Enriching Ontologies for Named Entity Disambiguation

Hien Thanh Nguyen
Ton Duc Thang University
98 Ngo Tat To St., 19 Ward, Binh Thanh District,
HCM City, Vietnam
hien@tdt.edu.vn

Tru Hoang Cao
HCM City University of Technology
268 Ly Thuong Kiet St., District 10,
HCM City, Vietnam
tru@cse.hcmut.edu.vn

***Abstract*— Detecting *entity mentions* in a text and then mapping them to their right *entities* in a given knowledge source is significant to realization of the semantic web, as well as advanced development of natural language processing applications. The knowledge sources used are often close ontologies - built by small groups of experts - and Wikipedia. To date, state-of-the-art methods proposed for named entity disambiguation mainly use Wikipedia as such a knowledge source. This paper proposes a method that enriches a close ontology by Wikipedia and then disambiguates named entities in a text based on that enriched one. The method disambiguates named entities in a text iteratively and incrementally, including several iterative steps. Those named entities that are identified in each iterative step will be used to disambiguate the remaining ones in the next iterative steps. The experiment results show that enrichment of a close ontology noticeably improves disambiguation performance.**

Keywords- entity disambiguation; ontology enrichment; annotation; named entity; ontology

I. INTRODUCTION

Named entities (NEs) are those that are referred to by names such as people, organizations, or locations. This paper addresses the named entity disambiguation problem (NED) that aims at mapping entity names in a text to right entities in a given source of knowledge. Having been emerging in recent years as a challenging problem, but significant to realization of the Semantic Web, as well as advanced development of Natural Language Processing applications, NED has attracted much attention by researchers all over the world. The problem in reality is that one name in different occurrences may refer to different entities 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, and so on. Such ambiguity makes identifying right entities in a text challenging and raises NED as a key research aspect in the above-mentioned areas.

NED can be considered as an important special case of Word Sense Disambiguation (WSD) [12]. 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 either appear in 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 the most widely-used encyclopedia in existence, Wikipedia[1] is used as a knowledge source for not only WSD, but also Information Retrieval, Information Extraction, Ontology Building, Natural Language Processing, and so on [9]. Proposed methods for WSD typically choose a set of features for representation of a target word (or its context) based on features of its surrounding words limited in a window context, and relationships among them and the target word. The context size is commonly set to ±3 or ±5 words around the target word. In recently years, some methods proposed for WSD have been adopted for NED [1][8][13]. When dealing with named entity disambiguation, many works focus on clues in a whole text [3][10][11] for disambiguation, but not just words around the named entity to be disambiguated.

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. 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 can detect all entities that have the same name in Wikipedia.

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 [17], WordNet [18]. 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

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

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 up date 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 [21], 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.

While state-of-the-art NED methods mainly use Wikipedia as the target knowledge source, there are still many application systems based on close ontologies. This paper thus focuses on mapping entity mentions in a text to a close ontology. It faces the following difficulties:

- Those methods proposed for NED using Wikipedia are not easy to adopt to close ontologies because they exploit Wikipedia-based features which do not appear in the close ontologies.
- While information describing entities in Wikipedia is diverse and rich, information describing entities in a close ontology is poor and mainly based on a given number of built-in properties of the entities in that ontology.

Therefore, for automatic mapping entity mentions in a text to a close ontology (henceforth *ontology*), we do need a new method to overcome the above-mentioned difficulties. This paper proposes a method that disambiguates named entities in a text using an ontology where descriptions of entities in that ontology are enriched by features extracted from Wikipedia. The contributions of our proposed method are as follows. First, the method enriches information describing entities in an ontology by their features extracted from Wikipedia, and then disambiguates named entities in a text based on that enriched ontology. Second, the method disambiguates named entities in a text iteratively and incrementally, including several iterative steps. Those named entities that are identified in each iterative step will be used to disambiguate the remaining ones in the next iterative steps. Third, the experiment results show that features extracted from Wikipedia to enrich representation of entities in an ontology noticeably improve disambiguation performance in comparable with not using those features.

