A Hybrid Model for Trilingual Entity Detection and Linking Tasks at TAC KBP 2017

Zhenzhen Li, Qun Zhang, Ting Li, Jun Xu, Dawei Feng
College of Computer, National University of Defense Technology, China
Changsha, Hunan 410073
lizhenzhen14@nudt.edu.cn, chriszhang0511@gmail.com
xujunrt@163.com, liting6259@yeah.net, davyfeng.c@qq.com

Abstract

This paper describes the our systems submitted to the Trilingual Entity Detection and Linking (EDL) track in 2017 TAC Knowledge Base Population (KBP) contests. Our system consists of three modules: a BiLSTM-CRF based sequence labeling model for entity discovery and mention detection (MD); a rule based candidate generation and a regular feedforward neural network for entity linking (EL); and the mention pair model with CART decision tree for NIL clustering. We only use the specialised word embedding for the end-to-end mention detection system without any handcrafted features. The search engine and wikidata API are helpful to query expansion, so they are used to enhance the candidate coverage in the process of entity linking. Since the challenge of NIL clustering task is coreference resolution, a learning approach of the decision tree with deliberately seleceted features is proposed to deal with the NIL mentions coreference task. Moreover, we evaluate our approach on the NIL mention dataset extracted from gold standard tables and obtain encouraging results.

1 Introduction

The EDL task requires to detect named entities and their nominal mentions in the raw text of three languages (English, Chinese and Spanish) and further link each detected mention to the corresponding node in an existing knowledge base, namely Freebase. For NIL mentions that do not exist in the knowledge base, the EDL system needs to cluster all NIL mentions and assign a unique ID to each NIL mention cluster.

The flow chart of our EDL systems is shown in Figure 1. Three stages are required to obtain the final results. Mention detection(MD) means to extract all the named entities and their nominal mentions in the raw text. Since author names in the raw xml files from discussion forum appear in certain format, such as in the angle brackets with indication of “author name=”, we can pick them up easily. What’s more, these kind of entities have few contextual texts and they are unfrequent personal name whose meanings are irrelevant to the text. Thus, we extract this kind of named entities separately, whose types are always person and nil. The other mentions are extracted by our end-to-end sequence labelling model.

Recently, sequence labelling models (Chiu and Nichols, 2016; Lample et al., 2016) via different neural networks are widely used in named entity recongnition task, which require no feature engineering or data preprocessing and achieve competitive perfomance. Thus, we use a similar sequence labelling model as (Ma and Hovy, 2016) in the mention detection phase and get the mention type, offsets simultaneously. To find the mention belongs to named or nominal type, a simple svm classification with gassian can obtain impressive results. After getting the extracted mentions and their type information, we use all the relevant information to generate linking candidates that may be the possible nodes to be linked in the given knowledge base. With the help of search engines and wikidata, we can get more relevant linking candidates, such as its alias, its formal name, or just its correct spelling format etc. Next, a regular feedforward neural network model is built to rank all the candidates, so a knowledge base node or NIL tag is determinated after this phase. In the last phase, a CART decision tree is constructed to judge if every two mentions are in the same cluster or not.

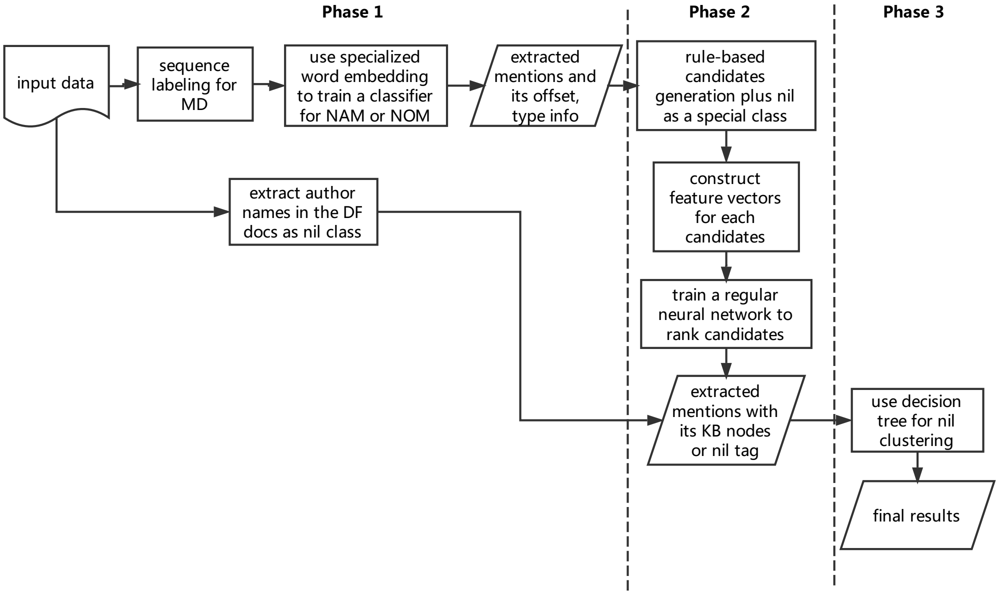
Phase 1
Phase 2
Phase 3
input data
sequence labeling for MD
use specialized word embedding to train a classifier for NAM or NOM
extracted mentions and its offset, type info
rule-based candidates generation plus nil as a special class
extract author names in the DF docs as nil class
construct feature vectors for each candidates
train a regular neural network to rank candidates
extracted mentions with its KB nodes or nil tag
use decision tree for nil clustering
final results

