The ZJU-EDL System for Entity Discovery and Linking at TAC KBP 2015

Hongliang Dai, Siliang Tang*, Fei Wu, Zewu Ma, Yueting Zhuang
College of Computer Science, Zhejiang University
{hldai, siliang, wufei, mazewu1102, yzhuang}@zju.edu.cn

Abstract

This paper describes a fully pipelined EDL system implemented by ZJU-DCD-EDL team for the TAC 2015 EDL (Entity Discovery and Linking) task. Our system mainly focuses on the linking of English name mentions. It is composed by mention extraction, candidate generation, candidate ranking, query expansion re-ranking, MLP re-ranking and NIL clustering modules. In candidate ranking stage, we propose a simple but effective measure named *IWHR* (Important Word Hit Rate) to improve the ranking performance. Apart from the approach that is used to deliver the final result, this paper also describes some other latest methods that we tried but fail to yield the expected superior results.

1 Introduction

A standard Entity Discovery and Linking (EDL) system can usually be divided into two parts. The first part is an Entity Discovery system, which aims to extract all the mentions of predefined types in a collection of textual documents and have their types identified. It is also known as Name Entity Recognition (NER) task, which is well studied and included in many commercial or free NLP toolkits (Durrett and Klein, 2014; Finkel et al., 2005). In this year's EDL task, participants are not only required to extract name mentions, but also nominal and title mentions.

After the Entity Discovery, Entity Linking is performed as the second stage of the EDL system. It aims to link extracted mentions to entities in a given knowledge base (KB). In addition to that, mentions that do not have corresponding KB entries should be clustered by the system. Quite a few research works have been carried out since the emerge of this task. There are non-collective approaches that focus on modeling the consistency between a candidate entity and the context of the mention, with methods range from measuring TF-IDF similarity (Mihalcea and Csomai, 2007) to building a sophisticated deep learning model (Sun el al., 2015). On the other hand, there are also some collective approaches, which take the coherence between different entities in a same document into consideration (Cucerzan, 2007).

The approach for entity linking in our system is non-collective. We use TF-IDF similarity to measure context consistency. To further improve performance, we propose a simple but effective measure named *IWHR* (Important Word Hit Rate).

We also reproduced a deep learning model based on (Sun el al., 2015)'s method, however it fails to yield the expected superior results. A method to re-rank the top two entity candidates of each mention with a multi-layer perceptron is also tried, but it achieves no significant improvement in performance. These two methods now serve as two optional modules in th system, and are not used for the results submitted to TAC.

The rest of paper is organized as follows. We first present our system in section 2 with detail descriptions of each modules in system pipeline. Then section 3 evaluates the performance of the system on TAC 2014 and 2015 EDL datasets in different as-

*Corresponding author

pects. Finally, in the last section, we draws some conclusions.

2 System Description

We used the 2015-01-25 Freebase dump[1] instead of the officially provided LDC2015E42 as the system's target KB, but there should not be much difference between the two. We also used a Wikipedia dump (2015-04-03)[2]. The "topic equivalent webpage" values of Freebase are used to map Freebase topics to Wikipedia articles.

The pipeline of our ZJU-EDL system is illustrated in Figure 1. For first stage, mention extraction, we use Berkeley Entity Resolution System (Durrett and Klein, 2014) after preprocessing the document. And for the final stage, NIL clustering, we simply map those mentions with a same surface name to a same cluster. The rest of the system will be described in the following subsections.

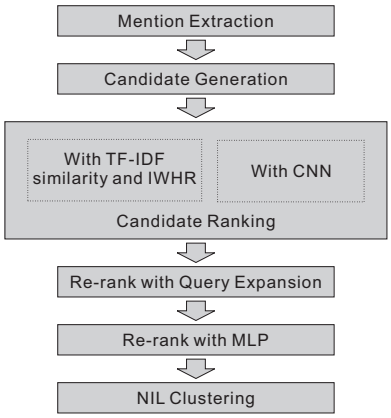

Figure 1: Pipeline of ZJU-EDL system. IWHR is short for *Important Word Hit Rate*.

2.1 Candidate Generation

For candidate generation, we adopt the most commonly used method which creates an alias dictionary out of disambiguation pages, redirect pages and anchor texts from Wikipedia (Cucerzan, 2007). We also add the "also known as" values of Freebase into the dictionary. However, a dictionary created in such way will often produce too many candidates for a mention. This may bring noise and slow down the remaining steps, therefore we ranked the candidates for surface names (a surface name is an alias of an entity) against their *commonness* (Medelyan and Legg, 2008), then take top 30 candidates for each surface name.

