An Optimization Framework for Entity Recognition and Disambiguation

Chuan Wu
School of Information Management
Wuhan University
Wuhan Hubei China
008602768756027
wu.chuan@whu.edu.cn

Wei Lu
School of Information Management
Wuhan University
Wuhan Hubei China
008602768752757
reedwhu@gmail.com

Pengcheng Zhou
School of Information Management
Wuhan University
Wuhan Hubei China
008602768756027
pc.zhou@whu.edu.cn

ABSTRACT

We present a system for entity recognition and disambiguation (ERD) in short text, aiming at identifying all text fragments referring to an entity contained in Freebase. The task is organized in two steps. Given a short text the first step is discovering text fragments which possibly refer to an entity. Since multiple entities may share common mention, identifying which entity the mention is referring to in the given short text is necessary. Our system integrates three kinds of features: mention-entity similarity, entity-entity similarity and context-mention entity similarity. By considering every possible combination of mention-entity pair, we select the one with highest confidence score. An implementation of our system is described, along with our evaluation results. Experiments show that the proposed features improve the performance to a certain extent.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information storage and retrieval - *Information Search and Retrieval*

General Terms: Algorithms, Experimentation.

Keywords

Entity Linking; Entity Recognition; Entity Disambiguation.

1. INTRODUCTION

Entity Linking is the task of identifying text fragments in text which refer to an entity in a knowledge base, such as Wikipedia and Freebase. It enriches unstructured text with entities contained in knowledge base, helping people understand web pages and other documents online when they encounter unfamiliar entities. It also has potential usage in many NLP tasks, such as information extraction, text classification and clustering, and information retrieval. For example, by linking entity mention "Michael Jordan" in the query "Michael Jordan basketball" to its referent entity "Michael Jeffrey Jordan", the NBA basketball player, we can better understand the user intent behind the query and provide accurate retrieval results.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ERD'14, July 06-11, 2014, Gold Coast, QLD, Australia.

Copyright © 2014 ACM 978-1-4503-3023-7/14/07$15.00.

http://dx.doi.org/10.1145/2633211.2634362

Entity Linking (EL) is generally divided into two steps: entity recognition and entity disambiguation. The major challenge of entity recognition lies in rich name variations of entities, which is difficult to collect completely. It is especially the case for those evolving social network services, such as Twitter and Microblog. The challenge of entity disambiguation can be summarized as two types of ambiguities: Polysemy and Synonymy. They refer to the one-to-many and many-to-one relationship between entity mention and entity respectively. For example, the mention "Obama" may refer to multiple entities, such as "Mount Obama", the highest point in Antigua and Barbuda, and "Barack Obama", the 44th President of the United States. The entity "Barack Obama" can be referred to by a set of mentions, such as "Obama", "Barack Obama", and "Barack Hussein Obama".

Entity Recognition and Disambiguation Challenge [1] (ERD) was organized to promote communication in this research field by providing standard evaluation to all participants. As described in the announcement of ERD Challenge, an ERD system should recognize mentions of entities in a given text, and disambiguate them by mapping them to known entities in a given knowledge base. Two tracks are included in ERD Challenge, i.e. Long Track and Short Track, and we mainly focus on the Short Track. Our ERD system works by publishing a web service, which receives a short text as input and outputs a confidence score in the range of (0, 1) for each mention-entity pair. The confidence score indicates the probability of the mention referring to its corresponding entity. Higher confidence score means more confidence on the linking decision.

We propose an optimization method which leverages context words and mentions to disambiguate among candidate entities. Given a short text, our model tries to identify possible mentions and its candidate entities, and then calculate the confidence score of each possible combination of mention-entity pairs. The one with highest score is selected as the result. Several local features are adopted, such as commonness, context similarity, to measure the possibility of a mention referring to an entity. In addition to adopting in-link based similarity, we define additional entity-entity similarity features, including category similarity, mutual reference, to model the similarity between entities. Finally, we introduce context mention similarity between a candidate entity and its context mention, i.e. all mentions of the short text except the one that it is referred to by.

We evaluate our method by the public available evaluation service provided by Entity Recognition and Disambiguation Challenge[1]. The service send short text queries to our system and receives linking results in a specified format. Then it analyze the results to calculate an expected F1 as evaluation result. Experimental results show that our method improves performance to a certain extent.