Figure 1: The flow chart of EDL system

2 Mention Detection

In our work, we consider the nominal mention as a special named entity and detect both named and nominal mentions in the same model. Our neural network is inspired by the work of (Lample et al., 2016).

The sentences from Chinese documents are segmented to words or phrases by NLP tools, while the sentences of the other two languages are split by space. Each word or phrase can have a pre-trained word embedding, concatenated with the char representation to form the input representation. We run a bi-LSTM over the sequence of character embeddings and concatenate the final states to obtain a fixed-size vector as char representation, which is a complement for the out-of-vocabulary words.

Then input representations are fed into a BiLSTM neural network. The output vectors of BiLSTM are fed to the CRF layer. With the last CRF layer, we can get the predicted label by a Viterbi-style algorithm which provides the optimal prediction for the measurement sequence as a whole.

The tagging scheme in our model is IOBES(Inside, Outside, Beginning, Ending, Singleton). Since there are five entity types, each corresponding to four tag prefix(IBES), output layer chooses label from 21 different tags including O(Outside) type.

After the sequence labelling process, every word in a sentence is assigned a tag, thus we can find all the mentions along with their types.

3 Entity Linking

This module consists of two steps: candidates generation and ranking, which learned some rules from the work of (Dan Liu, 2016)

3.1 Candidates Generation

The first step is called query expansion, each mention is first expanded into a number of different queries based on some pre-defined rules. By calling the wikidata API, we can get the wiki page id of each candidate. We can get the node id in Freebase with the unique page id by calling the wikidata API further. Thus, we can filter the candidates by choosing those which have wiki page id. In addition, the special candidate NIL is also added to the candidate list for each mention in order to process those NIL mentions.

The rules we pre-defined for the query expansion step are as follows:

- add the underlying mention to the query list.

- For each mention, we search the original document containing this mention. If we find this mention is a sub-string of other longer men tions. All of these longer mentions are added to the query list. For instance, if we have a mention like Bush, and we have found another mention, such as George Bush, from the same document and Bush George Bush, then George Bushis added to the query list of Bush.

- We found that: if a mention is in the abbreviation form, that is, made up of capital letters, its full name usually appears nearby; if a mention is the nominal mention, the named mention it refers to also appears nearby occasionally. Thus, if a mention meets the above cases, we search the context of this mention in the original document for a character string which is extracted as named entity, and its distance to this mention is less than two character. If such character string exists, add it to the query list.

- If a mention is Chinese or Spanish, we invoke a translation API to obtain its English translation. The English translation is used to go through the above rules 2, 3 to expand the query.

- For every query in the query list, we invoke the wikidata API to find the first three returned relevant entities and filter out those that do not have wiki page id and label simultaneously.

- If now the candidate set is still empty, we will invoke the search engine, such as google or bing to search the underlying mention on the website of wikipedia in the same language. For example, if the mention is English, “Mandela” , the search statement will be “Mandela site:en.wikipedia.org”, we only choose the first three returned entities and put them to rule5 to get the final candidates.

From the figure 1 and system description, we know that only the mentions extracted by the sequence labelling model are applied to generate candidates. We use the same criteria (Dan Liu, 2016) to measure the quality of candidate generation: the first one is the total number of different candidates generated for each mention in average (called average count), and the second one is how many candiate lists actually contain the true target node (called coverage).

test dataset	coverage	avg. count
2016 eval	0.78	3.80
2017 eval	0.75	7.69

Table 1: Performance of candidate generation on the EDL 2016 and 2017 eval dataset.

From table 1 we can see, compared to (Dan Liu, 2016), the coverage of our model is little less, but the average count is much less than their model. Therefore, we can conclude that the search engines and wikidata API are effective to generate precise candidates.

