ZJU Participation in TAC 2016 EDL task

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

Abstract

We participated in the Trilingual Entity Discovery and Linking (EDL) task at TAC KBP 2016 under the team name of WednesdayGO. This year, we made some improvements to our last year's system to achieve better overall EDL performance. We also tried a neural network approach for entity discovery, though in our experiments it performs slightly worse than the traditional approach we used. In this paper, we will mainly introduce the modifications we made to our last year's system and the neural network approach for entity discovery. We will also demonstrate the results of some experiments we conducted.

1 Introduction

The Trilingual Entity Discovery and Linking task requires submitted systems to extract mentions from documents in three languages (Chinese, English and Spanish), and then link the mentions to a Knowledge base, which in this case is a snapshot of English Freebase. An EDL system is also required to cluster mentions for those NIL entities that don't have corresponding KB entries. Due to limited time we have on this task, we only focus on English documents. Corresponds with the task requirements, our EDL system consists of two main parts: mention extraction and entity linking.

For the mention extraction part, the task requires to extract two types of mentions: name mentions and nominal mentions. We tried two approaches for extracting name mentions. In one approach, we directly apply the Stanford NER tool to get the named entity mentions it annotates. Stanford NER is a traditional Named Entity Recognizer that provides a general implementation of linear chain Conditional Random Field (CRF) sequence models and relies heavily on hand-crafted features. Since neural network approaches have been successfully applied to many NLP tasks in the past few years, in the other approach, we also try an end-to-end neural network method (Ma and Hovy, 2016). First, a convolutional neural network (CNN) is used to generate character embedding, then the character embeddings and the word embeddings are concatenated as input and is fed into a bi-directional LSTM neural network. Finally, a Chain-CRF is used to tag all the words with label.

Nominal mentions are expanded to all entity types this year, but we only extract person nominal mentions with a dictionary based method.

For the entity linking part, we made some small changes to the procedure of last year's system while left the two main steps – candidate generation and candidate ranking – unchanged. The small changes make the system run faster and easier to be understood.

We also add a step that tries to fix some of the entity type errors made in the mention extraction step with the entity linking result. Experimental results demonstrate that this simple step works well for linked mentions.

2 System Description

Figure 1 illustrates the pipeline of our EDL system. In the preprocessing step, we first extract the text

*Corresponding author

from the input files since the input files are XML formatted and contain tags not needed for subsequent steps. We also perform tokenization and POS tagging to the extracted text as it is required by both mention extraction and entity linking. After preprocessing, we extract mentions with the two approaches mentioned previously. In our three submitted runs, one of them uses the neural network method, the other two use the Stanford NER tool. Simple coreference resolution, candidate generation and candidate ranking are three entity linking steps. The "simple coreference resolution" step we put before candidate generation is one of the major changes we made to last year's entity linking system. This step reduces the number of mentions needs to be linked and makes the system less complicated than before. At last, we perform an entity type inference step, which takes advantage of the entity linking result to fix some of the entity type errors made in the mention extraction step.

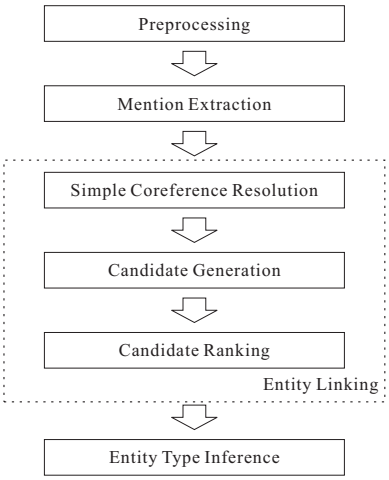

Figure 1: EDL system pipeline.

2.1 Name Mention Extraction

Since applying the Stanford NER tool to extract name mentions is quite straightforward, we only introduce the neural network approach here. The main architecture is illustrated in figure 2.

Previous studies have shown that convolutional neural network is an efficient way to extract morphological information from characters, on the bottom of our neural network architecture, we use a CNN to extract character representations of words.

Since the LSTMs hidden state only takes information from the past, and knowing nothing from the future, however we need both the past and future information to improve the accuracy of named entity tagging. The solution is bi-directional LSTM, which present each sequence forwards and backwards to two separate hidden states to capture both past and future information. Then we concatenate the character representations with pre-trained word embeddings as input to the bi-directional LSTM neural network, generate the context information of each word.

On the top of our neural network architecture, we feed the generated representations into a chain-CRF model for named entity tagging.

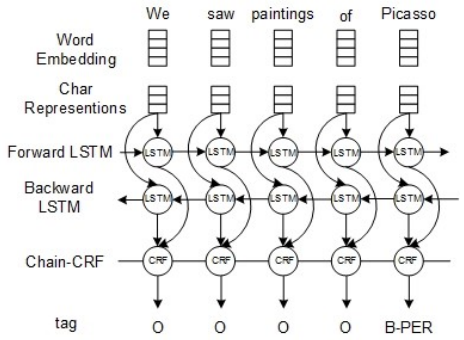

Figure 2: The architecture of the neural network approach for NER.

Post Authors and Adjectival GPE mentions. We extract these two types of mentions the same way with last year. Post authors are extracted with regular expressions. For adjectival GPE mentions, we simply match the document text against a list of adjectival forms of countries and geographic regions.

2.2 Nominal Mention Extraction

A nominal mention uses a common noun or noun phrase that refers to an entity in place of a name. Though nominal mentions are expanded to all entity types this year, we only extract person nominal mentions with a dictionary based method. The idea is to first build a dictionary of possible person nominal mention name strings, then match the document text against this dictionary to find nominal mentions.

