NLPR_KBP in TAC 2009 KBP Track: A Two-Stage Method to Entity Linking

Xianpei Han, Jun Zhao
National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences,
Beijing, China, 100190

Abstract

This paper describes the NLPR Knowledge Base Population system (NLPR_KBP) for the TAC 2009 KBP track. Our system employs a two-stage entity linking method, where the two stage corresponds to the two main components of our system: The first component is a Multi-way Entity Candidate detector, which identifies all the possible entities in the knowledge base for an entity mention based on a variety of knowledge sources, such as the Wikipedia Anchor Dictionary, the Web, etc. The second component is an Entity Linker, which links an entity mention with the real world entity it refers to by measuring the similarity between them based on the Wikipedia semantic knowledge. The evaluation proves the validity of our system.

1 Introduction

The KBP track had two tasks in the TAC 2009: the Entity Linking and the Slot Filling. We took part in the Entity Linking task, which was to link an entity mention to the real world entity (presented in a given knowledge base) it refers to. For example, given the following sentences containing the entity mention *ABC*:

1) ***ABC** could not directly access the underlying file system and operating system.*
2) *In American, **ABC** first broadcast on television in 1948.*

An Entity Linking system should link the two *ABC* mentions respectively with the real world entity *American Broadcasting Company* and the real world entity *ABC(programming language)*.

The main difficult of Entity Linking is caused by two reasons:

1) **The Mention Diversity of Entity**, that is, an entity can be mentioned in a variety of ways, which are hard to be enumerated. For example, the *IBM Company* can be mentioned in more than 40 forms, such as *IBM*, *Big Blue* and *International Business Machine*, which correspond to its abbreviation, nickname and its Official Title. Due to the mention diversity of entity, we usually cannot find the entity candidate of an entity mention easily, that is, given an entity mention such as the *ABC* in above, we usually cannot find all the entities it may refer to using a simple method.
2) **The Ambiguity of Entity Mention**, that is, an entity mention may refer to a dozen of entities. For example, the entity mention *AI* can refer to more than 10 real world entities, such as *Artificial intelligence*, *Game artificial intelligence*, *the Singer Ai*, etc. Due to the ambiguity of

entity mention, the entity linking system must determine the actual referent entity of an entity by leveraging their contextual information and background knowledge.

To solve the problems caused by the above two reasons, our NLPR_KBP system employs a two-stage method, corresponding to the two main components of our system, as shown in Figure 1. The first component of our NLPR_KBP system is a multi-way entity candidate detector, which detects all the possible entities an entity mention may refer to. Then, based on the entity candidates detected by the first component, the Entity linker component links the entity mention to the most likely entity candidate by comparing their contextual information. In following we describe each step in detail.