The rest of this paper is organized as follows. Section 2 presents our statistical ranking model. Section 3 presents a process of ontology enrichment. Section 4 presents the proposed method for NED. Section 5 presents experiment results. Section 6 presents related works and a conclusion is drawn in Section 7.

II. A PROPOSED STATISTICAL RANKING MODEL

In this section, we present a statistical ranking model where we employ the Vector Space Model (VSM) to represent *ambiguous*[2] mentions and entities in a given knowledge source by their features. The VSM considers the set of features of each entity or mention as a 'bag of words'. We present how each bag of words is normalized. 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.

Normalization

After extracting features for a mention 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.; (iii) removing stop words such as *a*, *an*, *the*, etc.; and (iv) 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, suppose there are *N* candidate entities for it in a given knowledge source. 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:

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

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:

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

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, $s_{ij} = log(tf_j+1)\ .log(N/df_j)$.

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 Equation 1 as the scoring function. What we have just described is implemented in Algorithm 1. *Sim* is used at Line 3 of the algorithm.

[2] An *ambiguous* mention is a mention that is used to refer to two or more entities in a given knowledge source. We call these entities *candidate entities* of that mention.

Algorithm 1 Statistics-based Entity Ranking

```
let $C$ a set of candidate entities of $m$
for each candidate $c$ do
    $score[c] \leftarrow Sim(FeatureVector(c), FeatureVector(m))$
end for
$c^* \leftarrow \arg\max_{c_i \in C} score[c_i]$
if $score[c^*] > \tau$ then return $c^*$
return $NIL$
```

III. Ontology Enrichment

Usually, a built-in ontology in a system does not represent enough information about NEs, which causes misclassification and mis-identification of NEs referred to in a text with respect to that ontology. There are two kind of missing information of entities in an ontology. First, the ontology defines not enough properties of many entities. For instance, persons in PROTON ontology are represented by only four properties *hasPosition*, *hasProfession*, *hasRelative* and *isBossOf*. In reality, a person has a lot of different relations with other entities such as relation to persons other than relatives (e.g., Hillary Clinton, wife of Bill Clinton), or notable achievements (e.g., John McCarthy, inventor of LISP), etc. Second, some properties of a certain entity may be not assigned values.

To overcome these shortages of a close ontology, we need to enhance representations of entities in that ontology to enrich their attributes and relations by new features from another source of knowledge. In particular, in this paper, we exploit Wikipedia to generate features whose values provide additional information about focused NEs, such as location where one was born, or fellow-workers, etc., for enriching representation of NEs in a given ontology by an enrichment process. Then the disambiguation is performed using that enriched ontology. Such enrichment leads to representations of those entities in a richer space, which facilitates employment of a statistical model for disambiguation.

Before performing enrichment, entities in Wikipedia and in the ontology are already represented by their features. We call features extracted from the ontology for representing entities in it *ontology features* (OF). We call features extracted from Wikipedia for representing Wikipedia entities *Wikipedia features* (WF). Here we describe the features.

Ontology features

We utilize ontological concepts, and properties of entities in a specific ontology to extract their features. In particular, let I be a set of entities of an ontology $\boldsymbol{O}$; for each entity $i \in I$, the following features are extracted to represent it: (1) all classes to which i belongs; (2) attribute values of i; and (3) all names and identifiers of entities that have relationship with i or vice versa.

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 that describes Professor John McCarthy who is the inventor of LISP programming language. We extract "John McCarthy (computer scientist)" for the entity Professor John McCarthy.

– *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; John Towner Williams; 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.

– *Inlink labels* (IL). For an entity in Wikipedia, there are some links from other Wikipedia entities pointing into it. We extract labels of those inlinks as its possible features.

