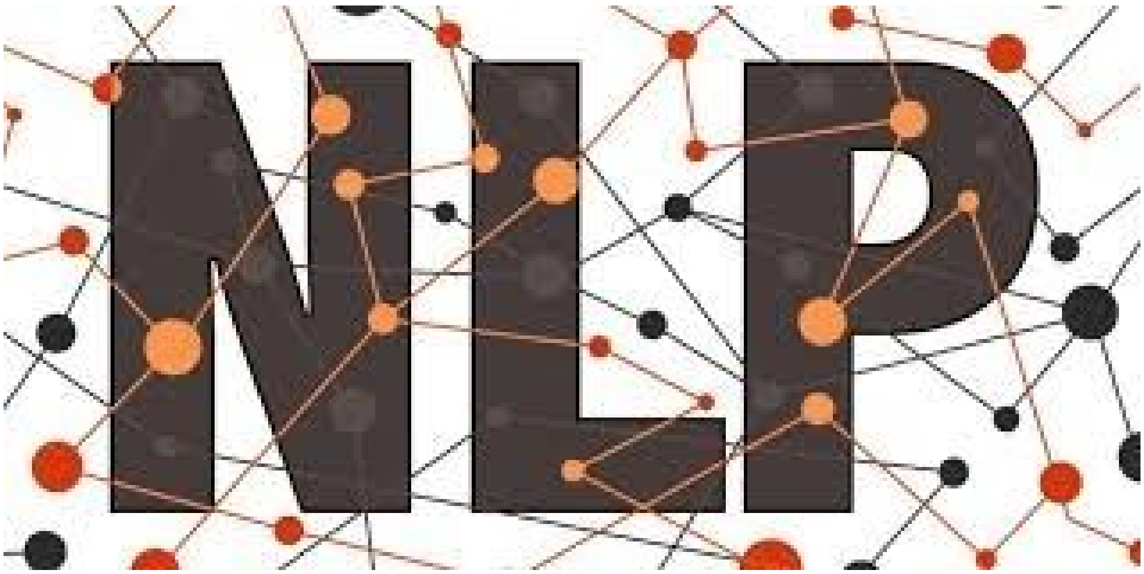
Natural Language Process(NLP) CSE 556

Assignment 2



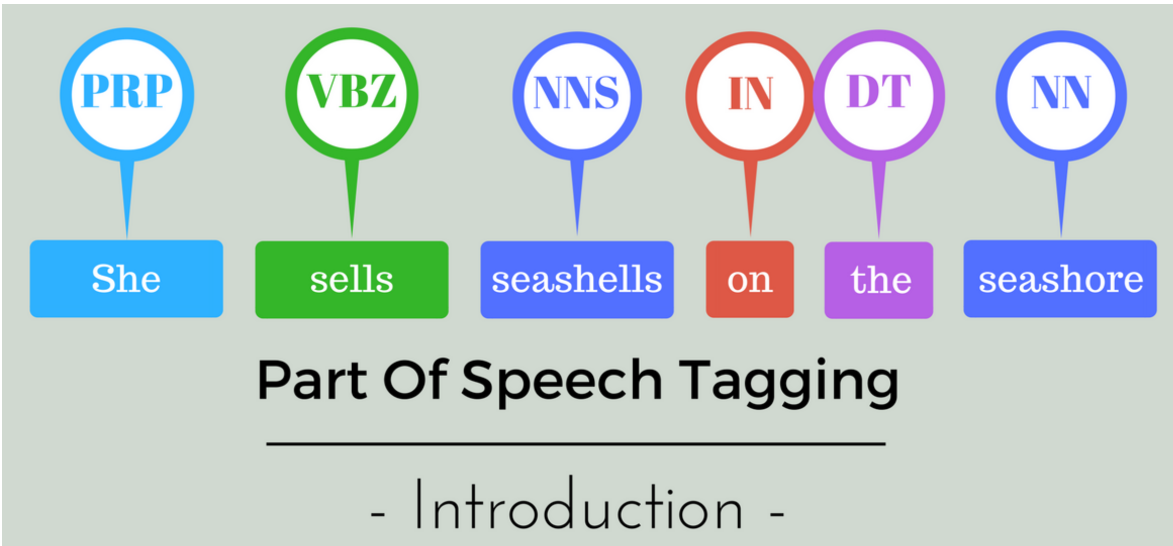
Lakshya(MT19067)

M.tech(3rd Sem)

Question 1

Introduction

Implementation of POS tagging using Hidden Markov Models(Viterbi algorithm).



