CMU System for Entity Discovery and Linking at TAC-KBP 2016

Xuezhe Ma[*], **Nicolas Fauceglia**[*], **Yiu-chang Lin**[*] and **Eduard Hovy**
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213, USA
{xuezhem, fauceglia, yiuchanl, hovy}@cs.cmu.edu

Abstract

This paper describes CMU's system for the Tri-lingual Entity Discovery and Linking (TEDL) task at TAC-KBP 2016. Our system is a unified graph-based approach which achieved competitive results for three languages.

1 Introduction

Typically, a EDL system is required to tackle three sub-tasks: (i) Entity Discovery – detecting mentions of entities appearing in a document; (ii) Entity Linking – linking each entity to the most suitable entry in a reference Knowledge Base (KB), and (iii) NIL Entity Clustering – clustering NIL mentions, which do not have corresponding KB entries.

The Tri-lingual Entity Discovery and Linking (TEDL) task at TAC-KBP 2016 extends the EDL task of 2015 from two perspectives. From the data perspective, TEDL targets at a larger scale data processing, by increasing the size of source collections from 500 documents to 90,000 documents. From the perspective of task design, TEDL individual nominal mentions are expanded to all entity types and all languages not only person nominal mentions for English.

This year's CMU TEDL system is largely based on the TEDL system from last year (Fauceglia et al., 2015). The major difference is that, in this year, we utilize Wikipedia as our Knowledge Base instead of Freebase. Our system first links all mentions to corresponding entities in Wikipedia. Then, at the output step, the system maps the Wikipedia entities back to Freebase by building a map between entities indexes from Wikipedia and Freebase.

Formally, our system for TEDL task consists of two main steps. First, we process the whole Wikipedia, representing it as a directed weighted graph, then computing semantic signature for each vertex (Section 2). We also need to do preprocessing for input data. Second, we build an end-to-end system for entity discovery and linking across three languages (Section 3). We use Babelfy[1] as the backbone of our system and extend it to be suited for the TEDL task. Briefly, our system is different from Babelfy in the following points:

- Our system uses the Wikipedia's Ontology directly, instead of merging WordNet into KB.
- For the construction of semantic signature, we use the algorithm of Personalized PageRank with node-dependent restart (Avrachenkov et al., 2014), instead of Random Walk with Restart (Tong et al., 2006) (see Section 2.1.3 for details).
- We modify the candidate extraction method and extend it to Chinese and Spanish.
- We introduce edge weights to semantic interpretation graph (Section 3.3).
- We propose a new rule-based entity type inference method (Section 3.4).

Our results show that our system obtains significant improvement on all the three languages, comparing with our system's performance in TEDL task at TAC-KBP 2015 (Section 4).

2 Data Preparation

In this section, we describe the preparation of data, including constructing the Wikipedia graph, computing Semantic Signatures, and preprocessing the input documents to transform them into Fragments.

[*] Equal contribution

[1] http://babelfy.org

2.1 Wikipedia

The reference knowledge base used in TEDL is a January 2015 snapshot of English Freebase, which includes about 81M nodes (mids) and 290M relations. However, as mentioned above, this year we utilize Wikipedia as our Knowledge Base to construct the Semantic Signatures. The snapshot of Wikipedia we used is the one on December 2015, which contains around 5M pages (nodes).

2.1.1 Preprocessing

We first use the WikiPrep toolkit [2] to preprocess the whole Wikipedia to extract all the text anchors of each page and to remove some irrelevant pages such that the ones for disambiguation. This yields a KB with around 4.9M pages.

2.1.2 Graph of Wikipedia

We first represent Wikipedia as a directed weighted graph, where the vertices in the graph are the entities and concepts in Wikipedia, there is an edge from vertex $v1$ to $v2$ if $v2$ appears in $v1$'s page as a text anchor. Following Moro et al. (2014), the weight of each edge is calculated as the number of triangles (cycles of length 3) that this edge belongs to. To implement the graph, we used the WebGraph framework (Boldi and Vigna, 2004).

2.1.3 Semantic Signature

A *semantic signature* is a set of highly related vertices for each concept or entity in Wikipedia graph. To calculate semantic signatures, we first compute the transition probability $P(v'|v)$ as the normalized weight of the edge:

$$P(v'|v) = \frac{w(v,v')}{\sum\limits_{v''\in V} w(v,v'')}$$

where $w(v',v)$ is the weight of the edge $(v \rightarrow v')$. With the transition probabilities, Semantic Signatures are computed using the algorithm of Personalized PageRank with node-dependent restart (Avrachenkov et al., 2014). It should be noted that the algorithm performed by Moro et al. (2014) to create semantic signatures is Random Walk with Restart (Tong et al., 2006), which is simulation of the Personalized PageRank algorithm used in our system. Finally, vertices with pagerank score higher than a threshold (η) are kept to build the semantic signature. In our system we set $\eta = 10^{-4}$.