After extracting features for entities in Wikipedia and a given ontology, we put them into 'bag of words'. Then the bag of words are normalized and converted to feature vectors. Now we are ready to present the enrichment algorithm.

Enrichment Algorithm

We present steps that enrich representation of an entity $i \in I$ in an ontology $\boldsymbol{O}$ as follows:

– Step 1: The longest name of i, namely n, is used as a query to retrieve candidate entities from Wikipedia.

– Step 2: If the number of candidate entities in the returned set is higher than 1, go to Step 5; otherwise, go to Step 3.

– Step 3: If the number of candidate entities in the returned set is 1, that only one entity, namely c, is checked to be sure that it is the same as i. In particular, let R_i be a set of entities that have relationship with i in the ontology and W_c be a set of entities that have relationship with c in Wikipedia; if R_i is a subset of W_c, then i and c are considered as the same referent.

– Step 4: If there are not any entity in the returned set, prefixes and postfixes (e.g., Mr., company, inc., co., etc.) of n are removed. Then n becomes n'. Go to Step 2. For instance, if using "Columbia Sportswear Company" to retrieve candidate entities and the returned set is empty, the postfix "Company" is removed and then "Columbia Sportswear" is used as a query.

– Step 5: When the number of candidate entities in the returned set is higher than 1, Algorithm 1 is applied to rank the candidate entities. The candidate entity with the highest rank is chosen and its features are used to enrich representation of the corresponding entity in $\mathcal{O}$**.** Note that this algorithm does not exploit identifiers of entities in $\mathcal{O}$ as their features**.**

These steps are applied to enrich all entities in $\mathcal{O}$. Then we obtain a new ontology whose entity representations are enriched. Note that the feature generation and enrichment is performed prior to NE disambiguation, and is completely independent of the later steps; therefore, it can be built once and reused for NE disambiguation tasks in the future.

IV. NAMED ENTITY DISAMBIGUATION

We recall that the method this paper proposes to NED is to map entity mentions in a text to right entities a close ontology $\mathcal{O}$. After ontology $\mathcal{O}$ is enriched by Wikipedia, we obtain an enriched ontology $\mathcal{O}_e$**.** Then the method performs disambiguation based on $\mathcal{O}_e$. Each entity in $\mathcal{O}_e$ is represented by the features OF and WF as described above. To map a mention in a text to the right entity in $\mathcal{O}_e$, our method extracts features in the text to represent that mention. We call these features *text features* and describe them below.

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 co-referent 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.
– *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 optimum performance in the related task of cross-document coreference resolution ([6]). 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.
– *Identifiers* (ID). All identifiers of identified entities in a text are features.

Disambiguation

The proposed method in this paper disambiguates named entities in text iteratively and incrementally, including several iterative steps. Those named entities that are identified in each iterative step will be used to disambiguate the remaining ones in the next iterative steps. In other words, we exploit identifiers of identified entities in the text as extended parts of that text. These identifiers are used as features of the remaining ones.

Algorithm 2 implements the method. The loop statement at Line 3 stops when the set of identified entities E has no change between two iteration steps or all mentions are mapped to an entities ontology $\mathcal{O}_e$. Line 7 call Algorithm 1 to rank candidate entities of a mention. The *revised* function at Line 9 adjusts E using the coreference chain of a mention. For example, assume that in a text there are occurrences of coreferent mentions “Denny Hillis” and “Hillis; if “Denny Hillis” is recognized as referring to W. Daniel Hillis in Wikipedia for instance, then “Hillis” also refers to W. Daniel Hillis.

