BUPTTeam Participation at TAC 2016 Knowledge Base Population

Yongmei Tan, Xiaoguang Li and Di zheng

Center for Intelligence Science and Technology and Technology
Beijing University of Posts and Telecommunications
Beijing, China
{ymtan, xgli, zhdi}@bupt.edu.cn

Abstract

The Entity Discovery and Linking (EDL) track at NIST TAC-KBP2016 aims to extract named entity mentions from a source collection of textual documents in multiple languages (English, Chinese and Spanish), and link them to an existing Knowledge Base (KB). In this paper, we describe the BUPTTeam's system that participated in this track. The system consists of six components: 1) preprocessing; 2) mention recognition; 3) mention expansion; 4) candidates generation; 5) candidates ranking; 6) clustering. We describe our underlying approach, which relates to our previous work, and describe the novel aspects of the system in more detail.

1 Introduction

The goal of EDL track at Text Analysis Conference (TAC) 2016 is to automatically discover entity mentions from three languages (English, Chinese and Spanish) raw texts and link them to an entity from knowledge base, and cluster NIL mentions across languages.

Compared to the KBP2015 EDL task, the main differences are concluded as the follows:

- Target at a larger scale data processing, by increasing the size of source collections from 500 documents to 90,000 documents.
- Individual nominal mention is extended to five entity types (PER, ORG, GPE, LOC and FAC) and three languages (Chinese, English and Spanish).

In this paper, we present our system which builds on the elements of the system described in (Tan et al., 2015). Our contributions are summarized as follows:

- We use a semantic representation for entities and mentions using the stationary distribution through a random walk with restart on a mention-entity graph.
- We use heuristic grammatical rules to discover nominal mentions.
- We construct a word list to solve provincial and national abbreviations.

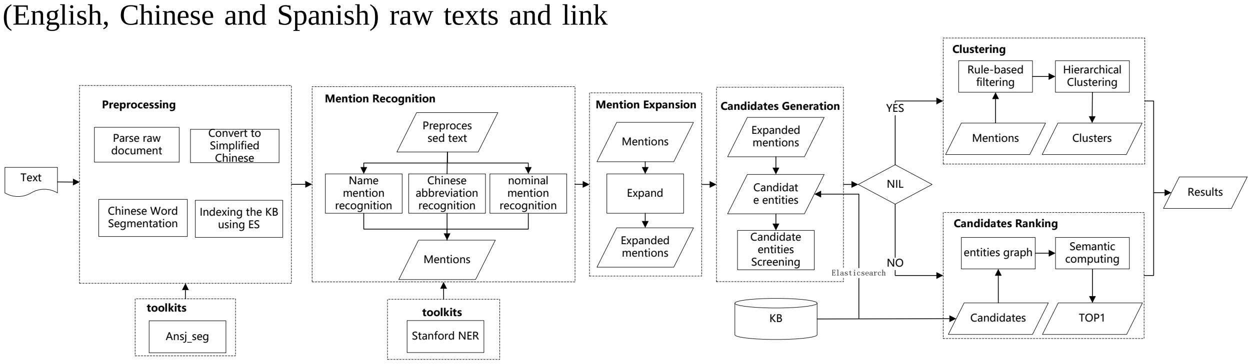

Figure 1: System Architecture

2 System Architecture

The architecture of our EDL system is described as Figure 1. It includes the following six components.

1) Preprocessing
2) Mention recognition
3) Mention expansion
4) Candidates generation
5) Candidates ranking
6) Clustering

2.1 Preprocessing

There are many xml tags in raw text, which influence mention recognition and the parts between “<quote>” and “</quote>” are also redundancy. So we remove these tags and parts.

There are many traditional Chinese words in raw texts and knowledge base. Text processing tools is good at processing simplified Chinese so that we convert traditional Chinese into simplified Chinese.

We use Ansj_seg [1] for Chinese word segmentation and Elasticsearch[2] for indexing the KB described in (Tan et al., 2015).

2.2 Mention Recognition

We use Stanford NER[3] to recognize most mentions.

In addition, mentions representing authors can be directly extracted from the raw texts. Their type is PER and linking results are always NIL.

