NLU Assignment 3: Constituency Parsing

Prateek Sachan
M.Tech (CSA)
SR: 15754
prateeksachan@iisc

Abstract

Report for implementation of CYK parser on Unlexicalized PCFGs obtained from the penn treebank dataset in nltk.

1 Dataset

Penn TreeBank dataset is the labelled availble with the nltk library is 10% of the total data. This dataset contains around 4K labelled sentences. Out of these around 3200 are used for training and rest 800 are for testing.

The vocabulary of the training data is around 10K along with the special characters presented in the data line **,**, **.**, **?** etc.

2 PCFG

Grammar is extracted from training set and then converted into Chomsky Normal Form. There are around 25K unique productions extracted from the training data along with the count. These productions are exracted along with their individual counts in the training data to calculate the probabilities for the CKY parser.

Laplace smoothing with paramter 0.5 is performed on the extracted grammar during calculation of the probabilities for each production. Exact procedure is explained in section 4.

3 CKY Parser

Parser is implemented is the same way as described in the lecture. During building of the the parser apways picks **S** as the root of the tree if the score of **S** is not zero in the score[0][len(sentence)]. If the score is zero then parser picks the non terminal which has highest score as root node for the tree to build the tree. After building the tree for each sentence the tree is converted back into Regular Context Free Grammar from Chomsky Normal Form for better evaluation.

Metric	Score
Precision	0.74707
Recall	0.74035
F1 Score	0.74329

Table 1: Parser evaluation scores

4 Smoothing

Due to the small vocabulary of the dataset, lot common words in the test data were out of vocabulary words. Out of vocabulary words are usually ignored in the parsing step as there is no production corresponding to these words in the dataset. But in my approach I found there are only 46 non terminals, such that production of the form non-terminal $\rightarrow$ terminal exist. So, I add these 46 additional productions per oov word into the grammar with 0 count. Then Laplace add 0.5 smoothing is performed to get the final PCFGs for the parser. Due to addition of extra productions in the grammar, smoothing is done for each sentence. But all counts are pre-calculated, thus smoothing takes very less time.

5 Evaluation

Evaluation of the parser is done using precision, recall and f1 score as taught in the class by comparing the parser tree with the gold labelled tree from the test data. The scores obtained from my implementation is shown in the Table 1.

6 Qualitative Analysis

I compared with the given online parsers and found these sentence failing.

Short Sentence 1: My laptop is heating too much.
Issue: Parse is similar except that laptop is tagged as NNS(noun, plural) where as it is a singlar noun, which is correctly tagged in barkeley's parser. The reason for this is word laptop is oov.

Short Sentence 2: Deep learning
Issue: Parse started with S but Dee learning is not a complete sentence

Long Sentence 1: Machine learning is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead.
Issue: Lower level tags are wrongly selected because they are out of vocabulary

Long Sentence 2: WhatsApp was founded in 2009 by Brian Acton and Jan Koum, former employees of Yahoo. After leaving Yahoo in September 2007, they took some time off in South America.
Issue: There are two sub sentences in the sentece which are correctly identified by online parser. But my parser did show two sente but failed to do so in correct manner.

Solution: To find correct label for the oov words it is better to use some classifier that decides the correct tag of that word. For multiple sentences, I was not able to think of solution but we can parse each sentence seperately to avoid mixup.