Algorithm 2 Iterative and Incremental Disambiguation

```
1:  let $\mathcal{N}$ be a set of mentions and $E$ be an *empty* set
2:  *flag* ← **false**
3:  **loop until** $\mathcal{N}$ *empty* or *flag is* ***true***
4:      $\mathcal{N}' \leftarrow \mathcal{N}$
5:      **for each** $n \in \mathcal{N}'$ **do**
6:          $C \leftarrow$ a set of candidate entities of $n$
7:          $\gamma^* \leftarrow$ run Algorithm 1 for $n$
8:          **If** $\gamma^*$ *is not NIL* **then**
9:              map $n$ to $\gamma^*$
10:             $E \leftarrow revised(E \cup \{n \rightarrow \gamma^*\})$
11:             remove $n$ from $\mathcal{N}$
12:         **end if**
13:     **end for**
14:     **if** *E no change* **then** *flag* = **true**
15: **end loop**
```

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 and is one of the longest mentions in its coreference chain, 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 entity in the ontology. Regarding the computational cost, since after each iteration of the outer loop there is at least one more mention resolved or E has no change, the worst case complexity is $O(N^2)$, where N is the number of mentions to be resolved.

V. Evaluation

First of all, we perform enrichment of KIM ontology by Wikipedia using the ontology enrichment algorithm presented in Section 3. For experiments, we build a dataset by collecting documents that contain mentions of entities in KIM ontology. All mentions are manually mapped to that ontology to form a golden standard corpus.

There are total 186 documents in the dataset. Table 1 presents information about the mentions that contain "Georgia" or "Columbia" in the dataset. The right column in the table shows the number of those mentions in the dataset referring to the corresponding entity in the left column. For instance, as showed in the second row of the table, there are 90 mentions referring to the entity Georgia – a state of the United States.

Since we aim at evaluating how good our method is in terms of disambiguation performance, we focus on ambiguous mentions. Therefore, in order to produce ambiguous mentions for the experiments, we replace each mention containing "Georgia" by only "Georgia" and each mention containing "Columbia" by only "Columbia". For instances, we replace "South Georgia and the South Sandwich Islands" by "Georgia", "Columbia University" by "Columbia", etc.

TABLE I. STATISTICS ABOUT AMBIGUOUS MENTIONS IN THE DATASET

Entity	# of mentions
Georgia (country)	318
Georgia (U.S. state)	90
South Georgia and the South	59
British Columbia	34
Columbia Sportswear Company	65
Columbia University	13
Columbia, South Carolina	15
Space Shuttle Columbia	80
District of Columbia	1
Total	675

TABLE II. STATISTICS ABOUT TOTAL AMBIGUOUS MENTIONS AND DISAMBIGUATED MENTIONS

Mention	# of candidate entities	# of total mentions	# disambiguated mentions
Georgia	7	468	463
Columbia	10	207	205
Total		675	668

Note that prior to disambiguation, we perform preprocessing tasks. In particular, we perform NE recognition and NE coreference resolution using natural language processing resources of Information Extraction engine based on GATE [5]. The NE recognition applies pattern-matching rules written in JAPE's grammar of GATE to detect and tag boundaries of mentions occurring in the dataset and then categorize corresponding entities as Person, Location and Organization, etc. After detecting all mentions occurring in the text, we run NE co-reference resolution [2] module in the GATE system to resolve the different mentions of a NE into one group that uniquely represents the NE. After that we run Algorithm 2 for disambiguation. In [16], the authors explored a range of features extracted from texts and Wikipedia, and vary combinations of those features to appraise which ones are good for NED. It shows that the Wikipedia features ET, RT, CAT and OL in combination with the text features EM, LW and CW give the best performance. Based on that finding, when conducting experiments, we focus on the combination OF + ET + RT + CL + OL with regard to Wikipedia features. Table 2 shows the number of candidate entities, the number of total ambiguous mentions and the number of disambiguated mentions.

We test the method in two settings of entity representation using the basic features extracted from the given ontology (i.e., OF) and using those basic features in combination with features extracted from Wikipedia (i.e., OF + ET + RT + CL + OL on the enriched ontology). Table 3 shows the experiment results in these settings. The third column of Table 3 shows the number of correct mappings of mentions in the dataset to their corresponding entities in the ontology. The results show that the features extracted from Wikipedia in combination with the basic features noticeably improve disambiguation performance in comparison with using the basic features only.

