Parts of Speech Tagger

Animesh Karmakar (120101010), Akib Khan (120101034)

March 14, 2016

1 Deliverable 1

In this deliverable we have implemented a Parts of Speech Tagger using Trigram Hidden Markov Model with Laplace smoothing. Here the training file used is "Brown_tagged_train.txt" and for testing we use "Brown_train.txt". The words with frequency less than 6 in the training file are mapped to key-word "rare" and similarly in the test set. Then we calculate counts like the word-tag, trigram, bigram and unigram of tags. The code for this part is mentioned in "part2.py".

1.1 Steps of execution

1. The input to the program is as discussed above which is inside the "Data" folder and the output file is by default "Data/output1.txt" (line number 127).
2. Execute the program using "python part1.py"

2 Deliverable 2

In this deliverable we apply a different smoothing method which is Linear Interpolation method to our trigram POS tagger. The parameters $\lambda 1$, $\lambda 2$ and $\lambda 3$ were calculated using Deleted interpolation method. Additionally, we have also applied a different categorization of rare words which is implemented in "sub_categorize(word)" module in which we map the rare words into groups of alpha-numerals, numerals, hyphenated words, capitalized, etc. This method results in increase in overall performance because the F1 score increases by two percentage.

2.1 Steps of execution

1. The input to the program is as discussed above which is inside the "Data" folder and the output file is by default "Data/output2.txt" (line number 203).
2. Execute the program using "python part2.py"

3 Deliverable 3

In this deliverable we calculate precision, recall and F1-score using inputs "Brown_tagged_train.txt" and other file being output generated by part1 or part2. As we can observe f1-score of output generated by part2 is more than that of part1 our model has improved. Also confusion matrix is printed, so we can observe that for most of the tags ”true positive” values are more.

3.1 Steps of execution

1. Change file names in line number 160 to desired input (”Brown_tagged_train.txt” and ”output1.txt” or ”output2.txt”)
2. Use the command python part3.py to execute the program
3. On the command prompt confusion matrix and Precision, Recall and F1-score will be displayed.
Part1 - Precision: 0.94, Recall : 0.81, F1 Score: 0.87
Part2 - Precision: 0.94, Recall : 0.84, F1 Score: 0.89