Specifically, we build the dictionary with the training data of TAC TEDL 2015. Since the name strings of nominal mentions have to be nouns or

noun phrases, we actually only match the nouns and noun phrases we find in the text of documents against the dictionary, the matched nouns and phrases are considered nominal mentions.

One problem with this method is that some of the "nominal mentions" we find may be generic instead of referring to a specific entity. This problem is as a matter of fact not easy to solve, so we only try to alleviate it. We use the above method to extract nominal mentions in the training data of TAC TEDL 2015 and then use the result to remove those terms in the dictionary that are more likely to give us false nominal mentions.

2.3 Entity Linking

Our entity linking part consists of 3 steps: simple coreference resolution, candidate generation and candidate ranking. The candidate generation step and the candidate ranking step do not differ much with our last year's system, thus they will only be described briefly.

The simple coreference resolution step is rule based and is performed on all the extracted mentions. For name mentions, assume there are two mentions A and B, then we consider them coreferent under the following circumstances: 1) Mention A and mention B are both post authors and the name strings of A and B are the same. 2) The name string of mention A is a possible acronym of the name string of mention B. 3) Mention B occurs before mention A, the name string of mention A contains only one word and the name string of mention B contains this word. 4) The name strings of mention A and mention B are the same and contain more than one words. 5) A is a person nominal mention, B is a person mention and is the closest mention before or after A.

Note that rule 5) determines how we link nominal mentions. We set the entity ID of a person nominal mention to NIL if we can not find a coreferent mention for it with rule 5).

After the simple coreference resolution step, the mentions in a document are grouped. Those in a same group are coreferent, i.e., they all refer to a same entity. Thus we only need to link one mention in each group. Intuitively, if the name string of a mention is longer, then it is easier to link, as longer names causes less ambiguity. For example, the full name of an organization is easier to link than its acronym, the full name of a person is easier to link than his mere first name or family name. So for each group, we only link the one that has the longest name string. This makes the linking process faster and the linking result more accurate.

For candidate generation, we create an alias dictionary from four sources: the disambiguation pages, redirect pages and anchor texts from Wikipedia (Cucerzan, 2007), the "also known as" fields of Freebase. For each surface name, we keep the top 30 candidates with the highest *commonness* (Medelyan and Legg, 2008). For a name string s and a candidate entity e, commonness is calculated as follows:

$$Commonness(s, e) = \frac{count(s, e)}{\sum_{e'} count(s, e')},$$

where $count(s, e)$ is the number of hyperlinks in Wikipedia with anchor text s and links to the page of e.

We rank the candidates with the linear combination of three scores: *commonness*, TF-IDF similarity and *import word hit rate* (IWHR). TF-IDF similarity and IWHR are calculated with the text description of the candidate entity on Wikipedia and the input document text. For a mention m and one of its candidate entities e, IWHR is calculated 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 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.

2.4 Entity Type Inference

After entity linking is performed, we get the ID's of the referred entities for mentions that can be linked to Freebase. With this ID, we are able to retrieve the attributes of the referred entity in Freebase. We hope to use the "type" attributes in Freebase to infer entity types. However, the types defined in Freebase are different from the types defined in the TEDL task. Thus we designed rules to map from Freebase types to TEDL task types. Some of these rules are listed in table 1.

Method	NERC	NERLC	KBIDs	CEAFm	CEAFmC
EDL1	**0.750**	0.682	0.740	0.711	0.684
EDL2	0.725	0.655	0.703	0.679	0.659
EDL1\NOM	0.743	**0.705**	**0.743**	**0.731**	**0.704**
EDL2\NOM	0.713	0.672	0.699	0.694	0.674

Table 2: EDL performance on TAC TEDL 2016 dataset.

Freebase	TEDL
location.country location.citytown location.administrative_division	GPE
organization.organization music.musical_group ...	ORG
people.person	PER
architecture.structure architecture.building ...	FAC
location.location	LOC

Table 1: Rules for mapping Freebase types to TEDL task types.

3 Experiments

We perform experiments on this year's TAC TEDL evaluation dataset, i.e., TAC TEDL 2016. The OntoNote 5.0 dataset is used to train the neural network mention extraction model. The parameters in the entity linking part of the system are tuned with the training data of TAC TEDL 2015.

We show both the performance of the mention extraction part and the performance of our EDL system as a whole.

Table 3 demonstrates the experimental results of mention extraction. Methods M1 and M2 use Stanford NER tool and neural network method to extract name mentions respectively. Both M1 and M2 extract adjectival GPE mentions, person nominal mentions and post author mentions. M1\NOM and M2\NOM are these two methods without person nominal mentions extracted. We report the F1 of NER and NERC. The results suggest that the neural network approach performs slightly worse than the traditional approach on this dataset. We think this may be caused by the difference between the test data and the data used to train the neural networks. We can also see that extracting nominal mentions helps to improve the final performance, even though the adopted method is quite simple.

Method	NER	NERC
M1	**0.799**	**0.757**
M2	0.763	0.724
M1\NOM	0.782	0.740
M2\NOM	0.745	0.707

Table 3: Mention extraction performance on TAC TEDL 2016 dataset.

The performance of the whole EDL system is shown in table 2. The difference between EDL1 and EDL2 is that EDL1 uses the Stanford NER tool to extract name mentions while EDL2 uses the neural network approach. EDL1\NOM and EDL2\NOM does not extract nominal mentions. As we can see, EDL1 performs better than EDL2, primary because the mention extraction part of EDL1 is better. Also, including the nominal mentions actually hurts the performance of our EDL system as a whole, probably because the way we link the nominal mentions are simple and may yield poor results.

References

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, volume 7, pages 708–716.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany, August. Association for Computational Linguistics.

Olena Medelyan and Catherine Legg. 2008. Integrating cyc and wikipedia: Folksonomy meets rigorously defined common-sense. In *In Proceedings of the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI)*.