Assignment 2: Part of Speech Tagging

Sabir Ismail

SBU ID: 111734933

1. **Introduction:** In this assignment I implemented the Viterbi algorithm and added new features for part-of-speech tagging using Max Entropy Markov Model and Conditional Random Filed.

2. **Viterbi Implementation**:

The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states—called the Viterbi path—that results in a sequence of observed events. The algorithm generates a path X = (x_1,x_2,…x_t) which is a sequence of states x_n $\in$ S =$\{s_1, s_2, \ldots, s_k\}$ that generate the observations Y = $\{y_1, y_2, \ldots, y_t\}$ with $y_n \in$ O = $\{o_1, o_2, \ldots, o_n\}$, n = number of observations.

Input:

Observed states: O = $\{o_1, o_2, \ldots, o_n\}$,

Hidden starts: S = $\{s_1, s_2, \ldots, s_k\}$

Start probability: $\pi = \{ \pi_1, \pi_1, \ldots, \pi_k \}$ where π_I stores the probability that $x_1 = s_i$

An observed sequence: Y = $\{y_1, y_2, \ldots, y_t\}$

Transition matrix T[i,j]= probability of going from the observed state s_i to s_j.

Emission probability E[I,j] = probability of the observing the hidden states o_j from the observe state s_i.

I used the algorithm described in the wiki, https://en.wikipedia.org/wiki/Viterbi_algorithm.

```
function VITERBI(O, S, Π, Y, A, B) : X
    for each state i ∈ {1, 2, ..., K} do
        T1[i,1] ← π_i · B_{i y_1}
        T2[i,1] ← 0
    end for
    for each observation i = 2, 3, ..., T do
        for each state j ∈ {1, 2, ..., K} do
            T1[j,i] ← max_k (T1[k, i − 1] · A_kj · B_{j y_i})
            T2[j,i] ← arg max_k (T1[k, i − 1] · A_kj · B_{j y_i})
        end for
    end for
    z_T ← arg max_k (T1[k, T])
    x_T ← s_{z_T}
    for i ← T,T-1,...,2 do
        z_{i-1} ← T2[z_i,i]
        x_{i-1} ← s_{z_{i-1}}
    end for
    return X
end function
```

Initialize the start probability:
Q = number of hidden states
T = number of observed states
p[i,j] = stores the maximum value
back[i,j] = store the path go get the maximum value

```
for q in Q:
      p[0,q] = start_scores[q] + emission_scores[0,q]
      back[0,q] = 0
```

For each observation, calculate the max value, for the probability from going one observation state to next and emission probability of the current observation and the transition probability at the previous observation, which is stored at p[i,j]. Also the index is stored at back[i,j].

```
for t in np.arange(T-1): #
      for q in Q:
            p[t+1,q] = np.max([emission_scores[t+1, q] + p[t, qp] +
trans_scores[qp, q] for qp in Q])
            back[t+1,q] = np.argmax([emission_scores[t+1, q] + p[t, qp]
+ trans_scores[qp, q] for qp in Q])
```

Get the maximum score and index after adding the end probability.

```
max_score = np.max([p[T-1, qp] + end_scores[qp] for qp in Q])
last_index = np.argmax([p[T-1, qp] + end_scores[qp] for qp in Q])
tag_list = []
tag_list.append(last_index)
back_tag = last_index
```

Now I start from the end index and backtrack the hidden states calculated previously and was stored in the back.

```
for t in np.arange(T-1,0,-1):
      tag_list.append(back[t][back_tag])
      back_tag = back[t][back_tag]
```

Final sequence is return after reverse.

```
tag_list = tag_list[::-1]
```

3. **Features:**

I go through the sentences in the training data set and after observing, I have tried with the following features. Some of the features are very basic, for example word length, number of digits. I also tried with some external dataset for the features. I used Brown Clustering and GloVe pretrained word embedding. First, I generate all the sentence form the training data set and run the Brown Clustering algorithm. When, I use it as a feature, I tried with different number of bits form the output of the Brown Cluster. In the GloVe, I used two pretrained corpus, (i) google-news and (ii) twitter. As the data set contains texts form the twitter, so I assume may be the word embadding form the twitter will give the better results and I also got the best model using the word embadding form twitter. First, I run the word embedding on all the words in my training dataset and retrieve most similar 5 words and use them as a feature.