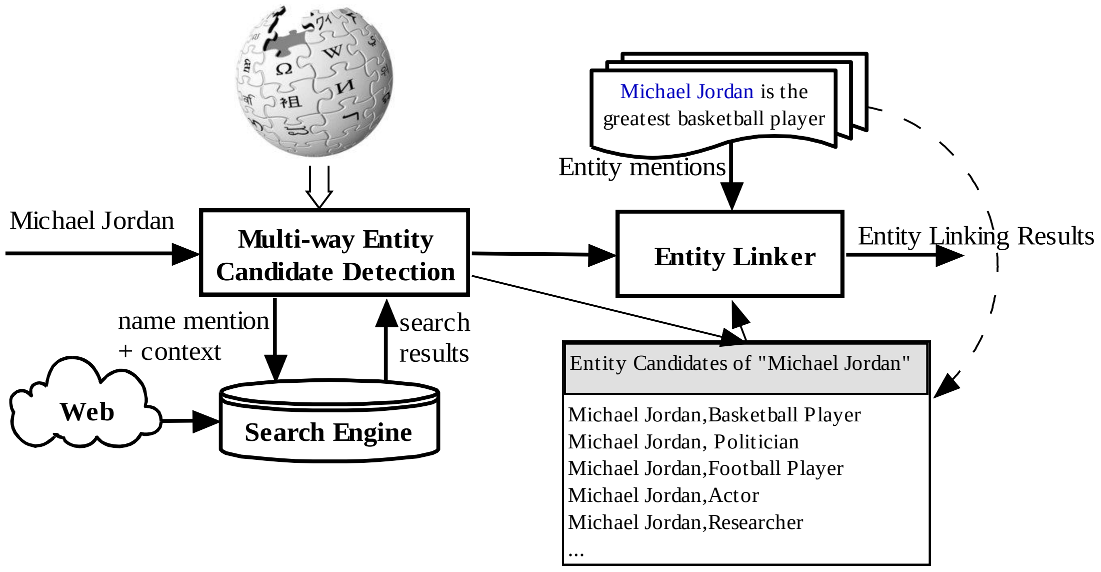

Michael Jordan is the greatest basketball player
Entity mentions
Michael Jordan
Multi-way Entity Candidate Detection
Entity Linker
Entity Linking Results
name mention + context
search results
Web
Search Engine
Entity Candidates of "Michael Jordan"
Michael Jordan,Basketball Player
Michael Jordan, Politician
Michael Jordan,Football Player
Michael Jordan,Actor
Michael Jordan,Researcher
...

Figure 1. The Framework of NLPR_KBP System

2 The Multi-way Entity Candidate Detector

In this section, we describe our entity candidate detector. As shown above, due to the mention diversity of entity, the entity candidates cannot be enumerated through heuristic rules, so we need to find ways to detect them. In NLPR_KBP system, we detect the entity candidates through multi-ways: 1) detecting using contextual information; 2) detecting using Wikipedia Anchor Dictionary; 2) detecting through web search. Through the above three ways, we would be able to detect the entity candidates of an entity mention by leveraging the information in the local document, the Wikipedia and the Web.

2.1 Candidate Detection Using Contextual Information

The context of entity mention usually contains rich information about its entity candidate, especially for abbreviation name mentions. For example, given the following sentences:

…the newly-formed All Basotho Convention (ABC) is far from certain…

…Abbott Laboratories (ABT:NYSE) …

…the Anti-Corruption Unit (ACU) of the International Cricket Council (ICC) …

…member countries of Asian Clearing Union (ACU) recorded…

We can easily identify the entities of all above abbreviations using simple rules, for example, *ABC* refers to *All Basotho Convention*, the first *ACU* refers to *Anti-Corruption Unit* and the second *ACU* refers to *Asian Clearing Union*.

In our NLPR_KBP system, we extract the entity candidate of abbreviation name through manually heuristic patterns, such as the Cap Pattern:

(Cap *?)$_1$\\(Abbr\\)$_2$➔1 is the entity Candidate of *Abbr*

2.2 Candidate Detection Using Wikipedia Anchor Dictionary

The second way to detect the entity candidate is to use the anchor dictionary of Wikipedia, which contains rich mention form information of entities. For example, in Figure 2, the three anchor texts of *IBM Company* are respectively its full name "International Business Machines", acronyms "IBM" and alternative name "Big Blue". Using the anchor text collection in Wikipedia, we can summarize the target entities of a mention form and the count information it's used as the mention form of the target entity. Part of the mention form table is shown in Table 1.

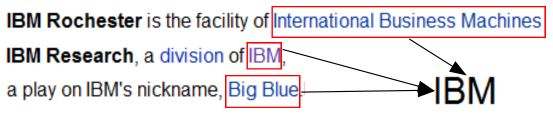

IBM Rochester is the facility of International Business Machines
IBM Research, a division of IBM,
a play on IBM's nickname, Big Blue.
IBM

Figure 2. Three anchor texts of *IBM*

Mention Form	**Target Entity**	**Count**
IBM	IBM	3685
	IBM mainframe	2
	IBM DB2	2
	…	…
International Business Machine	IBM	1
AI	Artificial intelligence	581
	Game artificial intelligence	48
	Ai (singer)	10
	Angel Investigations	9
	Strong AI	3
	Characters in the Halo series	2
	…	…