Nominal mention is expanded to five entity types (PER, ORG, GPE, LOC and FAC) and three languages (Chinese, English and Spanish). This is a new challenge. We use some heuristic grammatical rules to recognize nominal mentions.

In Chinese, two or more abbreviations representing states or provinces are often wrote as a whole, such as: "中美", where "中" refers to “China”, "美" refers to “the United States”. This phenomenon influences the performance of mention recognition, and so we collect the word list of provincial and national abbreviations to recognize those mentions.

2.3 Mention Expansion

Sometimes mentions are nickname, alias, acronyms or part of their full names. We use some heuristic rules to expand these mentions into their surface forms by their context.

2.4 Candidates Generation

This step attempts to search potentially correct entities for mentions from Freebase. We generate a candidate set E_m for each mention m by Elasticsearch.

Too many candidates will make it hard to choose the right one. In order to scale the candidate set as small as possible, we filter the candidates according to some constraints.

2.5 Candidates Ranking

In most cases, the size of E_m is larger than one. Therefore, we rank the candidates and select the top one by the random walk with restart algorithm.

2.5.1 Mention-entity Graph Construction

1) Semantic Relation between Mention and Entity

The semantic relation $SR(m,e)$ between mention m and entity e can be computed as follows:

$$SR(m,e) = \frac{\vec{m}\cdot\vec{e}}{|\vec{m}||\vec{e}|} \tag{1}$$

Where m is represented as a vector $\vec{m}$ according to its context, and e is represented as a vector $\vec{e}$ by its text description in Freebase. All words are weighted by the tf-idf schema.

2) Semantic Relation between Entities

The semantic relation $R(e_i, e_j)$ between e_i and e_j can be calculated as follows:

$$R(e_i, e_j) = \frac{w_{ij}}{\sum_{e_k \in OUT(e_i)} w_{ik}} \tag{2}$$

Where $OUT(e_i)$ is the set of entities directly reachable from e_i and w_{ij} is the number of triples (entity e_i, relationship, entity e_j) in Freebase (Guo, 2014).

3) Mention-entity Graph

The mention-entity graph $G = (V, E)$ is derived from a text T and Freebase. It is a weighted and directed graph. The set V contains the mentions discovered from T and entities retrieved from Freebase. And E is the set of edges which can be divided into two categories: 1) mention-to-entity edge. There are always edges reaching candidate entity e from mention m; 2) entity-to-entity edge. If the triple (entity e_i, relationship, entity e_j) exists in Freebase, there is an edge from e_i to e_j. The

[1] https://github.com/NLPchina/ansj_seg
[2] https://www.elastic.co/
[3] http://nlp.stanford.edu/ner/

weights of mention-to-entity edge and entity-to-entity are separately computed by $SR(m, e)$ and $R(e_i, e_j)$.

The graph can capture mention-to-entity and entity-to-entity semantic relations. Sometimes it is too sparse to represent the global semantic coherence of a text. Therefore, we expand the graph by adding entities which are semantically related to more than one candidate. The process is illustrated by Figure 2.

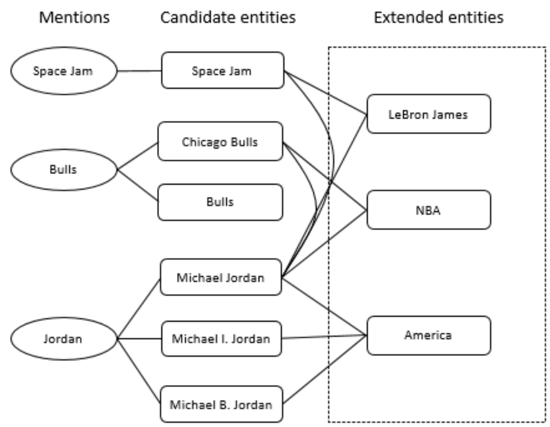

Figure 2: Expanded Graph

In Figure 2, the right dash rectangle shows extended entities that make the mention-entity graph dense and strongly connected.

2.5.2 Collective Entity Linking based on Random Walk with Restart

1) Random Walk with Restart

Random walk with restart is a stochastic process that iteratively travels the global structure of the graph with certain probability walking from one node to its neighbors. After reaching stability, the resulting probability distribution represents the relatedness between nodes in the graph.