We also study the effectiveness of each kind of features and demonstrate the effectiveness of our method.

The rest of this paper is organized as follows: Section 2 introduces related work in this research field. Then our method in entity recognition is described in Section 3. We present our disambiguation approach in Section 4. Section 5 describes our experiment results and discusses lessons learned. Finally, we draw the conclusion in Section 6.

2. RELATED WORK

Entity linking consists of two sub-steps: mention detection and entity disambiguation. Existing entity linking work can be divided into three categories according to how they organize this two steps. Methods of the first category go through these two steps in sequential order, using the output of the first step as the input of the second. They assume the output of the first step as ground truth, and disambiguate among candidate entities of detected mentions. Methods of the second category assume that entity mentions are provided by a separate NER system, and solely focus on entity disambiguation. The third category is similar with the first category, except that they do not fully trust the output of the first step. They assume possible false detected mention and focus on high recall in the first step, and then make joint inference towards the right mention-entity pair.

Examples of the first category include Bunescu & Pasca [7], Cucerzan [10]. Bunescu & Pasca [7] extracts information from pages in Wikipedia such as redirect pages, disambiguation pages, to detect entity mentions first, and then turns to a scoring function which considers various similarity features to disambiguate among candidate entities of each mention. Cucerzan [10] detects mentions in a similar way, adding category information of Wikipedia articles, and employs a vector space model to find mention-entity matches which maximize context coherence.

Studies belonging to the second category define the entity linking problem as linking entity mentions to entities, and mainly focus on entity disambiguation. Ratinov et al. [11] assume a document with a set of mentions, setting their goal as producing a mapping from the set of mentions to a set of Wikipedia definition pages. Han et al. [12] formulate the Entity Linking problem as disambiguating name mentions by the document containing those mentions and a knowledge base. In other words, they do not take mention detection into consideration and directly handle the disambiguation problem. Many other works formulate Entity Linking problem in a similar way and focus on exploiting interdependence between EL decisions.

Many researchers notice that mention detection itself can be erroneous. Therefore, joint inference on mention detection and entity disambiguation is necessary. Stern et al. [13] detect all possible mentions, preserving a number of ambiguous readings. Then the linking component will evaluate the most likely mentions and its corresponding entities. Sil & Yates [14] utilize base mention detection techniques to produce candidate mentions and propose promising entity links for each candidate mention using base EL system. A final re-ranking model is then used to choose among the set of all possible mention-entity pairs.

Traditional entity linking research mainly focus on long term text, such as web pages and other kinds of long documents. Recently the problem of linking on short term text attracts attention. As we know, the first work addressing this problem was TAGME [2]. It extends the approach of Milne & Witten [3] by adding a voting schema. Afterwards, various authors have attempted to annotate either general short texts or microblogs and tweets. Meij et al. [4] see the importance of understanding microblog posts and tried to add semantics to posts by first identifying concepts related to it and then generating links to corresponding articles in Wikipedia. However, the method proposed by Meij et al. [4] only considers the similarity between microblog posts and entities in Wikipedia, ignoring the similarity between different microblog posts. Liu et al. [5] propose a collective inference model that integrates three kinds of similarities, including mention-mention similarity, entity-entity similarity, and mention-entity similarity. Guo et al. [6] find that mention detection is often the bottleneck, and jointly optimize mention detection and disambiguation by combining various first-order, second-order and context-sensitive features.

3. MENTION DETECTION

Given the input short text ST, our system outputs a sequence of mentions $\overrightarrow{M} = (m_1, m_2, \cdots, m_n)$, and the corresponding entity sequence $\overrightarrow{E} = (e_1, e_2, \cdots, e_n)$, where e_i is the entity refer to by m_i. An entity refers to an entry of a knowledge base. Following most existing work, we use Wikipedia as our knowledge base. Note that as several kinds of page exist in Wikipedia, only definition pages are treated as entities.

Two stages are included in our system: a mention detection stage to identify all possible mentions through a mention-entity dictionary, and a disambiguation stage to link a mention to its most possibly referent entity.

This section describes our method in mention detection. We first describe how the mention entity dictionary is constructed. Then the process of mention detection is introduced. The disambiguation stage will be described in the next section.

3.1 Mention Entity Dictionary

We adopt the method proposed by Bunescu & Pasca [7], constructing our mention-entity dictionary by leveraging the rich link structure in Wikipedia.

