CKY Parser and Experiments

Lucky, lucky1@iisc.ac.in

Abstract

In this assignment, We have two tasks. Task 1 is PCFG probabilities from the train set. We have to implement a CKY parser and evaluate on the test set and report appropriate metrics.

Task 2 is to compare our best parser with some existing parsers available online. We are supposed to find two short and two long sentences (from outside the treebank) where our parser produces errors but an existing parser doesn't. In each case, we have to identify if we can (by manual inspection) change/add/delete a rule or it's probability to obtain a correct parse.

1 Dataset

We are using Penn Treebank Dataset (Ann Taylor, 1992). The Penn Treebank Project annotates naturally-occuring text for linguistic structure. Most notably, they produce skeletal parses showing rough syntactic and semantic information – a bank of linguistic trees. They also annotate text with part-of-speech tags, and for the Switchboard corpus of telephone conversations, dysfluency annotation.

```
(S
  (NP-SBJ (PRP They))
  (VP
    (VBP succeed)
    (NP
      (NP
        (NP (NNP Daniel) (NNP M.) (NNP Rexinger))
        (, ,)
        (NP
          (VBN retired)
          (NNP Circuit)
          (NNP City)
          (JJ executive)
          (NN vice)
          (NN president)))
      (, ,)
      (CC and)
      (NP
        (NP (NNP Robert) (NNP R.) (NNP Glauber))
        (, ,)
        (NP (NNP U.S.) (NNP Treasury) (NN undersecretary)))
      (, ,))
    (PP-LOC (IN on) (NP (DT the) (JJ 12-member) (NN board))))
  (. .))
```

Figure 1: Example of a tree from the dataset

They are located in the LINC Laboratory of the Computer and Information Science Department at the University of Pennsylvania.

All data produced by the Treebank is released through the Linguistic Data Consortium. Part of the Penn English Treebank is available within nltk. We have explored it within nltk, where trees are represented with the nltk.tree.Tree class. (Levy, 2015)

2 CKY Parser

The Cocke Younger Kasami Algorithm (CYK or CKY) (Ann Taylor, 1992) is a highly efficient parsing algorithm for context-free grammars. The standard version of CKY can only recognize languages defined by context-free grammars in Chomsky Normal Form (CNF).

Informally, the algorithm works as follows: In the first step we write the word in the first row and add each non-terminal symbol in the row underneath which deduces the terminal symbols. After that, for each cell in the grid start vertically at the top and go down towards the cell to be checked and the second cell up in diagonal. For each such step, combine the cells and check if the combination appears in the grammar. If it does, add the left-hand-side non-terminal to the grid-cell. If after all steps the start-symbol is contained in the last row, the word can be derived by the given grammar. The dynamic programming algorithm requires the context-free grammar to be rendered into Chomsky normal form (CNF), because it tests for possibilities to split the current sequence in half.

Any context-free grammar that does not generate the empty string can be represented in CNF using only production rules of the forms $A \rightarrow \alpha, A \rightarrow BC$.

3 Task 1

First, preprocessing of the tree bank is done to replace single occurence terminals with ”unk” symbol. Then the new tree bank is binarized and then PCFG is created. The result on the test data are following. Instructions are written in readme file at github.

1. The precision score got is 0.846
2. Recall is 0.816
3. F-1 score is 0.831

4 Task 2

Two short sentences: Sentence 1: ”This is a car.”

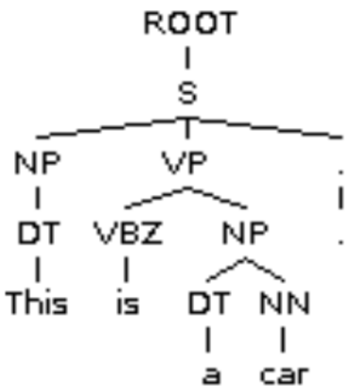

Figure 2: Short Sentence 1

Parsing by the algorithm:
[DT ’This’] [[VB ’is’] [VB ’a’]][[N ’car’] [. ’.’]]

Sentence 2: ”Hello, there!”

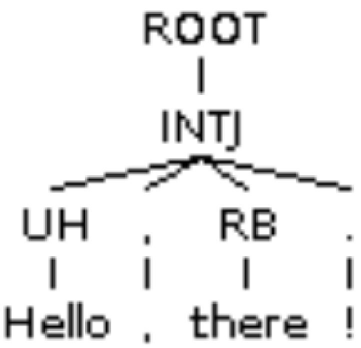

Figure 3: Short Sentence 2

Parsing by the algorithm:
[RB ’Hello’] [[[, ’,’] [RB ’there’]] [. ’!’]]

Two long sentences: Sentence 1:”I shot an elephant in my pajamas”

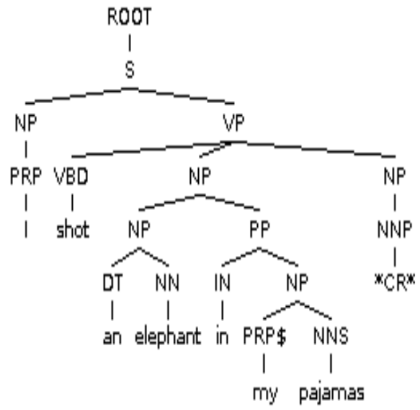

Figure 4: Long Sentence 1

Parsing by the algorithm:
[S [NP ’I’] [VP [V ’shot’] [NP [NP0 [Det ’an’] [N ’elephant’]] [PP [P ’in’] [NP [Det ’my’] [N ’pajamas’]]]]]]

Sentence 2: ”I am really upset that you always fight over small issues.”

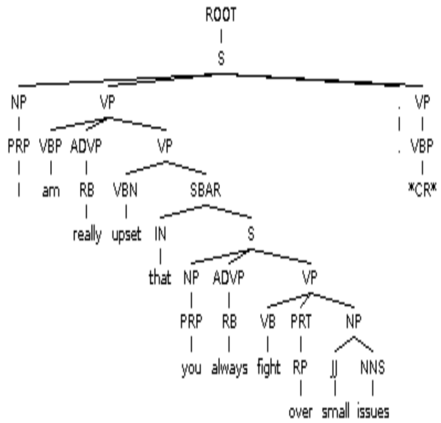

Figure 5: Long Sentence 2

Parsing by the algorithm:
[S [’I’ PRP] [VP [V ’am’] [NP [NP0 ’really’] [N ’upset’]][[IN ’that’] [NP ’you’] [NP ’always’][N ’fight’] [RP ’over’] [NP ’small’][NNS ’issues’] [.’.’]]

By changing the probabilities different tags can be obtained, but still one or two tags were wrong in many sentences.

References

Beatrice Santorini Ann Taylor, Mitchell Marcus. 1992. The penn treebank project.

Roger Levy. 2015. Treebanks and pcfgs.