The starting node is represented as an initial vector $\boldsymbol{s}$ with s_i referring to the probability of starting from entity e_i. Details about initialization of vector $\boldsymbol{s}$ will be described in the next section. After initialization of $\boldsymbol{s}$, we can perform the algorithm. The process of random walk with restart is illustrated by the following formulas.

$$r^0 = \boldsymbol{s} \tag{3}$$

$$r^{t+1} = (1-\alpha) \times r^t \times T + \alpha \times \boldsymbol{s} \tag{4}$$

Where r^t is the probability distribution at iteration t . Making $r^{t+1} = r^t$, stationary distribution can be calculated as follows:

$$r = \alpha(I - cT)^{-1}\boldsymbol{s},\ c = 1 - \alpha \tag{5}$$

Later we will use the stationary distribution as the semantic features of mentions or entities. Semantic features capture their relevance to others in the graph.

2) Semantic Feature of an Entity

In order to get semantic feature of an entity e_i, we need to let e_i be the starting node. This can be done by setting the initial vector $\boldsymbol{s}$ with $s_i = 1$, and $s_{j(j\neq i)}$=0.

3) Semantic Feature of a Mention

When computing semantic feature of mention m, the initial vector $\boldsymbol{s}$ can be calculated as follows:

$$s_i = \frac{SR(m, e_i)}{\sum_{e_k \in E_m} SR(m, e_k)} \tag{6}$$

Where s_i is the probability of starting random walk from e_i, E_m is the candidate set of m.

With the initial vector $\boldsymbol{s}$, the semantic feature of m can be computed by using a random walk with restart in the graph.

4）Semantic Relatedness

Let $SF(e_i)$ be the semantic feature of entity $e_i \in E_m$, and $SF(m)$ be the semantic feature of mention$m$. We use Hellinger distance to measure the difference of two probability distributions P and Q, which can be computed as follows:

$$H(P, Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{K}(\sqrt{p_i} + \sqrt{q_i})^2} \tag{7}$$

The semantic similarity $SS(m, e_i)$ between m and e_i is calculated as the following formula:

$$SS(m, e_i) = \frac{1}{H(SF(m), SF(e_i))} \tag{8}$$

5) Iterative Entity Linking Algorithm

Traditional collective entity linking methods link all mentions at the same time. Hence these methods rely heavily on the graph built from a text. Because mentions are always ambiguous, the initial graph brings in many noisy entities, resulting in a poor performance of the entity linking. In order to address this issue, we introduce an iterative entity linking algorithm which takes the linking results of previous iterations into consideration to prune irrelevant candidate entities and update weights of edges in the graph.

We take an easy-first strategy. If there is only one candidate for a given mention, we link the mention to the candidate. If there are more than two candidates for a given mention, we perform the following steps: (1) Use random walk with restart to obtain semantic features of mentions and entities; (2) Compute the semantic similarity measures between the mention and corresponding candidates; (3) Select the candidate with the highest score which exceeds a certain threshold. If the highest score is less than the threshold, NIL is assigned to the mention. If there are also mentions left to be linked, we use the previous linking results to update the graph, and preform the next iteration.

2.6 Clustering

If the candidate set E_m is empty, the linking result of mention m is NIL. We cluster the NIL mentions as the following two steps.

Firstly, NIL mentions are clustered by the strict rules:

1) All NIL mentions are divided into five types (PER, ORG, GPE, LOC and FAC);

2) If mention m_i and mention m_j meet any of the following conditions, we divide them into the same cluster:

- Mention m_i and mention m_j have the same surface string;
- Mention m_i is the prefixes or suffixes of mention m_j;
- Mention m_j is the prefixes or suffixes of mention m_i;

After the rough division, according to Harris's distributed hypothesis, if two words have similar context, their semantics are similar. We convert the mention's context into vector representation and use hierarchical clustering algorithm for clustering.

3 Results and Discussion

Table 1 lists the performance of NER and NER classification. The best result is in bold.

Table 1: The results of NER and classification of entity/mention type

	strong_typed_mention_match		
	P	R	F1
English	**0.866**	**0.602**	**0.71**
Chinese	0.81	0.609	0.695
Spanish	0.725	0.569	0.638
All	0.804	0.595	0.684

Table 2 describes the linking performance without NIL mentions. The best score is in bold.

