HIT Approaches to Entity Linking at TAC 2011

Yuhang Guo, Guohua Tang, Wanxiang Che, Ting Liu*, Sheng Li
Research Center for Social Computing and Information Retrieval
MOE-Microsoft Key Laboratory of Natural Language Processing and Speech
School of Computer Science and Technology
Harbin Institute of Technology, China
{yhguo, ghtang, car, tliu*, sli}@ir.hit.edu.cn

Abstract

This paper describes the system of HIT at the 2011 Text Analysis Conference (TAC) Knowledge Base Population (KBP) track English Entity Linking task. Based on structured and unstructured information extracted from Wikipedia, this system predicts the most probable entity that a query mention might refer to. A similarity score is assigned to the candidate entity by computing the the relatedness between the query and the entity, and augmented by the popularity of the entity. We model the query context as a graph of the entities and utilize the referential relationship between the context entities and the candidate entities in Wikipedia to measure the relatedness between the query context and the candidate entity. Evaluation results show the performance of our system reaches the median value of all the participating systems.

1 Introduction

Research Center for Social Computing and Information Retrieval from Harbin Institute of Technology (HIT) participated in the Entity Linking task at the 2011 Text Analysis Conference (TAC) Knowledge Base Population (KBP) track. This paper describes the system we implemented.

Entity Linking is the task of linking a name mention in a document to the correspondence entity in a Knowledge Base (KB) (McNamee and Dang, 2009; Ji and Grishman, 2011). In the TAC-KBP track, the input of the entity linking is comprised of a KB and a query which contains a name string and the source document which the string appears in. The KB of 818,741 entities in this track is a subset of the Wikipedia entity collection. The output of the task is the correspondence entity id in the KB for the input query. If the entity is out of the KB, the system returns a unique NIL id of this entity.

2 Our Approach

We resolve the entity linking problem in three steps. First, we generate a set of candidate entities for each query name. Then we rank these candidates according to the similarities between the candidate and the query context in the source document. Finally, we discriminate those queries of out-of-KB entities.

2.1 Candidate Generation

In this step, we aim to collect all potential entities of the query name. Probably the most direct way is to retrieve the query name in the Wikipedia, and then harvest the entity with the name. However, many complex cases make this step need more sophisticated processing.

Some query names cannot be directly found in the Wikipedia not because the corresponding entities are not in it but because the query names are aliases or alternative names of the entities which are not included in the name field of the relevant pages. Wikipedia provides a redirect mechanism to link popular aliases or synonyms to the corresponding pages. For example, the page titled with *Robert Gates* could be found through the redirect page of the alias *Bob Gates*.

Redirect pages cover most popular aliases. However there are still many names which could not be recalled in that way. We mine other aliases from fol-

lowing sources and map them to the corresponding entities. Here we use the term "alias" to represent all other names of the entity except for the article title of the entity.

In some Wikipedia articles, structured information is organized with **Infobox** template in attribute-value pair format. We extract the values of the attribute "fullname" or "nickname" in the Infobox to supplement the alias set of the entity of this article.

In the first paragraph of Wikipedia article, the name/names of the entity this article describes is/are usually highlighted in bold format. So we extract these **bold texts** as the aliases of the entity.

Wikipedia contains plenty of cross references in the form of hyperlink. The **hyperlink anchor texts** can be different from the name of the target pages. We collect these anchor texts as the aliases of the corresponding target entities.

In Wikipedia, if a name is shared by several entities these entities are usually listed in a **disambiguation page** of this name. We augment the alias set for each listed entities with this name (after removing the *(disambiguation)* suffix if it contains).

For the acronym query names, we try to find its full name coreference in the source document with patterns:

If the acronym is bracketed, we extract the name phrase immediately before the capitalized letter nearby (e.g. ... *The Mexican Football Federation (FMF) on Monday* ...).

If the acronym is followed by a bracket, we extract the phrase in the bracket (e.g. ... *From the PRC (People's Republic of China) we get much benefit.* ...).

Or else, we just find the phrase in the context with the same capitalized letter as the acronym (e.g. *ABC* $\rightarrow$ ... *he told the Australian Broadcasting Corporation.* ...).

When the full name is found, we use this full name to generate the candidates instead of the original query name.

In addition, we index the title and the text field of the track KB with lucene. Then we search the query name in these fields and select the top N (N set to be 40 in our system) returned entities as candidates.