Feature Name	Description
WORD_LENGTH	Number of letters in the word Observation: Some POS has words with smaller length. Example: pronoun, determiner.
FIRST-ONE-LETTER	First one letter of the word [Prefix] Observation: When converting from noun to adjective, some specific letters are added as prefix.
FIRST-TWO-LETTER	First two letters of the word [Prefix] Observation: When converting from noun to adjective, some specific letters are added as prefix.
FIRST-THREE-LETTER	First three letters of the word [Prefix] Observation: When converting from noun to adjective, some specific letters are added as prefix.
LAST-ONE-LETTER	First one letter of the word [Suffix] Observation: Noun, verb and adjectives has some specific letters when changing number, gender, verb form.
LAST-TWO-LETTER	First two letters of the word [Suffix] Observation: Noun, verb and adjectives has some specific letters when changing number, gender, verb form.
LAST-THREE-LETTER	First three letters of the word [Suffix] Observation: Noun, verb and adjectives has some specific letters when changing number, gender, verb form.
WORD-POSITION	Position of the word in the sentence. Observation: Normally sentence start with the noun and end with verb.
PUNCT-CNT	Numbers of punctuation in the word. Observation: Most likely those words are garbage.
DIGIT-CNT	Number of digits. Observation: Most likely they are number.
CAPS-CNT	Number of capital letters Observation: Noun are normally start with capital letter.
BIGRAM	Bigram include the current word Observation: POS of a word also depend on the neighboring word. ‘Book’ can be noun or verb based on surrounding words.
TRIGRAM	Trigram including the current word Observation: POS of a word also depend on the neighboring word. ‘Book’ can be noun or verb based on surrounding words.
WORD-RELATIVE-POS	Word’s relative position in the sentence, I assume the sentence length is 10. Observation: Normally sentence start with the noun and end with verb. A sentence has different length, so I used relative position of the word in the sentence.
BROWN:5	Output from the Brown Cluster, first 5 bits. Observation: Words in the same cluster may have same POS tag.
BROWN:10	Output from the Brown Cluster, first 10 bits. Observation: Words in the same cluster may have same POS tag.
BROWN:15	Output from the Brown Cluster, first 15 bits. Observation: Words in the same cluster may have same POS tag.
BROWN:20	Output from the Brown Cluster, first 20 bits. Observation: Words in the same cluster may have same POS tag.
BRPWN: 30	Output from the Brown Cluster, first 30 bits. Observation: Words in the same cluster may have same POS tag.
w2v- GloVe (wiki)	I use pre-trained word embadding from the GloVe (wiki) and take top 5 words. Observation: Words in the same cluster may have same POS tag.
w2v- GloVe (Twitter)	I use another version of the pre-trained word embadding from the GloVe which contains dataset from twitter and take top 5 words. Observation: Words in the same cluster may have same POS tag.

4. **Comparison:**

I tried with different combination of the features to get the best result. The following table shows the results. I tried with both CRF and LR model for the same set of features. The best score is showed in the bold for every case. In most cases the LR perform better, there are only few cases where CRF perform better than the LR. However, my best accuracy is achieved by the CRF. First, I run with adding one of my features at a time and then I combined multiple features with already provided features. Although, I run a lot of variation, I included a sub set of my results.

Comparison: One single feature does not increase the accuracy a lot except word embadding in CRF, and suffixes in LR which give be accuracy around 86. Also, adding bi-gram and tri-gram decrease the performance of both CRF and LR comparing with the accuracy with the default features. I got my best results by not using default features in the CRF.

Best Model Accuracy:

LR: 88.03

Features used: IS_AMNUL, IS_NUMERCI, IS_DIGIT, IS_UPPER, IS_LOWER, FIRST-ONE-LETTER, FIRST-TWO-LETTER, FIRST-THREE-LETTER, LAST-ONE-LETTER, LAST-TWO-LETTER, LAST-THREE-LETTER , PUNCT-CNT ,DIGIT-CNT, CAPS-CNT, BIGRAM, TRIGRAM, WORD-RELATIVE-POS , PREV, NEXT, BROWN: 30.

