**Introduction to Natural Language Processing**

**Homework Assignment 14**

רומן ציפרון Roman Tziprun 306591645

1. **Baseline Architecture**
2. Dummy parser: Precision = 0, Recall = 0, f-score is undefined
3. Tagging accuracy = 18.78

**2 Training PCFGs**

Which means that the probability of the emission is defined as the number of emissions out of the total number of any emission of the form (where is any terminal or non-terminal variable) in the training data.

1. The idea of the algorithm is to redefine any rule with more than two variables on its right hand side with new rules with exactly two variables on their right hand side. For example: the rule will be replaced with the rules:

This is equivalent to infinite horizontal Markovization.

We assume that if a rule has more than one symbol on its right hand side, these symbols are non-terminals. The pseudocode:

Given a rule of the form

1. Define a list of non-terminals
2. Add the rule to the grammar
3. For from to

3.1) Set

3.2) Append to

3.3) Add the rule

1. Add the rule

This algorithm has been implemented in train.Train.train(Treebank myTreebank).

1. The meaning of this transformation is that now every new non terminal that added in the Binarization phase is unique to its context. Now for example a VP coming after NP PP is different than a VP coming after NP JJ (if ). Horizontal Markovization can help solve the weakness of independence assumptions since now the context of each transition is remembered and represented by the fact that each new variable is defined by the variables preceding it.
2. **Decoding with PCFGs**
3. Given an array of length n which represents a sentence, we define the CKY algorithm as follows:

For from to

For all such that

()

For from down to

For from to

For all such that

If

()

():

Set

Set

Set

For all such that

While is not empty

For all in

For all

If

If

Space complexity: We have a chart of size where every entry contains a map of size at most . So the space complexity is .

Time complexity: For each entry of the chart we process other entries, and have at most (number of rules in the grammar) rules, so the time complexity is

1. **Results and Analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| h | Precision | Recall | f-score | Tagging accuracy |
| 0 | 66.07 | 56.78 | 61.08 | 86.17 |
| 1 | 64.92 | 59.28 | 61.97 | 86.30 |
| 2 | 65.90 | 62.27 | 64.03 | 86.23 |
| -1 | 63.99 | 62.07 | 63.02 | 86.24 |

1. Type 1: word is not in training vocabulary, type 2: ambiguity (the chosen parse for a word has more occurrences in train than the correct one), type 3: wrong parse leads to tree with better probability (the parse chosen for a word is not as frequent in train as the correct one)

|  |  |  |  |
| --- | --- | --- | --- |
| Sentence | Number of type 1 errors | Number of type 2 errors | Number of type 3 errors |
| 1 | 2 | 0 | 0 |
| 2 | 0 | 0 | 0 |
| 3 | 1 | 2 | 1 |
| 4 | 2 | 4 | 4 |
| 5 | 0 | 0 | 0 |
| 6 | 2 | 1 | 1 |
| 7 | 0 | 4 | 1 |
| 8 | 2 | 3 | 2 |
| 9 | 0 | 0 | 0 |
| 10 | 1 | 1 | 0 |