CMU System for Entity Discovery and Linking at TAC-KBP 2017

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 2017. Our system is a unified graph-based approach that achieves competitive results for English, Spanish, and Chinese.

1 Introduction

An EDL system must 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 mentions that do not have corresponding KB entries (so-called NILs) into coreference groups.

The Tri-lingual Entity Discovery and Linking (TEDL) task at TAC-KBP 2017 extends the 2015 EDL task from two perspectives. From the data perspective, TEDL targets larger-scale data processing, by increasing the size of source collections from 500 documents to 90,000 documents. From the task design perspective, TEDL individual nominal mentions are expanded from only person nominal mentions for English to all entity types and all three languages. (A nominal mention is an unnamed mention, such as "the president".)

CMU's TEDL system this year is largely based on our TEDL systems from the past two years (Fauceglia et al., 2015; Fauceglia et al., 2016). In the past, we treated nominal mentions independently using a separate additional component akin to an entity coreference resolution model. Its function was to link the nominal mention (such as "the President") to the KB entity representing the corresponding named mention(s) ("Barack Obama"). In this year's system, our EDL system linked named and nominal mentions to their corresponding entity entries in the KB simultaneously, by merging the mention embedding representations (named and nominal) within the densest subgraph liking algorithm (Moro et al., 2014).

Formally, our system for TEDL contains two main steps. First, we process the whole Wikipedia, representing it as a directed weighted graph and then computing a semantic signature for each vertex (Section 2). 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. Our system differs 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).
- We train a joint word and entity embeddings to handle nominal mentions.

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).

*Equal contribution

[1] http://babelfy.org

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.

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 from 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. An edge exists 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:

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

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

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 the Discussion Forum. The news' headlines 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. Authors whose linking result is always NIL for each post are 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 follow-up step after post extraction. Moreover, any text between "&lt" and "&gt"tags or in *URL* format is 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 year's system that extracted all sequences of words with certain length limit and POS constraints as possible mentions, this year we used a pre-trained NER system to extract mentions. This significantly reduced the number of

mentions and accelerated the linking algorithm. The NER system we used is the CRF-based statistical model implemented in the Stanford CoreNLP system (Manning et al., 2014).

3.2 Candidate Extraction

The task of candidate extraction is, given an entity mention, to return all the possible entities in the KB that could be associated with this mention. Consider the following example: *Donald Trump was born in New York. It is the most populous city in the United States.* We detect the following three named mentions: *Donald Trump*, *New York* and *the United States*, and one nominal mention: *the most populous city*.

For named mentions, we search the KB for candidate entities for which (one of) the names of the entity is a superstring of the text of the named mention. Thus, the candidates of mention *Donald Trump* include "Donald Trump" and "Donald Trump Jr.", the candidates of the mention *New York* include "New York City" and "New York State", and the ones of *the United States* include "the United States" and "the United States Air Force", etc.

For nominal mentions, however, we cannot directly extract candidate entities by name superstring matching, since the text of a nominal mention is general. To find possible candidate entities associated with a nominal mention, we first make an assumption that a nominal mention refers to a named mention in its near contexts. Based on this assumption, we search the candidate entities of nominal mentions only from the set of candidates of the named mentions within a window size of contexts. We compute the cosine similarities of the word embedding of the nominal mention's headword and the entity embeddings of candidate entities, filtering out those candidates whose similarities are below a threshold β. In the above example, we first initialize a set of candidates for the mention *the most populous city* with the union of the candidates of the three named mentions. Then, we use our joint embeddings to filter out candidates that are not coherent with the headword "city", retaining the associated candidates, including the correct one "New York City".

With the candidate extractor collaborating with our joint embeddings of words and entities, we can handle named and nominal mentions for entity linking simultaneously.

	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 among the 5 entity types.

It is worth mentioning that in our EDL system, this is the only part that deals with language per se: we have to implement different candidate extractors, one for each language. Once the candidates have been extracted, the entity liking component works in KB space, which is independent with the input language. This makes our EDL system easily adapted to new languages.

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.

3.3.2 Graph Densification

We implemented the graph densification algorithm presented in Moro et al. (2014). Basically, at each step of graph densification, we first find the most ambiguous mention (the one with the most candidate entities). Then we remove the candidate entity from the most ambiguous mention that has the smallest score. In our system, the score of a vertex (v, f) in the semantic interpretation graph is slightly different from the one in Babelf – 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

	NER			Linking			Clustering		
	P	R	F1	P	R	F1	P	R	F1
Eng	79.3	52.3	63.0	69.0	45.5	54.8	68.9	45.4	54.7
Spa	78.0	45.9	57.8	70.8	41.7	52.5	67.5	39.7	50.0
Chn	72.2	43.6	54.4	62.4	37.7	47.0	66.4	40.1	50.9

Table 2: The official 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 2017.

	NER			Linking			Clustering		
	P	R	F1	P	R	F1	P	R	F1
	Named Mention								
Eng	81.4	64.5	72.0	72.8	57.8	64.4	72.6	57.5	64.2
Spa	78.0	62.5	69.4	70.8	56.7	63.0	67.5	54.0	60.0
Chn	72.2	56.6	63.4	62.4	48.9	54.8	66.4	52.0	58.3
	Nominal Mention								
Eng	60.4	15.6	24.8	33.0	8.5	13.5	42.5	11.0	17.4
Spa	–	–	–	–	–	–	–	–	–
Chn	–	–	–	–	–	–	–	–	–

Table 3: The results on named and nominal mentions, respectively.

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 beforehand. 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 is discarded.

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 system for the TEDL task, in which we extract the top 100 candidate entities for each mention ($K = 100$), with the ambiguity 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).

Table 3 shows results of three languages on named and nominal mentions, respectively. It should be noted that our system can handle nominal mentions for English.

5 Conclusion and Future Work

We build a unified graph-based system for the TEDL task at TAC-KBP 2017. We developed an embedding-based method to train joint word and entity embeddings, and merged the embeddings with the densest subgraph linking algorithm to develop an EDL system that can link named and nominal mentions simultaneously, an advance over our system in TEDL task at TAC-KBP 2016.

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.

Nicolas R Fauceglia, Yiu-Chang Lin, Xuezhe Ma, and Eduard Hovy. 2016. Cmu system for entity discovery and linking at tac-kbp 2016. In *Proceedings of Text Analytics Conference (TAC 2016).*, 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.