TABLE III. DISAMBIGUATION PERFORMANCE IN TERMS OF PRECISION AND RECALL

Mention	Features	# of correct mappings	P (%)	R (%)
Georgia	OF	310	66.95	66.23
	OF + ET + RT + CL + OL	436	94.16	93.16
Columbia	OF	171	83.41	82.60
	OF + ET + RT + CL + OL	183	89.26	88.40
Total	OF	481	72.00	71.25
	OF + ET + RT + CL + OL	619	92.66	91.70

VI. Related Works

There are many methods proposed for NED in literature. Methods disambiguating named entities based on Wikipedia are overwhelming. The method in [19] relies on affiliation, text proximity, areas of interest, and co-author relationship as clues for disambiguating person names in calls for papers only. Meanwhile, the domain of [20] is that of geographical names in texts. The authors use some patterns to narrow down the candidates of ambiguous geographical names. For instance, "*Paris, France*" more likely refers to the capital of France than a small town in Texas. Then, it ranks the remaining candidate entities based on the weights that are attached to classes of the constructed Geoname ontology. The method in [13] generates a co-occurrence model from article's templates that served as training data and then employed the SVM for place-name disambiguation. This method only works on co-occurrence place-names. It chooses a window size of ±10 location references regardless of other words that are not part of place-names. In contrast, the problem that we address in this paper is more general, which is not limited to named entities of a particular class or domain, but for all that may occur in a text.

In [8], authors implemented and evaluated two different disambiguation algorithms that extracted terms in a document and linked them to Wikipedia articles using Wikipedia as a sense inventory. Then they reported the best performing algorithm was the one using a supervised learning model where Wikipedia articles, which had already been annotated, served as training data. This algorithm used the local context of three words to the left and right, with their parts-of-speech, as features for representing an ambiguous term. In 2007, we proposed an idea of exploiting identified entities to disambiguate remaining ones [14]. Later on, in 2008, the works in [10] bore a resemblance to our idea for disambiguating terms in a documents using Wikipedia. The works in [11] extended both works [8] and [10] by exploiting relatedness of a target term to its surrounding context, besides exploiting the feature as in the latter one.

The works in [1] and [3] exploit several of the disambiguation resources such as Wikipedia articles (entity pages), redirection pages, categories, and links in the articles. The methods in [1] extracted words inside a 55-word window around a mention to form its feature vector. Based on the cosine similarity between feature vectors, they ranked candidate entities for a mapping and chose the one with the highest similarity score. Due to too low similarity scores with the cosine-based ranking in many cases, the authors employed the Support Vector Machine model (SVM) to learn a mapping from the context window to the specific categories of articles. The method in [3] exploited the same resources of information in Wikipedia for the disambiguation task as in [1]. This method simultaneously disambiguates all mentions in a document by maximizing the agreement among categories of candidate entities and maximizing the contextual similarity between contextual information in the document and context data stored for the candidate entities. The context data comprise appositives in the titles of articles and phrases that appear as anchor texts of links in the first paragraphs of the articles. The contextual information of a document contains all phrases occurring in the context data. The method in [15] exploited ET, CAT, OL and the most frequency words in each Wikipedia article to represent entities in Wikipedia. Then it calculated semantic relatedness using a random walk model for simultaneously disambiguating all mentions in a document.

VII. Conclusion

We proposed a method that enriches a close ontology and then disambiguates named entities in a text based on that enriched one. Our proposed disambiguating method is iteratively and incrementally, including several iterative steps. Those named entities that are identified in each iterative step will be used to disambiguate the remaining ones in the next iterative steps. The experiment results show that disambiguating named entities based on an ontology enriched by Wikipedia noticeably improves disambiguation performance in comparison with that of disambiguation based on the original ontology. Our method solves the problems of named entity disambiguation on a close ontology with poor entity descriptions and limited number of entity properties.

