Ceng463 - Introduction to Natural Language Processing Spring 2017-2018 Assignment 3

Duygu Dogan
e1941962@ceng.metu.edu.tr

***Abstract*—This report aims to analyze Dependency Parsing; on two different mathematical models (Edge Prediction adn Classification).**

***Index Terms*—dependency parsing, graph based dependency parsing, maximum spanning trees**

I. Introduction

This report was written based on research papers rather than my own observations since the code was not implemented.

Here in this report, methodologies and terminologies of some basic concepts of Dependency Parsing in Natural Language Processing will be explained firstly. Then, the process of selecting features will be told.

II. Basic Concepts

A. Projective and Non-Projective Dependency Parsing

Projective dependency parsing is a method to parse sentences which do not contain long distance dependency while non-projective dependency parsing is used to parse sentences with long distance dependency.

That means, while constructing a dependency tree of a projective sentence, clashing arcs barely occurs while it is the other way round for non-projective parsing. Projective dependency parsing is useful for English since it usually has short distance dependency within a sentence. On the other hand, parsing for languages like Dutch which mostly contains non-projective sentences.

Projective:

Pat said that someone claimed you thought I would never find the solution to this problem.

Non-Projective:

The solution to this problem, Pat said that someone claimed you thought I would never find.

B. Transition and Graph-Based Dependency Parsing

Graph-based dependency parsing is based on learning a model to score possible edges within a given sentence and constructing a spanning tree with the highest score in total.

Transition-based parsing is based on learning a model to score transitions from one state to the next, by taking parse history into consideration and then it derives a complete dependency graph with the highest scored states.

C. Labeled and Unlabeled Attachment Scores

Labeled and unlabeled attachment scores are the most common methods to evaluate a dependency parser.

According to Jurafsky et al. (2017), both methods look at the correctness of the head of the token. However, unlabeled attachment ignores the dependency relation while labeled attachment refers it along with a correct dependency relation. Figure below represents the reference and system parsers for the sentence ”Book me the flight through Houston”, resulting in a Labeled Attachment Score 4/6 and an Unlabeled Attachment Score 5/6. [1]

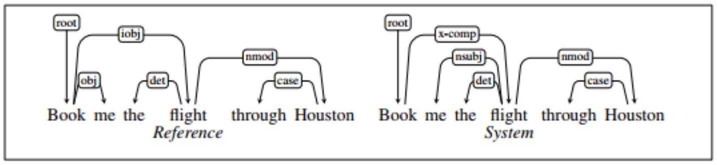
root
iobj
obj
det
nmod
case
Book me the flight through Houston
Reference
root
x-comp
nsubj
det
nmod
case
Book me the flight through Houston
System

Fig. 1: Reference and system parsers

III. Feature Selection

Since the parser was not implemented in this assignment, feature selection process could not be tested naturally. Yet, the rationality behind feature selection will be told in this section according to some experiments and researches.

A proper feature representation is necessary and crucial to get a better accuracy of dependency parsing. According to Huang (2008), feature extraction for edge prediction must depend on the pos tags of parent and child node and words of those. [2]

As can be seen from tables above which is constructed by Huang, first feature set (uni-gram features) contain parent’s and child’s word and pos tag informations separately. In order to get a higher accuracy result, second feature set (bi-gram features) which represent the relation between two-word pair should also be taken into consideration. Finally, we should also add third and forth features into our feature sets in order to get the context informations of the sentence.

Basic Uni-gram Features
p-word, p-pos
p-word
p-pos
c-word, c-pos
c-word
c-pos

TABLE I: Feature Set 1

Basic Bi-gram Features
p-word, p-pos, c-word, c-pos
p-pos, c-word, c-pos
p-word, c-word, c-pos
p-word, p-pos, c-pos
p-word, p-pos, c-word
p-word, c-word
p-pos, c-pos

TABLE II: Feature Set 2

In Between POS Features
p-pos, b-pos, c-pos
Surrounding Word POS Features
p-pos, p-pos-r, c-pos-l, c-pos
p-pos-l, p-pos, c-pos-l, c-pos
p-pos, p-pos-r, c-pos, c-pos-r
p-pos-l, p-pos, c-pos, c-pos-r

TABLE III: Feature Set 3

**p-: parent, c-: child, b-: between parent and child, -l: left of the referenced node, -r: right of the referenced node*

References

[1] D., Jurafsky, & J., Martin. (2017). Speech and Language Processing.

[2] C., Huang. (2008). Implementation of the Dependency Parser using Spanning Trees Algorithms.