For each entity in Wikipedia, we extract the following information as its mention:

Entity Title. The title of an entity is obviously a mention of the entity. For example, "Barack Obama" is the title of entity named "Barack Obama", and people often use the title to refer to the entity.

Disambiguation Page Title. The disambiguation page for an ambiguous name is also useful. All possible referent entities of a name are listed on the page, which directly reflect the one-to-many relationship between mention and entity. For example, the disambiguation page for the name "Obama" lists 17 possibly referent entities, including Barack Obama, Mount Obama, Obama Line and so on.

Redirect Page Title. The redirect page of an entity refers to a page without content except for a redirect link to the entity. It means that when people mention the title of the redirect page, what they really mean is the redirecting entity. Therefore, the title of a redirect page can be viewed as an alias of an entity. For example, the entity "Barack Obama" can be redirected from redirect page "Obama", which means that when we talk about Obama, we often mean "Barack Obama".

Anchor Text. Articles in Wikipedia contain rich labeled anchor texts referring to another article. These anchors and its referent articles are manually labeled one-to-one relationship between mention and entity, thus providing rich name variations for entities in Wikipedia. It's obvious that anchor text is sometimes the same with the title of the referent article, but it still provides many alias of entities because of the volume of Wikipedia. For example, the following sentence in entity "Obama" contains two anchors: "Obama called for [[United States Congress | Congress]] to pass legislation reforming [[health care in the United States]]". The former anchor "Congress" is an alias of its referent entity "United States Congress", while the latter anchor "health care in the United States" is exactly the same with the title of its referent entity.

As we focus on short text and all mentions in short text are lowercased, we lowercased all mentions of an entity in the mention entity dictionary for retrieval convenience. After building an entity-mention mapping dictionary, we convert it to an mention-entity dictionary. Thus, given a mention we could find all its possibly referent entities.

3.2 Mention Detection Process

In this process, we detect all mentions in short text and output it to the entity disambiguation component. The goal of mention detection is to achieve high recall by detecting all possible entity mentions. The mention detection process is shown in Figure 1.

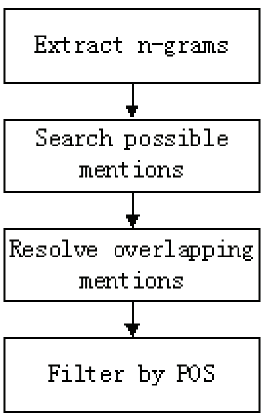

Figure 1: Mention Detection Process

We extract all n-grams in short text first. Then we search through the mention entity dictionary to find all n-grams which has an entry in the mention entity dictionary. However, overlapping problem may exist in this phase, when the intersection of two mentions is not empty. The most common case is that a mention with multiple words may contain one or more shorter mention. For example, given a short text "montclair elementary school", we detect four mentions: montclair, school, elementary school, montclair elementary school. The mention "elementary school" contains another mention "school", while the mention "montclair elementary school" contains all other three mentions. The strategy we adopted is Maximum Length Matching strategy, which means searching from the beginning of the short text, and then choosing the longest matched mention. The detection of the next mention begins from the end of the last longest matched mention. In the former example, we choose the longest mention, i.e. "montclair elementary school", and filter out all containing mentions.

Besides, we observe that some mentions can be filtered out by their part-of-speech. As entity mentions are nouns in most cases, non-noun mentions can be removed from mention set. We first detect the part-of-speech of each mention in the given context using Stanford POS tagger[1], and then filter non-noun mentions out. For example, given a short text "Barack Obama visit Japan", we detect "visit" as a mention, as visit may refer to State Visit. However, "visit" is a verb in the short text and should not be considered as a mention here. We remove "visit" from the mention set, thus reducing the time spent in disambiguation.

