MSRA at TAC 2011: Entity Linking

Yunbo Cao **Chin-Yew Lin**
Microsoft Research Asia
{yunbo.cao,cyl}@microsoft.com

Guoqing Zheng
Shanghai Jiao Tong University
gqzheng@apex.sjtu.edu.cn

1 Introduction

The Knowledge Base Population task aims at advancing the state of the art for systems that automatically discover information about named entities and then incorporate this information in a knowledge source. The overall task of populating a knowledge base is decomposed into two related tasks: Entity Linking, where names must be aligned to entities in the KB, and Slot Filling, which involves mining information about entities from text. We participated in the first task.

The entity linking task requires either linking entity mentions in the documents to entries in the Knowledge Base (KB) or highlighting these mentions as non-KB (NIL) entries. The task includes particularly difficult cases like ambiguous mentions (e.g George Bush), aliases (e.g Angela Kasner, more known as Angela Merkel) or even examples of both (e.g ABC). In addition, in order to create new KB entries, the task further requires the participating systems to group *NIL* mentions referring to the same entities together.

As has been done in most entity linking systems, our approach consists of two steps: Candidate generation and candidate disambiguation. For candidate generation, we make use of a recall-oriented retrieval model; and for candidate disambiguation, we treat it as a supervised learning problem. As for the details of grouping the *NIL* mentions, please refer to (Zhang et al., 2011).

We jointly participate in KBP 2011 with the I2I-NUS team. However, in this paper, we only report the results of the MSRA submissions. The joint submission is reported in (Zhang et al., 2011). In our participation, we are interested in answering the following question: Can an retrieval-based method do a good job for candidate generation? If the answer is YES, we can then leverage many research results with the area of information retrieval.

2 Our Approach

To link the mentions with the entries in KB or *NIL* for non-KB queries, we perform the following two steps:

2.1 Candidate Generation

A Knowledge Base usually contains millions of entries (each of which represents one entity). Therefore, a component capable of efficiently generating a manageable candidate list is essential for an entity linking system. As a first step of the entity linking system, the goal is to boost the recall as much as possible. Therefore, we refer to the component as 'recall-boosted candidate generation'.

Existing systems address candidate generation mostly by exploring name variants around title strings (Dredze et al., 2010; McNamee et al., 2009). However, as some query mentions are orthographically different from the titles of their referents in the KB, it may cause failures in the name-string-based candidate generation. Therefore the context should be considered at this early stage in case the name matching fails. We propose to augment the name-based candidate generation by a number of recall-boosting features. Specifically, we adopt a retrieval model and index the KB fields *title*, *article*, and *info box* by words, which are searched against by the query fields *name*, *acronym/context-document*, *name/context-sentence*. And we index the KB fields *title* and *acronym of title* by characters, which are searched against by the query fields *name* and *acronym*. The *title*s are augmented by their known aliases.

The above proposed retrieval based candidate generation aims to achieve the recall-boosting goal by employing the large number of attributes as the searchable fields. Moreover, pre-indexing the fields ensures high efficiency at the same time.

2.2 Candidate Disambiguation

The disambiguation problem can be formally defined as follows. Given the input and output spaces by $\mathcal{X}$ and $\mathcal{Y}$, where $\mathbf{x} \in \mathcal{X}$ represents a query and $\mathbf{y} \in \mathcal{Y}$ represents one of the candidate KB entries that are suggested by the candidate generation, the disambiguation task is formulated as to learn a hypothesis function $h : \mathcal{X} \rightarrow \mathcal{Y}$ to predict a $\mathbf{y}$ for a given $\mathbf{x}$.

Particularly, given a training set $S = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \in \mathcal{X} \times \mathcal{Y} : i = 1, \ldots, N\}$, we learn hypothesis functions that take the form $h(\mathbf{x}; \mathbf{w}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \mathcal{F}(\mathbf{x}, \mathbf{y}; \mathbf{w})$. Here $\mathcal{F} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{R}$ is the discriminant function where $\mathcal{F}(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y})$. $\Psi(\mathbf{x}, \mathbf{y})$ denote the feature functions dependent on a query $\mathbf{x}$ and a candidate entity $\mathbf{y}$. The features encapsulate the name-based and context-based attributes matching of $\mathbf{x}$ and $\mathbf{y}$ as will be detailed in next subsection.