References

[1] R. Bunescu and M. Paşca, “Using encyclopedic knowledge for named entity disambiguation,” in Proc. of the 11th Conference of EACL, 2006, pp. 9–16.

[2] K. Bontcheva, M. Dimitrov, D. Maynard, V. Tablan, and H. Cunningham, “Shallow methods for named entity coreference resolution,” in Proc. of TALN 2002 Workshop, 2002.

[3] S. Cucerzan, “Large-scale named entity disambiguation based on Wikipedia data,” in Proc. of EMNLP-CoNLL 2007, 2007, pp. 708–716.

[4] W. Cohen, P. Ravikumar, and S. Fienberg, “A comparison of string distance metrics for name-matching tasks,” in IJCAI-03 II-Web Workshop, 2003, pp. 73-78.

[5] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, “GATE: A framework and graphical development environment for robust NLP tools and applications,” in Proc. of ACL’02, 2002, pp.168-175.

[6] C. H. Gooi and J. Allan, “Cross-document coreference on a large-scale corpus,” in Proc. of HLT/NAACL’04, 2004, pp. 9-16.

[7] R. Mihalcea, “Using Wikipedia for automatic word sense disambiguation,” in Proc. of HLT/NAACL’07, 2007, pp. 196–203.

[8] R. Mihalcea and A. Csomai, “Wikify!: Linking documents to encyclopedic knowledge,” in Proc. of CIKM’07, 2007, pp. 233–242.

[9] O. Medelyan, D. Milne, C. Legg, and I. H. Witten, “Mining meaning from Wikipedia,” International Journal of Human-Computer Studies, 67(9), 2009, pp. 716-754.

[10] O. Medelyan, I. H. Witten and D. Milne, “Topic indexing with Wikipedia,” in Proc. of WIKIAI’08.

[11] D. Milne and I. H. Witten, “Learning to link with Wikipedia,” in Proc. of CIKM’08, 2008, pp. 509–518.

[12] R. Navigli, “Word sense disambiguation: A Survey,” ACM Computing Surveys, 41(2), 2009, pp. 1-69.

[13] S. Overell and S. Rüger, “Using co-occurrence models for placename disambiguation,” The IJGIS, Taylor and Francis, 2008, pp. 265-287.

[14] H. T. Nguyen and T. H. Cao, “A Knowledge-based approach to named entity disambiguation in news articles,” in Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), Springer, vol. 4830, 2007, pp. 619–624.

[15] A. Gentile, Z. Zhang, L. Xia and J. Iria, “Semantic relatedness approach for named entity disambiguation,” in Proc. of 6th Italian Research Conference on Digital Libraries - IRCDL 2010, 2010.

[16] H. T. Nguyen and T. H. Cao, “Exploring Wikipedia and text features for named entity disambiguation,” in: N.T. Nguyen, M.T. Le, and J. Świątek (Eds.): ACIIDS 2010, Part II, LNCS, Springer, vol. 5991, 2010, pp.11–20.

[17] A. Kiryakov, B. Popov, I Terziev, D. Manov, and D. Ognyanoff, “Semantic annotation, indexing, and retrieval,” Journal of Web Semantics, 2(1), 2005, pp.49-79 .

[18] G. A. Miller, “WordNet: A lexical database for English,” Communications of the ACM, 38, 1995, pp.39–41.

[19] J. Hassell, B. Aleman-Meza, and I. B. Arpinar, “Ontology-Driven Automatic Entity Disambiguation in Unstructured Text,” in Proc. of ISWC2006, 2006, pp. 44–57.

[20] V. Raphael, K. Joachim, and M. Wolfgang, “Towards Ontology-based Disambiguation of Geographical Identifiers,” in Proc. of the 16th WWW Workshop on I3: Identity, Identifiers, Identifications, 2007.

[21] Jim Giles, “Internet encyclopaedias go head to head,” Nature 438 (7070), 2005, pp. 900-901.