4. ENTITY DISAMBIGUATION

4.1 Framework

Given the input short text ST, our system outputs a sequence of mentions $\overrightarrow{M} = (m_1, m_2, \cdots, m_n)$, and the corresponding entity sequence $\overrightarrow{E^*} = (e_1^*, e_2^*, \cdots, e_n^*)$ according to Formula 1:

$$\begin{aligned}\overrightarrow{E^*} = argmax_{\forall \overrightarrow{E} \in C(\overrightarrow{M})} \alpha \cdot \sum_{i=1}^{n} \overrightarrow{a} \cdot \overrightarrow{f}(m_i, e_i) \\ + \beta \cdot \sum_{i \neq j} \overrightarrow{b} \cdot \overrightarrow{g}(e_i, e_j) \\ + \gamma \cdot \frac{1}{n-1} s(\sum_{k=1,k\neq i}^{n} m_k, e_i)\end{aligned} \quad (1)$$

Where:

- $C(\overrightarrow{M})$ is the set of all possible entity sequences for the mention sequence $\overrightarrow{M}$;
- $\overrightarrow{E}$ denotes an entity sequence, including n entities;
- $\overrightarrow{f}(m_i, e_i)$ is the feature vector that models the similarity between detected mention m_i and one of its candidate entity e_i;
- $\overrightarrow{a}$ is the weight vector related to $\overrightarrow{f}(m_i, e_i)$, where $a_k \in (0, 1)$, $k = 1, 2, 3, 4, 5$, and $\sum_{k=1}^{5} a_k = 1$.
- $\overrightarrow{g}(e_i, e_j)$ is the feature vector that models the similarity between two entities e_i and e_j;
- $\overrightarrow{b}$ is the weight vector related to $\overrightarrow{g}(e_i, e_j)$, where $b_k \in (0, 1)$, $k = 1, 2, 3, 4$, and $\sum_{k=1}^{4} b_k = 1$.
- $\overrightarrow{s}(\sum_{k=1,k\neq i}^{n} m_k, e_i)$ is the feature vector that models the coherence between an entity and all mentions except the mention by which it is possibly referred to;
- $\alpha, \beta, \gamma \in (0, 1)$ are systematic parameters, which is determined by the training data. They are used to adjust the trade-off among three set of features mentioned above. $\alpha + \beta + \gamma = 1$.

As sometimes there is only one mention in short text ST and the mention exactly equals with ST, we can use no context information and thus can not disambiguate among candidate entities. We just view each candidate entities as a valid interpretation and return the link probability as our confidence score.

Note that $C(\overrightarrow{M})$ represents the search space. It is generated by using mention entity dictionary. In disambiguation phase, we first find all candidate entities of detected mentions, and then get all possible combinations of mention-entity pair. We calculate the confidence score of each combination and find the one with highest score. As the size of possible combination increase

[1] http://nlp.stanford.edu/software/tagger.shtml

exponentially, we cut out the entities whose prior probability is smaller than one third of the maximum prior probability of all candidate entities.

4.2 Features

We use three kinds of features: local features related to mention-entity similarity, global coherence features among entities in each possible entity sequences for the input mention sequence and global features related to context-mention entity similarity.

4.2.1 *Local Features*

• Prior Probability

$$f_1(m_i, e_i) = \frac{count(m_i, e_i)}{\sum_{\forall e_k \in C(m_i)} count(m_i, e_k)} \quad (2)$$

where $count(m_i, e)$ denotes the frequency of mention m_i referring to entity e in Wikipedia articles.

• Context Similarity

$$f_2(m_i, e_i) = \frac{\text{cooccurence number}}{\text{short text length}} \quad (3)$$

where "co-occurrence number" is the number of words that occur in both the short text containing m_i, and the Wikipedia article of e_i; "short text length" denotes the number of words in the short text containing m_i.

• Edit Distance Similarity

If the equation $Abs(Length(m_i) - Length(e_i)) = ED(m_i, e_i)$ is true, $f_3(m_i, e_i)$ is 1, otherwise 0. $Abs(\cdot,\cdot)$ computes the absolute value of the given expression, and $ED(\cdot,\cdot)$ computes the character level edit distance of the given parameters.

• Mention Contains Title

$$f_4(m_i, e_i) = \begin{cases} 1 & \text{if } m_i \text{ contains title of } e_i \\ 0 & otherwise \end{cases} \quad (4)$$

• Title Contains Mention

$$f_5(m_i, e_i) = \begin{cases} 1 & \text{if title of } e_i \text{ contains } m_i \\ 0 & otherwise \end{cases} \quad (5)$$

4.2.2 *Global Features related to Entity Similarity*

• Category based Similarity

$$g_1(e_i, e_j) = \frac{|c(e_i) \cap c(e_j)|}{|c(e_i) \cup c(e_j)|} \quad (6)$$

where $c(e)$ is the set of categories of Wikipedia article related to entity e.

