BBM497 - INTRODUCTION TO NATURAL LANGUAGE PROCESSING LAB.
ASSIGNMENT 2

21328064 - Kübra HANKÖYLÜ
kubrahankoylu@gamil.com

1 Introduction

In this assignment, we implemented a PoS tagger using Hidden Markov Models (HMMs). Therefore, we practiced HMMs and Viterbi algorithm in this assignment.
PoS tagging determines the meaning that a word earns depending on its position. HMM does the prediction of the next ones depending on the events that occurred with PoS tagging.
The train data is 70% of the whole dataset, and the test dataset is 30% of the whole dataset.

2 Implementation

2.1 Build a Bigram Hidden Markov Model

First of all, the data was read and divided into train and test. Then the probabilities were calculated from the formulas that has given below. Each words in the train dataset are given with its PoS tag and those are the hidden states that provides the creation Hidden Markov Model. This model have three components:

- **Initial Probability** : This probability calculated for just first words in the sentences.Assume n sentence in the dataset:

$$p(t) = \frac{C(t)}{C(n)} \tag{1}$$

- **Transition Probability** : This is the probability that sequentially the i^{th} tag and $(i+1)^{th}$ tag will appear.

$$p(t_{i+1}|t_i) = \frac{C(t_i t_{i+1})}{C(t_i)} \tag{2}$$

- **Emission Probability** : The probability is that the word w is generated from the t tag.

$$p(w|t) = \frac{C(wt)}{C(t)} \tag{3}$$

2.2 Viterbi Algorithm

This algorithm was executed with test dataset. The algorithm calculates the probabilities for all possible situations. For calculating the each first word probability in the each line of the test dataset :

$$(initialprobability) * (emissionprobabilityforthisword) \tag{4}$$

Calculation of other words in the test dataset :

$$max((previoustagprobability) * (transitionprobability) * (emissionprobability)) \tag{5}$$

After the calculation finished, I did backprobagation for each sentence in the test dataset. The results are in the "output.txt" file.

The accuracy is : 0.28616317530319735