CRF: 88.46

Features used: FIRST-ONE-LETTER, FIRST-TWO-LETTER, FIRST-THREE-LETTER, LAST-ONE-LETTER, LAST-TWO-LETTER, LAST-THREE-LETTER, PUNCT-CNT, DIGIT-CNT, CAPS-CNT, BIGRAM, TRIGRAM, WORD-RELATIVE-POS, BROWN: 30, w2v Twitter.

Features	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**
SENT_BEGIN	y	y	y	y	y	y	y	y	y	y	y	y
SENT_END	y	y	y	y	y	y	y	y	y	y	y	y
WORD	y	y	y	y	y	y	y	y	y	y	y	y
LCASE	y	y	y	y	y	y	y	y	y	y	y	y
IS_AMNUL	y	y	y	y	y	y	y	y	y	y	y	y
IS_NUMERCI	y	y	y	y	y	y	y	y	y	y	y	y
IS_DIGIT	y	y	y	y	y	y	y	y	y	y	y	y
IS_UPPER	y	y	y	y	y	y	y	y	y	y	y	y
IS_LOWER	y	y	y	y	y	y	y	y	y	y	y	y
WORD_LENGTH			y	y								
FIRST-ONE-LETTER					y	y						
FIRST-TWO-LETTER							y	y				
FIRST-THREE-LETTER									y	y		
LAST-ONE-LETTER											y	y
LAST-TWO-LETTER												
LAST-THREE-LETTER												
WORD-POSITION												
PUNCT-CNT												
DIGIT-CNT												
CAPS-CNT												
BIGRAM												
TRIGRAM												
WORD-RELATIVE-POS												
PREV	y	y	y	y	y	y	y	y	y	y	y	y
NEXT	y	y	y	y	y	y	y	y	y	y	y	y
BROWN:5												
BROWN:10												
BROWN:15												
BROWN:20												
BRPWN: 30												
w2v												
w2v Twitter												
Accuracy	84.3	**84.39**	84.4	**85.53**	84.2	**85.1**	84.44	**84.77**	84.25	**84.48**	84.58	**84.67**

Features	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**
SENT_BEGIN	y	y	y	y	y	y	y	y	y	y
SENT_END	y	y	y	y	y	y	y	y	y	y
WORD	y	y	y	y	y	y	y	y	y	y
LCASE	y	y	y	y	y	y	y	y	y	y
IS_AMNUL	y	y	y	y	y	y	y	y	y	y
IS_NUMERCI	y	y	y	y	y	y	y	y	y	y
IS_DIGIT	y	y	y	y	y	y	y	y	y	y
IS_UPPER	y	y	y	y	y	y	y	y	y	y
IS_LOWER	y	y	y	y	y	y	y	y	y	y
WORD_LENGTH										
FIRST-ONE-LETTER										
FIRST-TWO-LETTER										
FIRST-THREE-LETTER										
LAST-ONE-LETTER	y	y								
LAST-TWO-LETTER			y	y						
LAST-THREE-LETTER					y	y				
WORD-POSITION							y	y		
PUNCT-CNT										
DIGIT-CNT										
CAPS-CNT										
BIGRAM										
TRIGRAM										
WORD-RELATIVE-POS									y	y
PREV	y	y	y	y	y	y	y	y	y	y
NEXT	y	y	y	y	y	y	y	y	y	y
BROWN:5										
BROWN:10										
BROWN:15										
BROWN:20										
BRPWN: 30										
w2v										
w2v Twitter										
Accuracy	84.58	**84.67**	85.67	**86.14**	84.91	**85.48**	83.63	**84.34**	84.25	**84.34**

Features	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**
SENT_BEGIN	y	y	y	y	y	y	y	y	y	y
SENT_END	y	y	y	y	y	y	y	y	y	y
WORD	y	y	y	y	y	y	y	y	y	y
LCASE	y	y	y	y	y	y	y	y	y	y
IS_AMNUL	y	y	y	y	y	y	y	y	y	y
IS_NUMERCI	y	y	y	y	y	y	y	y	y	y
IS_DIGIT	y	y	y	y	y	y	y	y	y	y
IS_UPPER	y	y	y	y	y	y	y	y	y	y
IS_LOWER	y	y	y	y	y	y	y	y	y	y
WORD_LENGTH										
FIRST-ONE-LETTER										
FIRST-TWO-LETTER										
FIRST-THREE-LETTER										
LAST-ONE-LETTER										
LAST-TWO-LETTER										
LAST-THREE-LETTER										
WORD-POSITION										
PUNCT-CNT			y	y						
DIGIT-CNT			y	y						
CAPS-CNT			y	y						
BIGRAM					y	y				
TRIGRAM					y	y				
WORD-RELATIVE-POS										
PREV	y	y	y	y	y	y	y	y	y	y
NEXT	y	y	y	y	y	y	y	y	y	y
BROWN:5							y	y		
BROWN:10									y	y
BROWN:15										
BROWN:20										
BRPWN: 30										
w2v										
w2v Twitter	y	y								
Accuracy	84.63	**85.05**	84.06	**84.2**	**83.44**	82.92	83.02	**83.68**	83.63	**84.06**