Learning Model:

We adopt a binary classification, by which we are to determine whether or not a candidate is referred to by the query. If none of the candidates is considered as the referent of the query, the query will be labeled as *NIL*. Otherwise (more than one candidates are considered related), we will choose as the prediction the candidate with the highest classification confidence.

In practice, we make use of SVM as our learning methods for classification.

Learning Features:

The features used by our disambiguation models are summarized in Table 1, categorized in the name-based and the context-based sets. We also use some additional features that cross the two categories, including (1) the number of times the query mention appears in the entity context, (2) the number of times the entity title appears in the query document, and (3) the rank and the raw score of the entity passed from the candidate generation step of the query. Generally, we see these features as context-based features.

3 Experiments

3.1 Experiment Setup

The training data of KBP 2011 for entity linking has 3,904 queries in Eval-09 set and 2,250 queries in Eval-10 set, across three named entity types: Person, Geo-Political Entity and Organization.

The scoring metric used in KBP 2011 to evaluate entity linking system is $B\text{-}Cubed^{+}$.

3.2 Submissions and Results

MSRA submitted two results to KBP 2011. MSRA1 made use of only Eval-09 for its model training and MSRA2 used both Eval-09 and Eval-10. The results are reported in Table 2. Our submissions perform comparably with the median.

System	Acc.	Precision	Recall	F1
MSRA1	0.745	0.695	0.723	0.709
MSRA2	0.739	0.690	0.714	0.702
Highest	-	-	-	0.846
Median	-	-	-	0.716

Table 2: Entity Linking submission scores

4 Conclusion

In this paper we reported our participation in KBP 2011. In the participation, we addressed the entity linking problem by a classification model, in which we introduced the use of a recall-oriented model for candidate generation. Our system achieves a F1 of 0.709.

References

R Bunescu and M Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, pages 9–16, Trento, Italy.

M Dredze, P McNamee, D Rao, A Gerber, and T Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Con-*

Attribute	**Feature Description**
***Name**-based* attributes and features	
Name or Alias	NN1: Exact match bet. query name and entity name; NN2: String similarity (longest common subsequence, Dice, edit distance) bet. query/entity name; NN3: Query/entity name contained in the other; NN4: Character unigram and bigram bet. query/entity name strings.
Acronym	NA1: Acronym exact match; NA2: String similarity bet. the capitalized characters in their original order in the two names.
***Context**-based* attributes and features	
Text context	CT1: Cosine similarity between the TF-IDF vectors of the query/entity article bodies; CT2: Cosine similarity between the TF-IDF vectors of the sentence containing the query mention and the first paragraph of the entity article.
Semantic context	CS1: Similarity between the two context articles' term vectors augmented by the category tags CS2: the Wikipedia taxonomy, as used in (Bunescu and Pasca, 2006); CS3: Type(ORG, PER, GPE, etc) match.
Attribute context	CA1: Overlap bet. the sets of countries extracted from the query document and the entity article; CA2: Overlap bet. the sets of time symbols extracted from the query document and the entity article; CA3: Overlap bet. the sets of person names extracted from the query document and the entity article; CA4: Overlap bet. the sets of NEs extracted from the two text contexts.
Social context	CS1: The rank of the entity's Wikipedia page in a web search engine's result for the query; CS2: The Wikipedia hyperlink graph indegree and outdegree of the candidate entity.

Table 1: Attributes of queries and KB entries, and the corresponding features

ference on Computational Linguistics, pages 76–82, Beijing,China.

P McNamee, M Dredze, A Gerber, N Garera, T Finin, J Mayfield, C Piatko, D Rao, D Yarowsky, and M Dreyer. 2009. Hltcoe approaches to knowledge base population at tac 2009. In *Proceedings of Test Analysis Conference 2009 (TAC09)*.

W Zhang, J Su, Chen B, Wang W, Toh Z, Sim Y, Cao Y, Lin CY, and CL Tan. 2011. I2r-nus-msra at tac 2011: Entity linking. In *KBP 2011*.