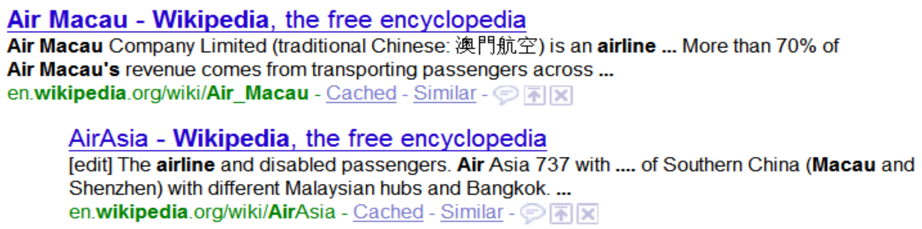
Table 1. Part of the mention form table of Wikipedia entities

Using the mention form table, we can easily identify the entity candidates of an entity mention through finding them in table.

2.3 Candidate Detection Using Web

Even through the above two ways, we still cannot detect all the entity candidates of an entity mention, so we try to leverage the whole web information for detecting the candidates through web search.

Given an entity mention together with their context, such as "*Macau's fledgling airline, **Air Macau***", we submit it to the Google API and retrieve only the web pages within Wikipedia (the knowledge base only contains the entities within Wikipedia), for example, the search result of the above mention will be as shown in Figure 3.



Air Macau - Wikipedia, the free encyclopedia
Air Macau Company Limited (traditional Chinese: 澳門航空) is an airline ... More than 70% of Air Macau's revenue comes from transporting passengers across ...
en.wikipedia.org/wiki/Air_Macau - Cached - Similar

AirAsia - Wikipedia, the free encyclopedia
[edit] The airline and disabled passengers. Air Asia 737 with .... of Southern China (Macau and Shenzhen) with different Malaysian hubs and Bangkok. ...
en.wikipedia.org/wiki/AirAsia - Cached - Similar

Figure 3. The Google Search Results of Target Candidate

From the search result, we can detect the *Air Macau* and the *AirAsia* as the entity candidate of *Air Macau*.

3 The Entity Linker

Given the detected entity candidates $\boldsymbol{E = \{e_1, e_2, \ldots, e_n\}}$ of an Entity mention $\boldsymbol{m}$, we then link the Entity *m* with the entity it refers to. For example, given the following two entity *ABC* mentions:

$\mathbf{m_1}$: ABC could not directly access the underlying file system and operating system.

$\mathbf{m_2}$: In American, ABC first broadcast on television in 1948.

and the three entity candidates:

$\mathbf{e_1}$: American Broadcasting Company: a private US broadcaster.

$\mathbf{e_2}$: Associated British Corporation: a former British film and television company

$\mathbf{e_3}$: ABC(programming language)

The entity linker should link the first entity mention $\mathbf{m_1}$ with $\mathbf{e_3}$, while it links the second entity mention $\boldsymbol{m_2}$ with Entity $\boldsymbol{e_1}$. In NLPR_KBP system, the entity linker links an entity mention with the most similar entity candidate by measuring the similarity between them. Specifically, the Entity Linker works as follows: 1) Measuring the similarity between a mention and the entity candidates of the mention; 2) linking the entity mention with the most similar entity candidate. In following we respectively describe each step in detail.

3.1 Measuring the Similarity between Entity Mention and Entity Candidate

The main component of our entity linker is a similarity measure between entity mention and entity candidate. Based on the computed similarities, our Entity linker can link each entity mention with the most similar entity candidate. For example, if the similarity measure assigns a larger similarity to ($\boldsymbol{m_1}$, $\boldsymbol{e_3}$) than ($\boldsymbol{m_1}$, $\boldsymbol{e_1}$) and ($\boldsymbol{m_1}$, $\boldsymbol{e_3}$), then the entity linker can successfully link the $\boldsymbol{m_1}$ with $\boldsymbol{e_1}$.

There were two types of similarity we take into consideration: 1) the similarity based on the *bag of words* (BOW) model, which can capture the word co-occurrence information; 2) the similarity using the Wikipedia semantic knowledge, which can capture the semantic relation information. In following we introduce how to compute these two types of similarity and how to combine these two similarities into a hybrid similarity.