2.2 Input File

For each language, two kinds of data, Newswire and Discussion Forum are given in *xml* format. As described in the task definition, every document is represented as a UTF-8 character array and begins with the <DOC> tag. The "<" character has index 0 and offsets are counted before XML tags are removed. Therefore, to preserve the offset for each sentence, a line-by-line file reader is implemented instead of using an *xml* file parser.

In Newswire data, the tags are relatively simple and clean compared to Discussion Forum. The news' headline and paragraphs are extracted between "<HEADLINE>, </HEADLINE>" and "<P>, </P>" tags, respectively. In discussion forum data, similarly, the headline and posts are obtained between "<headline>, </headline>" and "<post>, </post>" tags. The author whose linking result is always NIL of each post is detected at the same time. However, in each post, there might exist more than one *quote*, which are repetitive text from previous posts. Quote removal is therefore a followed-up step after post extraction. Moreover, any text that are between "&lt" and "&gt"tags or in *URL* format are removed from the post as well.

3 System Architecture

Our end-to-end EDL system includes entity mention detection (Section 3.1), candidate extraction (Section 3.2, entity linking (Section 3.3), type inference (Section 3.4), and NIL entity clustering (Section 3.5). We use the Stanford CoreNLP pipeline (Manning et al., 2014) for preliminary steps, and adapt and extend (Moro et al., 2014) for entity extraction and linking.

3.1 Entity Mention Detection

Different from last years system which extract all sequences of words with certain length limit and POS constraints as possible mentions, this year we use a pre-trained NER system to extract mentions. It significantly reduced the number of mentions and accelerated the linking algorithm. The NER system we used is the CRF-based statistical model implemented in Stanford CoreNLP system (Manning et al., 2014).

[2] http://www.cs.technion.ac.il/~gabr/resources/code/wikiprep/

3.2 Candidate Extraction

The task of the Candidate Extractor (CE) is, given an input string, return all the possible entities in the graph that could be associated with a substring of the input string. When processing the Freebase Dump, we keep an additional parallel data structure holding information about the names of the graph entities. For each Freebase entity we keep string labels provided by 3 predicates: name, label and alias. In the original Freebase Dump, string values have an associated language, so we only kept the values in our three languages. We implemented this name map as a Lucene index, that given a string returns all the nodes in our graph that have a label (name, label, alias) that contains the given string.

It is worth mentioning that for our multilingual task, this is the only part that deals with languages: we have different implementations of this component, one for each language. For English and Spanish, the approach is similar to the Babelfy implementation: perform POS tagging for each input sentence, and choose n-grams of length 1 to N (we used N = 5), that contain at least one NOUN, and which do not end or start in prepositions, conjunctions, punctuation, among others. For each one of these candidate fragments, we query the name index to retrieve all possible entities. For Chinese, the approach is completely different: we work at a character level, and we start with strings of N characters (we used N = 10) and search in the name index, and if there is no match, we search with N-1, and so on, until we have a match, and return each match as a Candidate Meaning.

Once the candidates have been extracted from the input document, the rest of the pipeline works in graph-space and does not depend on the input language. This makes it relatively easy to add a new language, provided Freebase has names for the new language.

3.3 Entity Linking

3.3.1 Semantic Interpretation Graph Construction

The semantic interpretation graph is constructed using a procedure similar to Moro et al. (2014). The difference is that we introduce edge weights to this graph – the weight of the edge between two vertices (v_1, f_1) and (v_2, f_2) is defined as the pagerank score between v_1 and v_2 in the Wikipedia graph.

	Type in Freebase
PER	people.person
GPE	location.country location.administrative_division location.statistical_region
ORG	organization.organization
LOC	location.location
FAC	architecture.structure

Table 1: Rules applied to distinguish between the 5 entity types.

3.3.2 Graph Densification

We implemented the graph densification algorithm presented in Moro et al. (2014), too. Basically, at each step of graph densification, we first find the most ambiguous mention, the one has the most number of candidate entities. Then we remove the least possible candidate entity from the most ambiguous mention, the one has smallest score. In our system, the score of a vertex (v, f) in the semantic interpretation graph is slightly different from the one in Babelfy – we use the sum of the incoming and outgoing edge weights instead of the sum of incoming and outgoing degree. Formally, the score of the vertex (v, f) is:

$$score((v, f)) = \frac{w(v, f) \cdot sum((v, f))}{\sum\limits_{(v', f)} w(v', f) \cdot sum((v', f))}$$

where $sum((v, f))$ is the sum of the incoming and outgoing edge weights of (v, f) and $w((v, f))$ is the number of fragments the candidate entity v connects to.

