## Table of Contents

# General Overview:

In Programming Assignment 4, the task was to train a Perceptron model for Part of Speech Tagging.

All the code of the assignment is in a folder **'code'** under the current directory. Shell script **'run.sh'** will run all necessary scripts for the programming assignment and produce the required outputs inside a directory called ***'output'*** under the current directory.

For executing the code for the assignment, go to the current directory and type the following commands in the command prompt:

***>> chmod 777 run.sh***

***>>./run.sh***

# Question 4:

## Bigram Tagging decoder with a pre-trained model

### Features used

$$g_{BIGRAM:u:v}(< t_{i-1}, x, i >, t_i) = \begin{cases} 1 & if\ t_{i-1} = u\ and\ t_i = v \\ 0 & otherwise \end{cases}$$

$$g_{TAG:u:r}(< t_{i-1}, < w_1 \dots w_n >, i >, t_i) = \begin{cases} 1 & if\ w_i = r\ and\ t_i = u \\ 0 & otherwise \end{cases}$$

### Results

Based on the pre-trained weights provided in the file 'tag.model' and the steps mentioned in the question, the highest scoring tagging is found and is stored in **'tag_dev.out'** in the **'output'** folder under the current directory.

The tagger performs with an accuracy of **90.53%.** The tagger is able to tag 2226 words correctly out of a total of 2459 words.

# Question 5:

## Perceptron

A new set of features based on the suffix (of length 1, ,2 and 3) of the current word and the current tag is generated. The Perceptron model is run for 5 iterations on the training data 'tag_train.dat' to generate the weight vector. This weight vector is stored in **'suffix_tagger.model'** in the **'output'** folder under the current directory.

## Additional Feature used: The SUFFIX feature

$$g_{SUFF:u:j:v}(< t_{i-1}, (w_1 \dots w_n), i >, t_i) = \begin{cases} 1 & if\ u = suffix(w_i, j) and\ t_i = v, j \in \{1,2,3\} \\ 0 & otherwise \end{cases}$$

## Results

**Execution time** for the 5 iterations of the **Perceptron model** on the training set in order to compute the suffix features is **approximately 1.92 minutes.**

Once the perceptron model is trained, the new suffix features were combined with the features from Question 4. The resulting tagger showed an **improvement** in performance in Part of Speech tagging.

This time it could tag 2270 words correctly thus raising its performance to **92.31 %.**

## Question6

This question is all about experimentation. 3 experiments are carried out. Each time a new type of feature is added to the existing set of features. Every time the Perceptron model is trained with the new set of features and the tagged output is evaluated.

### Experiment 1:

#### Additional Feature used: <previous word, current tag>

$$g_{u:v}(< t_{i-1}, (w_1 \dots w_n), i >, t_i) = \begin{cases} 1 & if\ w_{i-1} = u\ and\ t_i = v \\ 0 & otherwise \end{cases}$$

#### Results

**Execution time** for the 5 iterations of the **Perceptron model** on the training set in order to compute the suffix features is **approximately 2.1 minutes.**

Once the perceptron model is trained, the new suffix features were combined with all features computed previously. The resulting tagger showed an **improvement** in performance in Part of Speech tagging.

This time it could tag 2302 words correctly thus raising its performance to **93.62 %.**

## Experiment 2:

### Additional Feature used: <previous two words, current tag>

$$g_{u:v:s}(< t_{i-1}, (w_1 \dots w_n), i >, t_i) = \begin{cases} 1 & if\ w_{i-2} = u, w_{i-1} = v\ and\ t_i = s \\ 0 & otherwise \end{cases}$$

### Results

**Execution time** for the 5 iterations of the **Perceptron model** on the training set in order to compute the suffix features is **approximately 2.38 minutes.**

Once the perceptron model is trained, the new suffix features were combined with all features computed previously. The resulting tagger showed an **improvement** in performance in Part of Speech tagging.

This time it could tag 2309 words correctly thus raising its performance to **93.89 %.**

## Experiment 3:

### Additional Feature used: <current word, next word, current tag>

$$g_{u:v:s}(< t_{i-1}, (w_1 \dots w_n), i >, t_i) = \begin{cases} 1 & if\ w_i = u, w_{i+1} = v\ and\ t_i = s \\ 0 & otherwise \end{cases}$$

### Results

**Execution time** for the 5 iterations of the **Perceptron model** on the training set in order to compute the suffix features is **approximately 2.61 minutes.**

Once the perceptron model is trained, the new suffix features were combined with all features computed previously. Unfortunately adding this feature diminished the performance by 9 tags.

This time it could tag 2300 words correctly thus its performance is **93.53 %.**

## General Observations:

The performance of the Part of Speech tagger showed a consistent improvement in performance when the suffix, previous word and previous two words features were used. But computing a feature based on the current and next word did not prove very helpful. Thus adding more features does not always imply that the performance of the tagger is going to improve. Feature Selection is a challenging task, which can be concluded from Experiment 1, 2 and 3 where the features selected in Experiment 1 and 2 led to an improvement in performance while the features in Experiment 3 did not benefit the Part of Speech tagging in any tangible way.

Also as we increase the number of features, the execution time of the code increases slightly each time.