In our system, we employed an open-source Java-based Wikipedia API (Zesch et al., 2008) to extract the Wikipedia texts.

2.2 Candidate Ranking

After the candidate generation step, nearly every query got more than one candidates. In this step, we need to identify which candidate is most likely to be the referent entity of the query. We rank the candidates by their similarities to the query. In order to build this similarity function, we extract multiple features from Wikipedia and the query. Then we use a heuristic method to combine these features into a reference score.

2.2.1 Features

We organize the features that used for the candidate ranking into three groups and summarize them in Table 1.

The first group is the surface features. This group of features focus on the literal similarity between the query and the candidate name. The similarity is measured in several methods, such as the inclusiveness of the query name and the candidate name, whether their double metephone values are equal or not and the Dice coefficient and edit distance between the two strings. The Link-Count feature gives the frequency of the candidate is linked from the query name as a hyperlink anchor text in Wikipedia. The normalized form of the Link-Count feature, Link-Prob shows the proportion of the query name is linked to the candidate.

Feature SCont-Dice-Coef, SCont-Comm-Word-Count and SCont-Jaccard-Sim extend the query name to a longer name, or short context. Context words around the query name are extracted within a window size (set to 20 characters). These features are especially helpful to some "Geo-Political Entities" in newswire corpus. For example, the query name *WESTLAKE* is followed by term *Louisiana* in query *EL_02168*. Obviously this context word provides a strong clue that the query name should be linked to the candidate *Westlake, Louisiana*. We utilize this characteristic by using the Dice coefficient between the short context string and the candidate name string, their word overlap and the Jaccard similarity of the two word sets.

The semantic features capture the semantic relatedness between the candidate entity and the query context. We identify the NE type of the query name in context by using a supervised classifier. In the TAC-KBP track, only three types of entities are in-

No.	Name	Value Type	Description
Surface			
1	Cand-in-QName	{0,1}	1 if the name of the candidate is the substring of the query name, otherwise 0
2	QName-in-Cand	{0,1}	1 if the query name is the substring of the name of the candidate, otherwise 0
3	QName-eq-Cand	{0,1}	1 if the query name equals to the name of the candidate, otherwise 0
4	QName-eq-Acro-Cand	{0,1}	1 if the query name equals to the acronym form of the candidate name, otherwise 0
5	Double-Mete-eq	{0,1}	1 if the double metephone value of the query name string equals the value of the candidate, otherwise 0
6	Dice-Coef	double	the Dice coefficient between the string of query name and the candidate
7	Edit-Dist	int	the edit distance between the string of the query name and the candidate
8	Link-Count	int	the frequency of the query name links to the candidate in Wikipedia
9	Link-Prob	double	the percent of the query name links to the candidate in Wikipedia
10	SCont-Dice-Coef	double	the Dice coefficient between the string of the short context of the query and the candidate
11	SCont-Comm-Word-Count	int	the number of the common words in the short context of the query and the candidate
12	SCont-Jaccard-Sim	double	the Jaccard similarity between the short context bag-of-words and the candidate bag-of-words
Semantic			
13	Type-eq	double	1 if the Named Entity (NE) type of the query name is identical to the class of the candidate, 0.3333 if the NE type of the candidate is unknown (UKN),otherwise 0
14	Cand-in-Cont	{0,1}	1 if the candidate entity is in the context, otherwise 0
15	Indegree	int	the indegree of the candidate node in the entity-link graph
16	Outdegree	int	the outdegree of the candidate node in the entity-link graph
Nonlinear Combination			
17	3*4	{0,1}	1 if the name of the candidate is identical to the query name and both of them are in acronym form, otherwise 0
18	9*13	double	Link-Prob if the NE type of the query and the candidate are identical, otherwise 0
19	14*15	int	Indegree if the candidate entity is in the context, otherwise 0

Table 1: Features for the entity linking

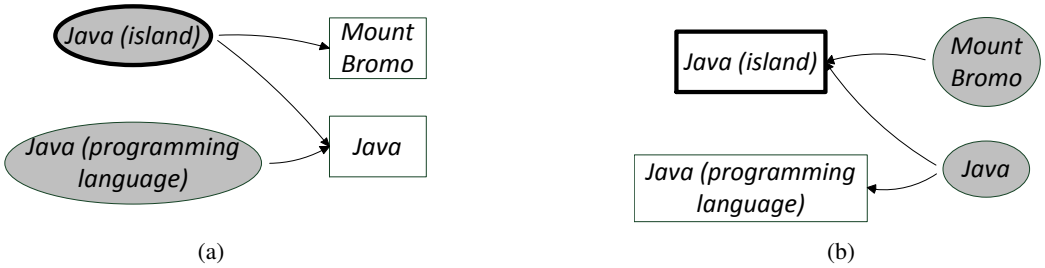

Figure 1: An example of the context graph.

