 900-901, 2005.
48. G. Mann and D. Yarowsky, Unsupervised Personal Name Disambiguation, in *Proc. of the 17th Conference on Natural Language Learning*, pp. 33–40, 2003.
49. X. Li, P. Morie, and D. Roth, Robust Reading: Identification and Tracing of Ambiguous Names, i*n Proc. of HLT-NAACL 2004*, pp. 17-24, 2004.
50. C. Niu, W. Li, and R. K. Srihari, Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction, in *Proc. of the 42nd Annual Meeting on Association for Computational Linguistics*, 2004.
51. V. Rastogi, N. N. Dalvi, and M. N. Garofalakis. Large-scale Collective Entity Matching, in *The Proceedings of the VLDB Endowment (PVLDB)*, 4(4):208-218, 2011.
52. A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate Record Detection: A Survey, in *IEEE Transactions* on *Knowledge* and *Data Engineering*, 19(1):1-16, 2007.
53. I. Bhattacharya and L. Getoor, Collective Entity Resolution in Relational Data, in *TKDD*, 1(1), 2007.
54. M. Bilenko and R. J. Mooney, Adaptive Duplicate Detection Using Learnable String Similarity Measures, in *Proc. of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD 2003), pp. 39-48, 2003.
55. I. Bhattacharya and L. Getoor, A Latent Dirichlet Model for Unsupervised Entity Resolution, in *The SIAM Conference on Data Mining (SIAM-SDM)*, 2006.
56. N. R. Smalheiser and V. I. Torvik, Author Name Disambiguation, in *Annual Review of Information Science and Technology*, 43, 287-313.
57. X. Wang, J. Tang, H. Cheng, and P. S. Yu, ADANA: Active name disambiguation, in *Proc. of 2011 IEEE International Conference on Data Mining* (ICDM'2011).
58. Z. Harris, Distributional structure, in *Word*, 10(23): 146-162, 1954.
59. S. Zhou, K. Li, and Y. Liu, Text Categorization Based on Topic Model, in *International Journal of Computational Intelligence Systems*, 2(4):398-409, 2009.

Published by Atlantis Press