CS564 ML

Assignment-3

Name: **M Maheeth Reddy**	Roll No.: **1801CS31**	Date: **09-Nov-2021**

In this assignment, the following has been done:

Implementation of a HMM model for PoS tagging on the Brown dataset using PoS tags as Hidden States and Tokens/Words as the observed variable.

Implementation of Viterbi algorithm for the Decoding Problem and to obtain the POS for the test data. We also report the f1 score, the precision and recall for all the POS tags.

Code for Emission Probabilities

```
def create_transition_prob(data_seq, states):
    transit_prob = {}

    for row in states.keys():
        transit_prob[row] = {}
        for col in states.keys():
            transit_prob[row][col] = 0

    for sample_seq in data_seq:
        for i in range(len(sample_seq)-1):
            transit_prob[sample_seq[i]][sample_seq[i+1]] += 1

    for row in states.keys():
        for col in states.keys():
            transit_prob[row][col] = (transit_prob[row][col]+1)/(states[row]+len(states))

    return transit_prob
```

Code for Transition Probabilities

```
def create_emission_prob(data_obs, data_seq, states, corpus):
    emiss_prob = {}

    for state in states.keys():
        emiss_prob[state] = {}
        for obs in corpus.keys():
            emiss_prob[state][obs] = 0

    for t in range(len(data_seq)):
        for w in range(len(data_seq[t])):
            emiss_prob[data_seq[t][w]][data_obs[t][w]] += 1

    for state in states.keys():
        for obs in corpus.keys():
            emiss_prob[state][obs] = (emiss_prob[state][obs]+1)/(states[state]+len(corpus))

    return emiss_prob
```

Precision, Recall and F1 Score for each Tag:

	Precision	Recall	F1 Score
PRON	0.74056072	0.6679715	0.70239565
VERB	0.94985435	0.56122449	0.70556414
DET	0.98251266	0.59026818	0.73747841
NOUN	0.97167022	0.65400984	0.78180437
.	0.90753425	0.6339239	0.74644602
ADJ	0.87960856	0.51273547	0.64783775
ADP	0.98903344	0.63594884	0.77413111
PRT	0.94654641	0.78521898	0.85836824
ADV	0.90529032	0.62609316	0.74024056
NUM	0.92633606	0.61215399	0.73716475
CONJ	0.99560117	0.48258706	0.6500718
X	0.00273186	0.8220339	0.00544561

Overall Accuracy:

Bigram Model for HMM	**60%**	**Conclusion**: The **Trigram model performs better** than the Bigram HMM model for PoS tagging, because the transition probabilities are calculated for more words.
Trigram Model for HMM	**90%**	