• In-link based Similarity

$$g_2(e_i, e_j) = \frac{|il(e_i) \cap il(e_j)|}{|il(e_i) \cup il(e_j)|} \quad (7)$$

where $il(e)$ is the set of Wikipedia articles that have a link to entity e.

• Out-link based Similarity

$$g_3(e_i, e_j) = \frac{|ol(e_i) \cap ol(e_j)|}{|ol(e_i) \cup ol(e_j)|} \quad (8)$$

where $ol(e)$ is the set of Wikipedia articles that entity e link to at least once.

• Mutual Reference

$$g_4(e_i, e_j) = \begin{cases} 0 & \text{if } e_i \not\leftrightarrow e_j \\ 0.5 & \text{if } e_i \rightarrow e_j \text{ or } e_j \rightarrow e_i \\ 1 & \text{if } e_i \leftrightarrow e_j \end{cases} \quad (9)$$

The arrow in the equation means that an entity has a link to the pointing entity. This features helps to detect whether two entity are directly related.

4.2.3 *Global Features related to Context Mention Entity Similarity*

Context mention entity similarity is defined in Formula 10:

$$s(\sum_{k=1,k\neq i}^{n} m_k, e_i) = \frac{\sum_{k=1,k\neq i}^{n} contains(e_i, m_k)}{n-1} \quad (10)$$

If m_k is found in Wikipedia article related to e_k, the equation $contains(e_i, m_k)$ equals 1, otherwise 0.

5. EXPERIMENT

5.1 Experimental Settings

Following most previous works, we choose Wikipedia[1] as our knowledge base. We process Wikipedia definition pages, disambiguation pages and redirect pages using JWPL[2]. In total, the knowledge base we used contains over 4,000,000 pages, including definition pages, disambiguation pages and so on.

As Freebase is the chosen knowledge base of ERD Challenge, we use the Freebase Wikipedia Mapping provided by ERD Challenge Organizers to generate required output. If an entity is recognized in Wikipedia and not found in the Freebase Wikipedia Mapping, we label it as "NIL", which means that no referent entity of the mention is found.

5.2 Results and Discussion

To train our model, we extract a gold-standard data set from Wikipedia. As we focus on short text, we construct a collection of randomly selected fragments of text from Wikipedia pages, containing one or more anchors annotated by Wikipedia contributors. They are sent to our system as short text queries and the output is compared with the annotations in Wikipedia.

We choose those text fragments according to rules as follows:

• Containing one to three anchors;

• One to two words before the first anchor and after the last anchor are included as context;

We evaluate our method in different settings as follows:

(1) using only local features;

(2) using local features and global features related to entity-entity similarity;

(3) using all three set of features.

We use average F1 as our evaluation metrics, as designed by ERD Challenge. That is, the F1 is calculated for each query, and then

[1] We download the 20131202 version of Wikipedia.

[2] http://code.google.com/p/jwpl/

the F1 for each query are averaged to get the final F1 on a set of queries. The metrics are computed as:

$$Precision_i = \frac{|M \cap M^*|}{|M|}$$

$$Recall_i = \frac{|M \cap M^*|}{|M^*|}$$

$$F1_i = \frac{2PR}{P+R}$$

$$F1 = \frac{\sum_{i=1}^{n} F1_i}{n}$$

M means the output mention-entity pair of our system for a given short text, while M^* means the gold standard mention-entity pair annotated by ERD Challenge organizers. After the F1 for each query is calculated, all F1 is averaged to get the final F1.

Table 1 reports the results when using only local features. We evaluate the effectiveness of each feature by incrementally adding features. The result shows that using Prior Probability feature yields a reasonable F1, while adding Context Similarity and Edit Distance Similarity shows negative contribution to the F1. The performance of the latter two features is better, while still do not help. It means that token level context information of a mention may not be effective in disambiguation.

Table 1: Results of Local Features added incrementally. P.P., C.D., E.D.S., M.C.T.S. and T.C.M.S. means Prior Probability, Context Similarity, Edit Distance Similarity and Mention Contains Title, and Title Contains Mention Similarity, respectively.

Local Features	Expected F1
P.P.	0.5254
+C.S.	0.5214
+E.D.S.	0.5214
+M.C.T.S.	0.5254
+T.C.M.S.	0.5274

Table 2 shows the results with global features related to entity-entity features added incrementally. The results show that, in-link based feature and out-link based feature do not improve the performance in short text linking, while Mutual Reference is effective in indicating the coherence of candidate entities in a candidate mention-entity pair combination.