3.2 Neural Networks Ranking Model

As for each mention and its candidate, eight typical features for entity linking are extracted and projected to dense vectors. At last, a regular feedforward neural network is used to classify the pairs of the mention and one of its candidates. This phase is similar to the work of (Dan Liu, 2016), we do not describle it in detail.

4 NIL Clustering

NIL clustering aims to cluster the entities that are unable to be linked to the knowledge base, which belongs to the coreference resolution problem essentially. Our method is based on mention-pair model (Soon et al., 2001). The main idea is to treat every mention-pair as a sample, and extract pairwise features to train the CART decision tree. If the mention pair is coreferred, the label is 1, else 0.

The features are defined according to the specific task requirements and can not be generalized. Eight features are defined for the TAC KBP EDL task. As for each pair of two mentions, the first one is called the antecedent and the latter is called the anaphor. For the TAC KBP TEDL task, the features are defined as follows:

- Language match featureits possible values are 0,1. if mentions belong to the same language, then the value is 1, else is 0 We simply judge the language what the mention belongs to by the document name.

- String match feature: its possible values are 0,1. if language matched, then compare mention name, else compare translated mention name. If the string fully matched, then is 1,else is 0. For example, if both entity types are PER, mention types are NAM and string matched, then the probability that the two mentions are co-reference is large.

- Mention class of antecedent feature: if the mention class of the antecedent is NOM then value is 1, else value is 0.

- Mention class of anaphor feature: if the mention class of anaphor is NOM then value is 1, else value is 0.

- Alias featureits possible values are 0,1. For person name, if the shortname is part of fullname, then the value is 1, else the value is 0, such as “Beckham” and “David Beckham”. For orgnazition or geopolitics name, alais name may be abbreviation.

- Entity class featureits possible values are 0,1,2,3,4. if entity class is PER, then value is 0, the value of LOC is 1, the value of FAC is 2, the value of ORG is 3, the value of GPE is 4.

- The same document featureits possible values are 0,1. If the mentions have the same characters, types and they are in the same document, the value is 1, else the value is 0. For example, all the mentions “spokesman ” with the same type(PER, NOM) apper in the same document, then it is possible that they refer to the same entity, else it is possible that one refers to “Hong Lei”, anthor refers to “Lin Chun-sheng”.

- Adjacencyits possible values are 0,1. If the anaphor follows the antecedent, then the value is 1, else is 0. For example, “president” and “obama”, they are both PER,NAM, and in the origin document, the anaphor follows the antecedent, thus we infer that they are coreferred.

After classifying the mention pairs, we cluster the coreferred mentions into the same cluster. In order to consider the NIL clustering performance without the influence of previous phases, we extract all the NIL mentions to form datasets, which are from the gold standard tables of the training data or eval data in this EDL task in 2015,2016 and 2017. Our model is trained on the “ dataset1”, which includes the training data from 2015 and the eval data from 2015 and 2016. We test our model on the “dataset1” and “dataset2” that is from the gold standard table of 2017 eval dataset of this TEDL task.

Data	bcubed	Entity ceaf	Mention ceaf
dataset1_CMN	0.93	0.91	0.93
dataset1_ENG	0.90	0.88	0.85
dataset1_SPA	0.91	0.86	0.86
dataset1_All	0.90	0.86	0.85
dataset2_All	0.838	0.755	0.745

Table 2: NIL clustering performance(in terms of the F value by three scoring software)

Language	P	R	F
CMN	0.752	0.503	0.603
ENG	0.694	0.638	0.665
SPA	0.725	0.643	0.682
All	0.725	0.583	0.646

Table 3: The ofcial trilingual mention evaluation performance of our system in 2017 KBP EDL evaluation (in terms of strong typed mention match).

As table 2 shows, when dealing with the nil clustering problem at a new dataset, our approach can still present a good generalization.

5 Results

In table 3, the performance of our mention detection model shows a medium level. However, the linking performance of our model is subaverage as table 4 shows. The bottleneck of our system in entity linking is the candidates ranking model, which may due to the inappropriate feature extracttion and representation.

Language	P	R	F
CMN	0.413	0.276	0.331
ENG	0.342	0.314	0.327
SPA	0.409	0.362	0.384
All	0.725	0.583	0.646

Table 4: The ofcial trilingual entity linking performance of our system in 2017 KBP EDL evaluation (in terms of strong typed all match).

References

Jason P. C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *TACL* 4:357–370.

Wei Lin Si Wei Shiliang Zhang Hui Jiang Dan Liu. 2016. The ustc nelslip systems for trilingual entity detection and linking tasks at tac kbp 2016 .

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. pages 260–270.

Xuezhe Ma and Eduard H. 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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Wee Meng Soon, Hwee Tou Ng, and Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics* 27(4):521–544.