The BOW Based Similarity. Using the *BOW* model, both the entity mention and the entity candidate are represented as a Vector of word features, and each word is weighted using the standard TFIDF measure. In NLPR_KBP system, the word features are extracted by tokenizing the text using the OpenNLP Tokenizer[1], with all stop words are filtered. Then, given the vector representation of entity $e = \{w_1, w_2, ..., w_n\}$ and the vector representation of entity mention $m = \{w_1^{'}, w_2^{'}, ..., w_n^{'}\}$, the similarity between them is computed as the cosine similarity between *e* and *m*:

$$SIM_{BOW}(e,m) = \frac{\sum_i w_i w_i^{'}}{\sqrt{\sum_i w_i^2}\sqrt{\sum_i (w_i^{'})^2}}$$

The Wikipedia Semantic Knowledge Based Similarity. The BOW based similarity can only capture the word co-occurrence information, which is usually not enough for Entity linking task. For example, the Entity Mention $\boldsymbol{m_1}$ and the entity candidate $\boldsymbol{e_3}$ have no co-occurrence words, so the BOW based similarity will unable to linking them together. To overcome this problem, we compute another similarity between entity mention and entity candidate based on the semantic relations in Wikipedia. For example, as shown in Figure 4, when computing the similarity between $\boldsymbol{m_1}$ and $\boldsymbol{e_3}$, we also take the semantic relation between *file system* and *programming language*, and the semantic relation between *operating system* and *programming language* into consideration.

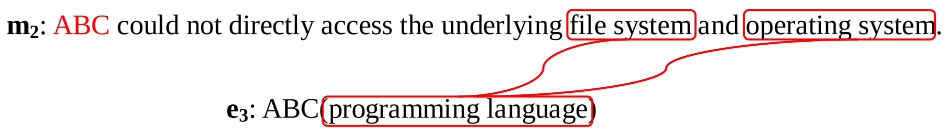

$\mathbf{m_2}$: ABC could not directly access the underlying file system and operating system.
$\mathbf{e_3}$: ABC(programming language)

Figure 4. The semantic relations between Wikipedia concepts

Specifically, the Wikipedia similarity is computed as follows:

1) **Wikipedia concept detection**. The appearances of Wikipedia concepts are detected using the method describe in Milne and Witten [2]. Then, the entity mention and the entity candidate are represented as a vector of Wikipedia concepts $\{c_1, c_2, ..., c_m\}$. For example, the Entity Mention $\mathbf{m_2}$ and the Entity $\mathbf{e_3}$ is represented as follows:
   $\mathbf{m_2}$ = { *file system*, *operating system*},
   $\mathbf{e_2}$ = { *programming language* }.
2) **Wikipedia concept weighting**. After the first step, the entity mention and the entity candidate are represented as Wikipedia concept vectors. However, not all the concepts in representation are equally helpful, so this paper selects the helpful concepts by assigning each concept with a weight which indicating its relatedness to the entity mention or the entity candidate. In detail, for each concept *c* in representation, we assign it a weight by averaging the semantic relatedness of *c* to all other concepts Wikipedia concept vector, i.e.:

$$w(c,e) = |e|^{-1} \left(\sum_{c_i \in e, c_i \neq c} sr(c, c_i) \right),$$

where *sr(c, c_i)* is the semantic relatedness measure between two concepts *c* and *c_i*, which is computed as the same as described in Milne and Witten [3].

3) **Computing Similarity**. Then, the Wikipedia similarity is computed as:

$$SIM_{wiki}(e,m) = \frac{\sum_{c_i \in m} \sum_{c_j \in e} w(c_i, m) \times w(c_j, e) \times sr(c_i, c_j)}{\sum_{c_i \in m} \sum_{c_j \in e} w(c_i, m) \times w(c_j, e)}$$

which is the weighted average of all semantic relatedness between the entity mention ***m*** and the entity candiate ***e***.