[1]https://developers.google.com/freebase/data

[2]http://dumps.wikimedia.org

2.2 Candidate Ranking

After candidate generation, we use *commonness* and a score that measures the consistency between a candidate entity and the context to rank the candidate entities. We currently have two ways to measure context consistency, one is a combination of TF-IDF similarity and important word hit rate, the other is a model similar to the one proposed by (Sun el al., 2015). We will describe the first one next and leave the second one to 2.4.

TF-IDF similarity is computed between the input document and the Wikipedia article of the candidate entity. It tells how similar two documents are in general, but for entity linking, this is not enough. Instead of taking all the words into consideration, it is sometimes more helpful to focus just on a few words that really matters. For example, suppose we have a mention with name string "Portland", and we want to tell whether it refers to *Portland, Maine* or *Portland, Oregon*, both of which are cities in America. If we use a simple combination of commonness and TF-IDF similarity, we may easily fail on this case. But if we find the word "Maine" in that document, we know it is very likely that this "Portland" refers to *Portland, Maine* instead of *Portland, Oregon*. Based on this observation, we get our next measure, which we call "important word hit rate", and is defined as follows.

$$f(e,m) = \frac{\sum_{w \in W_d \cap W_e, \text{idf}(w) > T} \text{idf}(w)}{\sum_{w \in W_d, \text{idf}(w) > T} \text{idf}(w)}$$

Where e is the candidate entity, m is the mention, W_d is the set of words in the input document, W_e is the set of words in the entity's Wikipedia article, $\text{idf}(w)$ is the IDF value of word w, T is a threshold to get "important" words.

2.3 Re-rank with Query Expansion

The purpose this step is to handle the problem of name variations in a single document. For example, after a first mention, "Hilary Clinton" could be

referred to as "Clinton", and "American Film Institute" could be referred to as "AFI". It is quite obvious that to disambiguate "Clinton" would be much harder than to disambiguate "Hilary Clinton".

Usually, query expansion is performed at the beginning of an entity linking progress to address this problem. We delay this step till each candidate entity is ranked with a score, so that the scores can also be utilized. We expand the name string s of mention m to the name string s' of mention m' only when: m' appears before m in the document, s is a word in s' or an acronym of s', the top ranked candidate entity of m' is a person or has a higher score than the top ranked candidate entity of m. We try to find such an m' that is closest to m in the document. The effect of query expansion here is that m is assigned the top ranked candidate entity of m'.

2.4 Modeling Context with CNN

As another attempt to model context consistency, we reproduced an approach similar to the one proposed by (Sun el al., 2015). SkipGram model (Mikolov, Tomas, et al., 2013) is used to generate a vector representation for each word. On the context side, a fixed amount of words around the mention are fed to the model proposed by (Yoon Kim, 2014) to get a vector representation v_m. On the entity side, vector representations of the words in its Wikipedia title and the words in its Freebase "notable for" value are averaged respectively. Then these two averaged vectors are concatenated to get a vector representation v_e for the entity. We then put a fully connected layer on top of v_m to get v'_m, and another one on top of v_e to get v'_e. The cosine similarity between v'_m and v'_e is used to represent the semantic relatedness between the entity and the context. We use the same way in (Sun el al., 2015) to train the parameters.

2.5 Re-rank with MLP

It can be seen that our entity linking approach takes two factors into consideration: how popular is the candidate entity and how well does it fit into the context. Normally, we combine them linearly so that when ranking candidate entities, we would consider exactly this much of popularity and this much of context consistency. But we hope to be more flexible. Maybe sometimes when two candidate entities are both popular enough, we should consider more about context consistency, and when two candidate entities can both fit into the context, we turn to popularity instead.

In order to achieve this kind of flexibility, we rerank the top two candidate entities with a multi-layer perceptron. This MLP is a two-class classifier indicating whether we should choose the first candidate as the target entity. Its input are the three measures in 2.2 plus an additional feature of the top two candidate entities, together, they make an 8-dimentional vector. The additional feature is the fraction between the total number of anchor texts in Wikipedia that link to the entity's article and the total number of anchor texts in Wikipedia, which intends to measure an entity's popularity. Since it is a pretty small MLP, we only need a few training data to train it. The use of this step is optional since we later found out that almost no improvement can be obtained with it.