Features	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**
SENT_BEGIN	y	y	y	y	y	y	y	y	y	y
SENT_END	y	y	y	y	y	y	y	y	y	y
WORD	y	y	y	y	y	y	y	y	y	y
LCASE	y	y	y	y	y	y	y	y	y	y
IS_AMNUL	y	y	y	y	y	y	y	y	y	y
IS_NUMERCI	y	y	y	y	y	y	y	y	y	y
IS_DIGIT	y	y	y	y	y	y	y	y	y	y
IS_UPPER	y	y	y	y	y	y	y	y	y	y
IS_LOWER	y	y	y	y	y	y	y	y	y	y
WORD_LENGTH										
FIRST-ONE-LETTER										
FIRST-TWO-LETTER										
FIRST-THREE-LETTER										
LAST-ONE-LETTER										
LAST-TWO-LETTER									y	y
LAST-THREE-LETTER										
WORD-POSITION										
PUNCT-CNT										
DIGIT-CNT										
CAPS-CNT										
BIGRAM										
TRIGRAM										
WORD-RELATIVE-POS										
PREV	y	y	y	y	y	y	y	y	y	y
NEXT	y	y	y	y	y	y	y	y	y	y
BROWN:5										
BROWN:10										
BROWN:15	y	y								
BROWN:20			y	y						
BRPWN: 30					y	y				
w2v							y	y	y	y
w2v Twitter										
Accuracy	83.68	**83.92**	84.15	**84.58**	83.87	**84.53**	**84.53**	85.24	85.71	**87.13**

Features	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**	**CRF**	**LR**
SENT_BEGIN	y	y	y	y	y	y	y	y	y	y
SENT_END	y	y	y	y	y	y	y	y	y	y
WORD	y	y	y	y	y	y	y	y	y	y
LCASE	y	y	y	y	y	y	y	y	y	y
IS_AMNUL	y	y	y	y	y	y				
IS_NUMERCI	y	y	y	y	y	y				
IS_DIGIT	y	y	y	y	y	y				
IS_UPPER	y	y	y	y	y	y				
IS_LOWER	y	y	y	y	y	y				
WORD_LENGTH										
FIRST-ONE-LETTER	y	y	y	y	y	y	y	y	y	y
FIRST-TWO-LETTER	y	y	y	y	y	y	y	y	y	y
FIRST-THREE-LETTER	y	y	y	y	y	y	y	y	y	y
LAST-ONE-LETTER	y	y	y	y	y	y	y	y	y	y
LAST-TWO-LETTER	y	y	y	y	y	y	y	y	y	y
LAST-THREE-LETTER	y	y	y	y	y	y	y	y	y	y
WORD-POSITION										
PUNCT-CNT	y	y	y	y	y	y	y	y	y	y
DIGIT-CNT	y	y	y	y	y	y	y	y	y	y
CAPS-CNT	y	y	y	y	y	y	y	y	y	y
BIGRAM	y	y	y	y	y	y	y	y	y	y
TRIGRAM	y	y	y	y	y	y	y	y	y	y
WORD-RELATIVE-POS	y	y	y	y	y	y	y	y	y	y
PREV	y	y	y	y	y	y				
NEXT	y	y	y	y	y	y				
BROWN:5										
BROWN:10										
BROWN:15										
BROWN:20					y	y				
BRPWN: 30		y	y	y			y	y	y	y
w2v	y		y	y	y	y	y	y		
w2v Twitter									y	y
Accuracy	86.61	**88.03**	86.85	**87.7**	86	**87.37**	**87.7**	86.94	**88.46**	86.54