The Hybrid Similarity. As shown above, the BOW measure can capture the word co-occurrence information, while the Wikipedia similarity can capture the semantic relations between concepts. So we can derive a hybrid similarity which could combine both the above similarities for achieving better entity linking performance. Specifically, the hybrid similarity is computed as:

$$SIM_{Hybrid}(e,m) = \lambda \times SIM_{BOW}(e,m) + (1-\lambda) \times SIM_{wiki}(e,m)$$

3.2 Entity Linking

Given the computed similarity, the entity linking process is trivial: we select the entity candidate with the largest similarity to the entity mention. There is only one problem unresolved: under what condition an entity mention is linked to the NIL entity. In our NLPR_KBP system, we use a NIL threshold *T* to determine which entity mention is linked to NIL: that is, an entity mention is linked to the NIL Entity if the similarities between the entity mention and the entity candidates of it are all smaller than *T*. Now, the entity linking formula can be written as:

$$e = \begin{cases} \arg\max_{e_i} \; SIM_{Hybrid}(m, e_i), & where\ SIM_{Hybrid}(m, e_i) > T \\ NIL, & other\ wise \end{cases}$$

4 Submission and Results

TAC 2009 Entity Linking task contains 3904 topics. For each submission, Micro-average and Macro-average are used to evaluate the performance of Entity Linking system. We submitted three automatic runs with different parameter settings. The description of our runs is shown in Table 2. The results of our runs are shown in Table 3.

Run	**Description**
NLPR_KBP1	**Similarity**: 0.4 SIM_{BOW} + 0.6 SIM_{wiki}; **NIL Similarity Threshold**: 0.4
NLPR_KBP2	**Similarity**: SIM_{wiki}; **NIL Similarity Threshold**: 0.44
NLPR_KBP3	**Similarity**: SIM_{wiki}; **NIL Similarity Threshold**: 0.0

Table 2. Description of Our Runs

Runs	Micro-averages			Macro-averages		
	3904 queries	1675 non-NIL	2229 NIL	560 entities	182 non-NIL	378 NIL
NLPR_KBP1	**0.7672**	0.6925	0.8232	0.7281	0.6485	0.7669
NLPR_KBP2	**0.7585**	0.6358	0.8506	0.7347	0.5956	0.8017
NLPR_KBP3	**0.7559**	0.6412	0.8421	0.7214	0.6002	0.7798

Table 3. Result of Our Runs

From results in Table 3, we could obtain the following conclusion:

1) Our system achieved competitive results: all the three runs achieved a Micro-average more than 0.75, which shows our NLPR_KBP system's efficiency.
2) The hybrid similarity can achieve the best performance, that is, both the BOW similarity and the Wikipedia similarity achieve worse performance than the combined similarity. This result meets our intuition: the combination of the word co-occurrence information and the semantic relation information will perform better than only using only one type of information as they are complementary of each other.
3) The performance of the NIL entity mentions and the non-NIL entity mentions are often positively correlated, that is, with a larger NIL similarity threshold, the 2229 NIL can usually achieves better performance, meanwhile the performance of the 1675 non-NIL is reduced.

5 Conclusion

This paper described our NLPR_KBP system in detail for TAC 2007 Entity linking task. Our system has two important components. The first important component is Multi-way Entity Candidate Detector. In this component, our system detects the entity candidate of an entity mention by leveraging the information in local document, the Wikipedia and the web. In the second important component, the Entity Linker, our system links an entity mention to the most similar entity candidate by measuring the similarity between the entity mention and all the entity candidates of it. Both the word co-occurrence information and the semantic relation in Wikipedia are employed in similarity measure. Our three submitted runs achieved competitive results. The evaluation of our three runs demonstrates the validity of two important components in our system.

6 Acknowledgements

The work is supported by the National High Technology Development 863 Program of China under Grants no. 2006AA01Z144, and the National Natural Science Foundation of China under Grants no. 60673042 and 60875041.

7 Reference

[1] http://opennlp.sourceforge.net/

[2] D. Milne, Ian H. Witten. Learning to Link with Wikipedia. In Proc. of CIKM, 2008.

[3] D. Milne and Ian H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In Proc. of AAAI, 2008.