3 Experiments

We show the performance of our system on TAC 2014 and 2015 EDL evaluation datasets (LDC2014E81 and LDC2015E103). Parameters of the system are tuned or trained with TAC 2014 and 2015 EDL training datasets (LDC2014E54 and LDC2015E75). The system outputs Freebase ID's for mentions, they will be mapped to the knowledge base used in previous years when necessary. Since the system only deals with name mentions, we filter nominal type and TTL type mentions in 2015 EDL datasets. We use the same evaluation measures that was used in the TAC-KBP 2014 EDL task overview (Ji el al., 2014).

Table 1 shows entity discovery performance. While many top ranked EDL systems in 2014 achieved NER scores higher than 75% (Ji el al., 2014), we can see that our entity discovery part is not good enough. This surely will impact the full EDL performance. We think the problem here lies in the preprocessing of the input documents.

Dataset	NER	NERC
2014	0.698	0.649
2015	0.729	0.653

Table 1: Entity discovery performance.

Full EDL performance is shown in table 2. Due to

the poor performance of entity discovery, this result is also not good compared with top systems.

Dataset	NERL	CEAFm
2014	0.618	0.650
2015	0.626	0.672

Table 2: Full EDL performance.

We also provide system with gold-standard name mentions to evaluate its entity linking performance. The result is shown in table 3. The performance on TAC 2014 dataset is comparable with the top ranked systems in TAC 2014 diagnostic EDL task (Ji el al., 2014).

Dataset	NERL	NEL	B-Cubed+	KBIDs
2014	0.852	0.823	0.790	0.835
2015	0.798	0.806	0.747	0.788

Table 3: Entity linking performance.

Next we show the performance of the two methods described in 2.4 and 2.5. For convenience, we call the method in 2.4 MA and the method in 2.5 MB.

Table 4 compares the performance of TF-IDF similarity and MA in modeling the consistency between a candidate entity and the context. Here, we only use the similarity value they provide to rank the candidates. The step described in 2.3 is not performed either. We can see that using MA achieves a much lower accuracy than using TF-IDF similarity. But since this model is quite similar to (Sun el al., 2015) and their experiments showed fair results, we think the cause of poor performance here might be that: 1. Some training data from Wikipedia anchor texts that link to entities whose types are not included in the task should be filtered; 2. On the entity side of the model, the Wikipedia titles and Freebase "notable for" values both vary too much, which makes the model hard to train; 3. Something is wrong with the code.

Table 5 compares the performance with and without using MB. Almost no improvement is obtained with MB. We think it is probably because the features used for the MLP are not enough. What is more, we would not get much improvement even if the correct one is always selected when it is in the top two candidates.

Method	NEL recall
TF-IDF	0.576
MA	0.345

Table 4: Performance of TF-IDF and MA on TAC 2014 dataset.

Dataset	Without MB	With MB
2014	0.823	0.824
2015	0.806	0.807

Table 5: NEL scores with and without MB.

4 Conclusion

We participated in the tri-lingual entity discovery and linking task as well as the diagnostic entity linking task, but only focused on the discovery and linking of English name mentions. The system produces competitive results for entity linking. We may need further investigation on the two methods that failed to yield superior performance.

Acknowledgments

This work was supported in part by the China Knowledge Centre for Engineering Sciences and Technology (CKCEST), the NSFC (No. 61402401), and the Zhejiang Provincial NSFC (No. LQ14F010004).

References

Silviu Cucerzan. 2015. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. EMNLP-CoNLL, volume 7, pages 708–716.

Greg Durrett and Dan Klein. 2014. A Joint Model for Entity Analysis: Coreference, Typing, and Linking. *Transactions of the Association for Computational Linguistics (TACL)*, 2:477–490

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–3701.

Heng Ji, H.T. Dang, J. Nothman and B. Hachey. 2014. Overview of tac-kbp2014 entity discovery and linking tasks. In *Proc. Text Analysis Conference (TAC2014)*, pages 1333–1339.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751.

Olena Medelyan and Catherine Legg. 2008. Integrating Cyc and Wikipedia: Folksonomy meets rigorously defined common-sense. In *Prob. AAAI 2008 Workshop on Wikipedia and Artificial Intelligence*, page 65.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, pages 3111–3119.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, Xiaolong Wang. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 1333–1339. AAAI Press.