COM6513 Lab 6: Viterbi and Beam Search
Registration Number: 160122440

1. INTRODUCTION

The aim of this lab session was to accelerate part-of-speech tagging with the structured perceptron, by implementing Viterbi and Beam search for the argmax operation.
The dataset used for this particular task were sentences of up to length 5, which were imported from Brown corpus. The first 2500 sentences were used as training data and the remaining 733 were used as the testing data.
For the feature function in the algorithm, assuming each sentence-pos tag sequence (x, y) is represented as $(word_1, tag_1 ... word_N, tag_N)$, two kinds of features were implemented: $word - tag_i$ and $tag - tag_{-1}$.
For the implementation of the algorithm, the first function created ($split\_data\_to\_train\_and\_test$) was used in order to split the dataset to training and testing parts. The second function implemented ($features$) was used to extract the features explained above. Moving on, a function to get all the possible part-of-speech tags was created ($tags$), and a total of 13 tags were collected (['.', 'START', 'PRON', 'VERB', 'NOUN', 'ADJ', 'DET', 'ADV', 'PRT', 'CONJ', 'NUM', 'ADP', 'X']). Then, a function to extract all the words from each sentence was implemented ($sentence\_words$), along with a function to calculate the dot product (dot). $Training$ and $testing$ functions were created to train the structured perceptron and test it by evaluating the prediction accuracy, and finally $viterbi$ and $beam_search$ functions were used to accelerate the 'argmax' operation of both training and testing.

2. IMPROVEMENTS

Shuffling, multiple passes and averaging improvements were applied to the algorithm, since these improvements greatly improve the accuracy. Shuffling is a generally a very important improvement since the results provided are considered to be more reliable. This is because data partitions normally come from different sources and shuffling them improves the training procedure. Moving on, multiple passes benefits the accuracy because the training procedure is repeated several times, and at each iteration the accuracy is slightly improved. Finally, averaging provides a better overall representation of the weights.

3. ANSWERS TO QUESTIONS

3.1 Viterbi Speed

Viterbi greatly reduces the execution time required for the training operation. The execution time required was 3 seconds.

3.2 Beam Search

Beam search does indeed affect the word level accuracy. The lower the beam size chosen, the lower the word level accuracy, but faster speed. As the beam size is increased, accuracy increases as well but time required (speed) increases. If beam size is chosen to be the same value as the number of all the tags (13) then the accuracy and speed are almost identical to the viterbi's.

3.3 Features

Yes, same features can be used in both cases (viterbi and beam search) because 'label in sequence features' were not used. This type of features cannot be used for implementations following the Markov assumption.
Yes, more training data can be used when using bigger sentences, such as higher order n-grams.