Table 2: The performance of linking to the reference KB

	strong_all_match		
	P	R	F1
English	0.744	0.496	0.595
Chinese	**0.787**	**0.591**	**0.675**
Spanish	0.642	0.504	0.565
All	0.728	0.532	0.615

The performance NIL clustering is shown in the Table 3. The best score is in bold.

Table 3: The results of NER and clustering

	mention_ceaf		
	P	R	F1
English	0.817	0.521	0.636
Chinese	**0.821**	**0.617**	**0.704**
Spanish	0.694	0.545	0.611
All	0.757	0.553	0.639

Table 4 describes all kinds of evaluation measures on five mention types. The best result is in bold.

All kinds of evaluation measures on two different text genres are shown in Table 5.

Table 4: The results of NER, NER Classification, Linking and Clustering on the pre-defined Five Types

	strong_typed_mention_match			strong_all_match			mention_ceaf		
	P	R	F1	P	R	F1	P	R	F1
PER	0.865	0.709	0.779	0.732	0.599	0.659	0.72	0.59	0.648
ORG	0.545	0.364	0.437	0.416	0.277	0.333	0.499	0.333	0.4
LOC	0.578	0.225	0.324	0.516	0.2	0.289	0.58	0.225	0.325
GPE	**0.887**	**0.728**	**0.8**	**0.827**	**0.679**	**0.745**	**0.843**	**0.692**	**0.76**
FAC	0.353	0.017	0.032	0.324	0.015	0.029	0.353	0.017	0.032

Table 5: The results of NER, NER Classification, Linking and Clustering on the Different Text Genres

	strong_typed_mention_match	strong_all_match	mention_ceaf

	P	R	F1	P	R	F1	P	R	F1
NW	0.812	0.563	0.665	0.689	0.478	0.565	0.747	0.518	0.612
DF	0.798	0.629	0.703	0.75	0.591	0.661	0.758	0.598	0.669

4 Conclusions

We built a complete and robust system, including mention recognition, mention expansion, candidates generation, candidates ranking and clustering. In our work, we use the probability distribution resulting from a random walk with restart on a mention-entity graph to represent the semantics of entities and mentions. The semantic representation uses relevant entities from Freebase as features, thus reducing data sparseness.

References

Alhelbawy Ayman, and Robert Gaizauskas. Graph Ranking for Collective Named Entity Disambiguation. The 52nd Annual Meeting of the Association for Computational Linguistics (ACL2014), Maryland, USA, 2014.

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. SIGIR, 2009, pp. 267–274.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Furstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. EMNLP, 2011, pp. 782–792.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. SIGMOD, 2008, pp. 1247–1250.

Maolin Li. An Entity Linking Approach Based on Topic-Sensitive Random Walk with Restart. Acta Scientiarum Naturalium Universitatis Pekinensis, 2016, 52(1): 17-24.

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

Rudi Cilibrasi, and Paul Vitanyi. The Google Similarity Distance. Knowledge & Data Engineering IEEE Transactions on, 2007, 19(3): 370–383.

Silviu Cucerzan. TAC Entity Linking by Performing Full-document Entity Extraction and Disambiguation. In Proceedings of Text Analysis Conference, 2011.

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. LINDEN: Linking Named Entities with Knowledge Base via Semantic Knowledge. Proceeding of the 21st international conference on World Wide Web, 2012.

Wei Zhang, Yan Chuan Sim, Jian Su, and Chew Lim Tan. Entity Linking with Effective Acronym Expansion, Instance Selection and Topic Modeling. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, 2011.

Xianpei Han, Le Sun, and Jun Zhao. Collective entity linking in web text: a graph-based method. Proceedings of International Conference on Research & Development in Information Retrieval, 2011:765-774.

Yongmei Tan, Di Zheng, Maolin Li, and Xiaojie Wang. BUPTTeam Participation at TAC 2015 Knowledge Base Population. TAC, 2015.

Zhaochen Guo, and Denilson Barbosa. Robust Entity Linking via Random Walks. Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM, 2014.

Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Learning to link entities with knowledge base. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2010: 483-491.