

ANSWER 5

To improve/enhance the F1 score / performance of the CKY parser, I implemented the standard practices of replacing rare/infrequent words having a frequency less than 2 with <unk>. Preprocessed the train.trees file to binarize and removed unary relations between nodes. Used log probabilities to prevent underflow. Used a training dictionary of words with a frequency more than 1 to identify sparse words and replace them with <unk>. Recorded the replaced words so that they can be used to replace the <unk> tag at the time of post processing of the syntax trees formed by CKY parsing of the test sentences.

Modification 1

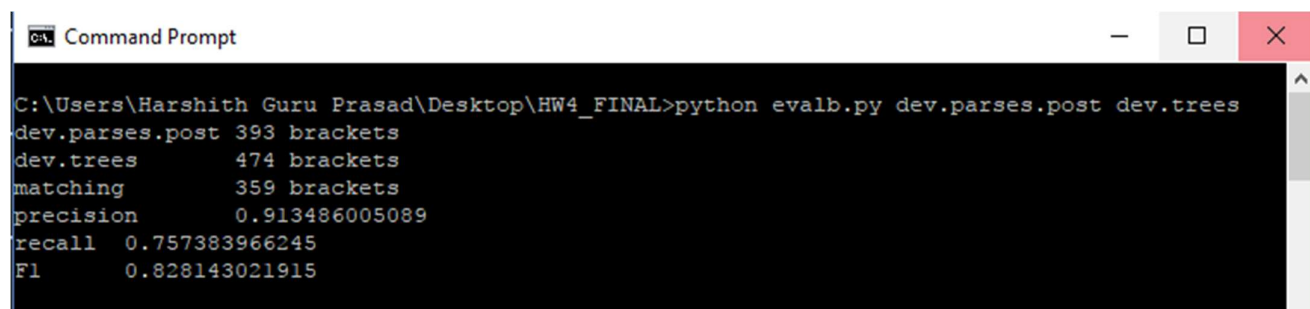
Smoothing for lexicalised pCFG – replaced infrequently seen/used words with <unk> when parsing and replaced some interesting infrequent words with stems like <unk>-ing. Improved the F1 score by 4-6%. Used the snowball stemmer to stem infrequent words with interesting suffixes to increase the overall parse recall thereby leading to a better F1 Score. Tried head lexicalising and binarization. Achieved slightly better results.

Modification 2

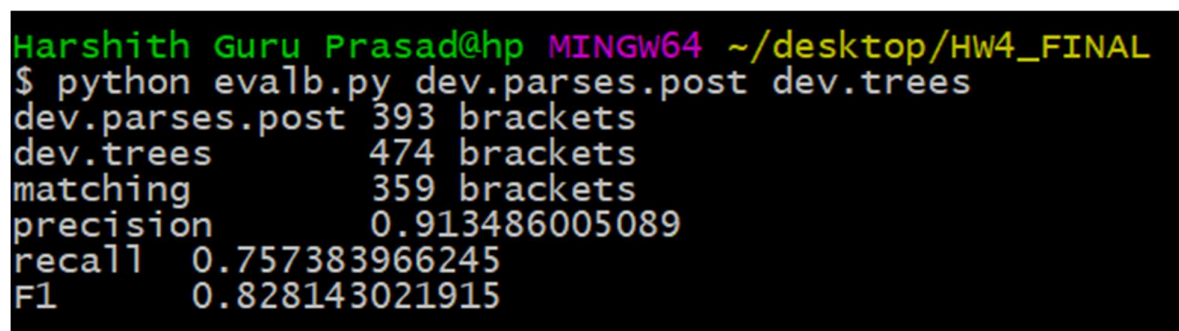
Parent annotation: Appended the label of the parent to each of its binarized dependents. Provides more specific rules to parse the sentence and is simultaneously not too specific that it cannot generalize well to new sentences/words. Improved the accuracy by 3-5%. Tried to annotate sibling information into the nodes at the expense of a quadratic runtime complexity that made the model more prone to overfitting.

Modification 3

Beaming: Tried to enhance the parser's F1 score by neglecting all the states that have the least probabilities. Once a higher probability is found, the beaming procedure eliminates the states having lower probabilities including their back tracking records. However, it did not affect the F1 score by the expected measure and it runs the risk of returning a suboptimal solution and leads to search errors for some inputs. Made other slight variations to achieve a F1 score of nearly 0.83.



```
ca. Command Prompt
C:\Users\Harshith Guru Prasad\Desktop\HW4_FINAL>python evalb.py dev.parses.post dev.trees
dev.parses.post 393 brackets
dev.trees      474 brackets
matching      359 brackets
precision      0.913486005089
recall 0.757383966245
F1 0.828143021915
```



```
Harshith Guru Prasad@hp MINGW64 ~/desktop/HW4_FINAL
$ python evalb.py dev.parses.post dev.trees
dev.parses.post 393 brackets
dev.trees      474 brackets
matching      359 brackets
precision      0.913486005089
recall 0.757383966245
F1 0.828143021915
```