Table 2: Results of Global Features related to Entity-Entity Similarity. C.b.S., I.b.S., O.b.S. And M.R. Means Category based Similarity, In-link based Similarity, Out-link based Similarity and Mutual Reference respectively.

Global Features related to Entity-Entity Similarity	Expected F1
C.b.S.	0.5274
C.b.S.+I.b.S.	0.5294
C.b.S.+O.b.S.	0.5274
C.b.S.+M.R.	0.5374
C.b.S.+I.b.S.+O.b.S.	0.5274
C.b.S.+I.b.S.+O.b.S.+M.R.	0.5374

Table 3 shows the results with Context-Mention Similarity. We explore the effectiveness of context mention feature, and find that it does not affect the performance. We analyze at our results and find that the number of context mentions in short text is sparse. In most short text query, only two or three mentions are detected. Thus only one or two context mentions can be used, which makes the feature ineffective.

Table 3: Results with and without Context Mention Entity Similarity. C.M.E.S. means Context Mention Entity Similarity.

Context Mention Entity Similarity	Expected F1
+C.M.E.S.	0.5374

6. CONCLUSIONS AND FUTURE WORK

In this paper, we propose an optimization framework that links mentions found in short text to its corresponding entity in Wikipedia and Freebase. We propose some new features to reveal the coherence among candidate entities of mentions. We evaluate our method by a publicly available web service. The experimental result shows that our method improve the performance.

We do no utilize the rich structured information in Freebase. In the future, we plan to first enlarge the entity mention list using alias information in Freebase. In addition, we will leverage information in various fields of Freebase entities.

7. ACKNOWLEDGMENTS

The work is supported by the National Social Science Fund of China (Grant No. 12&ZD221 and the National Natural Science Foundation of China (Grant No. 71173164) . We also thank the ERD organizers for their efforts in ERD Challenge 2014.

8. REFERENCES

[1] Carmel, D., Chang M.W., Gabrilovich E., Hsu B. P., Wang K. 2014. ERD 2014: Entity Recognition and Disambiguation Challenge. SIGIR Forum.

[2] Ferragina, P. and U. Scaiella. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*. 2010: ACM. DOI= http://dx.doi.org/10.1145/1871437.1871689

[3] Milne, D. and I.H. Witten. Learning to link with wikipedia. in *Proceedings of the 17th ACM conference on Information and knowledge management*. 2008: ACM. DOI= http://dx.doi.org/10.1145/1458082.1458150

[4] Milne, D. and I.H. Witten. Learning to link with wikipedia. in *Proceedings of the 17th ACM conference on Information and knowledge management*. 2008: ACM. DOI= http://dx.doi.org/10.1145/1458082.1458150

[5] Meij, E., W. Weerkamp and M. de Rijke. Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on Web search and data mining*. 2012: ACM. DOI= http://dx.doi.org/10.1145/2124295.2124364

[6] Liu, X., et al. Entity linking for tweets. in *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. 2013.

[7] Guo, S., M. Chang and E. Kıcıman. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of NAACL-HLT*. 2013.

[8] Bunescu, R.C. and M. Pasca. Using Encyclopedic Knowledge for Named entity Disambiguation. in EACL. 2006.

[9] Mihalcea, R. and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. 2007: ACM. DOI= http://dx.doi.org/10.1145/1321440.1321475.

[10] Zhang, W., et al. Entity linking leveraging: automatically generated annotation. In *Proceedings of the 23rd International Conference on Computational Linguistics*. 2010: Association for Computational Linguistics.

[11] Ratinov, L., et al. Local and Global Algorithms for Disambiguation to Wikipedia. In ACL. 2011. DOI= http://doi.acm.org/10.1145/90417.90738.

[12] Han, X., L. Sun and J. Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 2011: ACM. DOI= http://dx.doi.org/10.1145/2009916.2010019.

[13] Stern, R., et al. A joint named entity recognition and entity linking system. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*. 2012. Avignon, France: Association for Computational Linguistics.

[14] Sil, A. and A. Yates. Re-ranking for joint named-entity recognition and linking. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. 2013: ACM. DOI= http://dx.doi.org/10.1145/2505515.2505601.