The above steps are repeated until every mention has less than a certain number (μ) of candidate entities. Finally, we link each mention f to the highest ranking candidate entity v^* if $score((v^*, f)) > \theta$, where θ is a fixed threshold.

3.4 Entity Type Inference

Before inferring the entity, our system first maps the Wikipedia entities back to Freebase via a map built before hand. Entity type is obtained from each entity's *Types* in Freebase. We define different rules to determine such entity types. If a candidate entity has the predefined types (2nd column in Table 1), its entity type is assigned as the corresponding value (1st column). Else, it is not treated as an entity and it is discarded.

	NER			Linking			Clustering		
	P	R	F1	P	R	F1	P	R	F1
	TEDL 2015								
Eng	50.9	56.0	53.3	42.1	46.3	44.1	49.1	54.0	51.5
Spa	60.2	60.8	60.5	47.3	47.7	47.5	54.1	54.5	54.3
Chn	50.0	61.4	55.1	44.5	54.7	49.1	48.9	60.1	53.9
	TEDL 2016								
Eng	81.4	49.1	61.3	72.4	43.0	54.5	77.5	46.7	58.3
Spa	76.5	51.6	61.6	69.6	47.0	56.1	74.6	50.4	60.1
Chn	67.1	47.3	55.5	58.1	40.9	48.0	65.6	46.2	54.3

Table 2: The offical results precision, recall and F1 measures over all three languages for our best run for three key metrics: strong typed mention match (NER), strong typed all match (Linking) and mention ceaf (Clustering) in TEDL at TAC-KBP 2016.

	NER			Linking			Clustering		
	P	R	F1	P	R	F1	P	R	F1
	TEDL 2015								
Eng	50.9	58.4	54.4	42.1	48.3	45.0	49.1	56.1	52.4
Spa	60.2	60.8	60.5	47.3	47.7	47.5	54.1	54.5	54.3
Chn	50.0	61.4	55.1	44.5	54.7	49.1	48.9	60.1	53.9
	TEDL 2016								
Eng	81.4	64.7	72.1	72.4	57.5	64.1	77.5	61.5	68.5
Spa	76.5	66.7	71.3	69.6	60.8	64.9	74.6	64.9	69.3
Chn	67.1	57.1	61.7	58.1	49.4	53.4	65.6	55.7	60.2

Table 3: The results exclude all nominal mentions.

3.5 NIL Entity Clustering

The final step in our system is clustering NIL entities. In our system, we simply merge candidates with exactly the same name spelling.

4 Experiments

We submitted one for TEDL task, in which we extract top 100 candidate entities for each mention ($K = 100$), and the ambiguous parameter $\eta = 10$.

Table 2 shows the results precision, recall and F1 measures over all three languages for our best run for three key metrics: strong typed mention match (NER), strong typed all match (Linking) and mention CEAF (Clustering), together with the results in last year's evaluation.

It should be noted that in this year, the percentage of NOM mentions is much more than last year, for which our system cannot do anything. The percentages of NOM mentions for the three languages are given in Table 4. To make a more reasonable comparison with last year's results, we exclude the nominal mentions from the evaluation. The results are provided in Table 3. We can see that comparing with last year's performance, our system achieved significant improvement on all the three languages.

	2016	2015
Eng	43%	10%
Spa	43%	0%
Chn	30%	0%
Total	38%	3.5%

Table 4: Percentages of nominal mentions of three languages in 2015 and 2016.

5 Conclusion and Future Work

We build a unified graph-based system for the TEDL task at TAC-KBP 2016. According to the official results, our system obtains significant improvement on all the three languages, comparing with our system's performance in TEDL task at TAC-KBP 2015

References

Konstantin Avrachenkov, Remco Van Der Hofstad, and Marina Sokol. 2014. Personalized pagerank with node-dependent restart. In *Algorithms and Models for the Web Graph*, pages 23–33. Springer.

Paolo Boldi and Sebastiano Vigna. 2004. The webgraph framework i: compression techniques. In *Proceedings of the 13th international conference on World Wide Web*, pages 595–602. ACM.

Nicolas R Fauceglia, Yiu-Chang Lin, Xuezhe Ma, and Eduard Hovy. 2015. Cmu system for entity discovery and linking at tac-kbp 2015. In *Proceedings of Text Analytics Conference (TAC 2015).*, November.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. In *Proceedings of ICDM 2006*. IEEE.