volved which are Person (PER), Geo-Political Entity (GPE) and Organization (ORG). In the KB, some of the entities are annotated with these NE types and others are left with an unknown label (UKN). We set the Type-eq feature 1 if the NE type of the query equals to the candidate's NE type from the KB, 0.3333 if the type of the candidate is UKN, or otherwise 0. We learn an SVM classifier using the NE type labeled data from TAC-KBP 2009 and TAC-KBP 2010. The NE feature set includes bag-of-words, part-of-speech and part-of-speech-bigram.

In our system, we model the query as a graph of entities. We extract all segments in the source document of the query which match the longest titles in Wikipedia. The corresponding articles of these titles are represented as context nodes in the graph. All candidates are also added into the graph as candidate nodes. If a context node's title is contained in a candidate's article in Wikipedia, we draw a directed edge from the candidate node to the context node. On the other hand, if a candidate node's title is contained in a context node's article, we draw a directed edge from the context node to the candidate node. We utilize three features from this graph: Cand-in-Cont, Indegree and Outdegree of the candidate. We set Cand-in-Cont feature as 1 if the candidate is also included in the context node set, otherwise 0.

For example, given a query name *Java* and its source document: *Mount Bromo is one of Java's most popular tourist attractions*, we model this context as the graph shown in Figure 1. The referent entity, *Java (island)* has more out links and in links than the other candidate *Java (programming language)* in the graph.

The third group of features are the nonlinear combination of some of the above features. These features highlight the candidates with higher probability to be the answers in specific conditions. Such as the Link-Prob if the NE type of the candidate equals to the query.

2.2.2 Ranking Method

Candidate entities of each query are represented as feature vectors. The values of each dimension are the corresponding similarity/relatedness scores. These scores are then linearly combined into one score, which we use to rank the candidates by descending order. The weights of each features are assigned manually based on the TAC-KBP 2009 and TAC-KBP 2010 data sets.

2.3 NIL Labeling

Some of the entities for the query are out of the track KB. We label NIL to the queries for which the score of the top ranked candidate is under a threshold. In TAC-KBP 2011, queries need to be clustered and labeled with KB ID or NIL ID. The NIL ID should start with "NIL" and be suffixed with an identifier of the cluster.

We implemented a simple labeling method based on following rules:

If the score of the top candidate is higher than the threshold and the top candidate is in the KB, then label the query with the KB ID of the candidate.

If the query is an acronym, then suffixes the NIL mark with the full form of the acronym.

Or else if the candidate set is empty, then suffixes NIL with the query name.

Otherwise suffixes NIL with the name of the top score candidate.

3 Results

In the TAC-KBP 2011 Entity Linking track, systems are evaluated by B-Cubed+ precision, recall and F1

Runs	P	R	F1
Highest	-	-	0.846
Median	-	-	0.716
HIT	0.723	0.709	0.716

Table 2: The highest, median and our entity linking results

score[1] (F1 score is the official score). 21 teams submitted 44 runs in total. The results of the highest, median and our run are listed in Table 2.

4 Conclusion

In this paper, we describe our entity linking system for TAC-KBP 2011. We extract both surface and semantic features from Wikipedia and query context and use a linear model to combine them. For the NIL processing we leverage a simple heuristic method. Evaluation results show that the system performs median in the track.

Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) via grant 60803093, 60975055, 61073126, 61133012 Natural Scientific Research Innovation Foundation in Harbin Institute of Technology (HIT.NSRIF.2009069), and Fundamental Research Funds for the Central Universities (HIT.KLOF.2010064).

References

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1148–1158, Portland, Oregon, USA, June. Association for Computational Linguistics.

P. McNamee and H.T. Dang. 2009. Overview of the tac 2009 knowledge base population track. In *Proceedings of the Second Text Analysis Conference (TAC2009)*.

Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*.

[1]See http://nlp.cs.qc.cuny.edu/kbp/2011/scoring.html for the metric details.