

COMP6714 Project Report

Tong Chen

5103316

A. Description of Approach:

The task assigned is to extract the date of birth and has parent relation from the training data set. Before doing relation extraction, I processed each word based on its annotation from the original sentence, and only extract useful words referring to current relation. For example, the date of birth relation, I only extract the words which are PERSON, DATE, and O. The words have annotation O are the key words such as born and words with similar meanings.

For instance, the sentence: 'Steve Vai, a descendant of Italian immigrants, was born in Carle Place, New York on June 6, 1960.' would be processed into *[('Steve Vai', 'PERSON'), ('was born', 'O'), ('June 6 , 1960', 'DATE')]*.


For the has parent relation, I extract the words with annotations such as PERSON, O. Similarly, words with O are the key words such as born, parent, child, son, daughter that can determine the has parent relationship between two objects.

For instance, the sentence 'Jennifer Lynn Lopez was born on July 24, 1969, in the Castle Hill neighborhood of The Bronx, New York, to Puerto Rican parents Guadalupe Rodriguez and David Lopez.' would be processed into *[('Jennifer Lynn Lopez', 'PERSON'), ('born', 'BORN'), ('to', 'TO'), ('Guadalupe Rodriguez', 'PERSON'), ('and', 'AND'), ('David Lopez', 'PERSON')]*

After that, I process the remaining words made by certain domain. The general approach I used is first to apply a most general pattern based on common sense for those two relations on the data set. For date of birth, I first apply PERSON BORN DATE. For has parent relation, I first apply PERSON BORN TO PERSON AND PERSON.

Then, check for the F1 score of both relations, see how much pattern I have covered, and check for those uncovered patterns, and modify my extractor pattern based on the uncovered sentences. By doing this repeatedly, the F1 scores of these two relations reach to a pleasant point.

A simple flow char can explain the process:

preprocess → pattern design → apply relation extraction → check F1 → modify pattern 

In this way, I can only focus on the useful information and based on the meaning of the words, whether it' s a PERSON or DATE, as well as its position in the sentence to deduce the relation. A simple program can match a sentence to the designed pattern, and determine whether within this sentence, the relation exists or not.

B. Description of pattern used in extractor:

In the date of birth relation, the pattern I used is:

1. PERSON BORN DATE

This is the most important pattern. It will find sentence like : Somebody was born on DATE.

2. DATE BORN PERSON

This can match sentence like: Born in DATE, Somebody ...

In either case, the PERSON I choose has a positional index less than DATE.

This two pattern works well, which can have a F1 score 98.6 based on the training date set.

For the has parent relation, the pattern I used is :

1 PERSON BORN TO PERSON1 AND PERSON2

This is the most common case, and the key relation words are born to. When the program find the key words born to, it can recognize that PERSON1 and PERSON2 are the parents of PERSON.

2 PERSON son of PERSON1 AND PERSON2

The key words are the son of which can be replaced as daughter of, child of, adopted by, raised by and etc.

When the program encounters the key relation words, it can recognize that PERSON1 and PERSON2 are the parents of PERSON.

The most common patterns were found by first reading the sentence from the training data set. Then, I wrote a method to print out the sentence ID where this sentence has the relation to be extracted while it does not find by my program. Additionally, the program prints out the sentences which do not have the relation but it was recognized by my program. The patterns, therefore, will be modified based on those sentences, and cover most of the corner cases.

C. Improvement of extractor

The improvement of F1 score is based on modifying the relation extractor patterns. A coarse pattern was first applied to do the relation extraction. Then, checking the missed sentences which should be extracted, modify the patterns to include those cases, until most of the cases can be extracted by the designed pattern.

Further improvement can be done by feeding more training data, so that the designed pattern can include more cases.