Part of Speech Tagging (POS) is a process of tagging sentences with part of speech such as nouns, verbs, adjectives and adverbs, etc.
Hidden Markov Models (HMM) is a simple concept which can explain most complicated real time processes such as speech recognition and speech generation, machine translation, gene recognition for bioinformatics, and human gesture recognition for computer vision, and more.

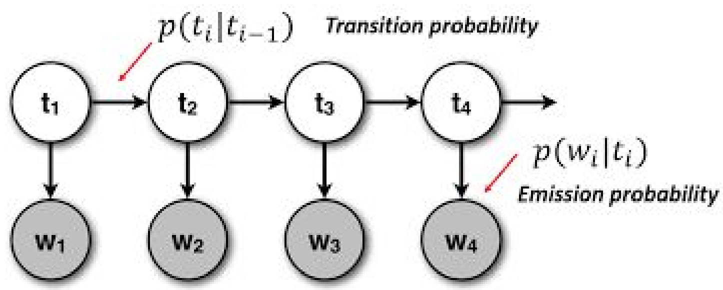
DataSet

The Brown University Standard Corpus of Present-Day American English (or just **Brown Corpus**) is an electronic collection of text samples of American English, the first major structured corpus of varied genres. This corpus first set the bar for the scientific study of the frequency and distribution of word categories in everyday language use. Compiled by Henry Kučera and W. Nelson Francis at Brown University, in Rhode Island, it is a general language corpus containing 500 samples of English, totaling roughly one million words, compiled from works published in the United States in 1961.

Methodology

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, especially in the context of Markov information sources and hidden Markov models (HMM).

A first-order Markov process is a stochastic process in which the future state solely depends on the current state only. The first-order Markov process is often simply called the Markov process. If it is in a discrete space, it is called the Markov chain.
The assumption of the Markov process may not be true in reality. But even it is not true, we can model extra states in the system to make it closer to the Markov process sometimes. In practice, the Markov process can be an appropriate approximation in solving complex ML and reinforcement learning problems.



a)

Output

With trigrams

```
Precision on the test data is with trigrams using k fold:  0.9172460779919319
```

using 3 fold validation the accuracy is 91%.

With bigrams

```
Precision on the test data is with bigrams using k fold:  0.8255214701927387
```

using 3 fold validation the accuracy is 82%.

Bigrams

1st fold

tag Precision on the test data is with bigrams at 1 iter is 0.48671096648942453

tag recall on the test data is with bigrams at 1 iter is 0.3636586419785581

tag f1_score on the test data is with bigrams at 1 iter is 0.3968509431903068

Sentence Precision on the test data is with bigrams at 1 iter is 0.8652853085463466

Sentence Recall on the test data is with bigrams at 1 iter is 0.8652853085463466

Sentence F1_score on the test data is with bigrams at 1 iter is 0.8652853085463466

2nd fold

tag Precision on the test data is with bigrams at 2 iter is 0.5249226474542196

tag recall on the test data is with bigrams at 2 iter is 0.4360374046645681

tag f1_score on the test data is with bigrams at 2 iter is 0.47636924831772

Sentence Precision on the test data is with bigrams at 2 iter is 0.8726047228091682

Sentence Recall on the test data is with bigrams at 2 iter is 0.8726047228091682

Sentence F1_score on the test data is with bigrams at 2 iter is 0.8726047228091682

3rd fold

tag Precision on the test data is with bigrams at 3 iter is 0.4126454895277855

tag recall on the test data is with bigrams at 3 iter is 0.37132704950013434

tag f1_score on the test data is with bigrams at 3 iter is 0.3908974472648825

Sentence Precision on the test data is with bigrams at 3 iter is 0.9136658235343335

Sentence Recall on the test data is with bigrams at 3 iter is 0.9136658235343335

Sentence F1_score on the test data is with bigrams at 3 iter is 0.9136658235343335

<u>Trigrams</u>

1st fold

```
tag Precision on the test data is with bigrams at 1 iter is 0.52654412

tag recall on the test data is with bigrams at 1 iter is 0.41860121

tag f1_score on the test data is with bigrams at 1 iter is 0.39213685


Sentence Precision on the test data is with bigrams at 1 iter is 0.88

Sentence Recall on the test data is with bigrams at 1 iter is 0.85678219754

Sentence F1_score on the test data is with bigrams at 1 iter is 0.930086432954
```

2nd fold

```
tag Precision on the test data is with bigrams at 2 iter is 0.519845231

tag recall on the test data is with bigrams at 2 iter is 0.4578326090

tag f1_score on the test data is with bigrams at 2 iter is 0.52072156


Sentence Precision on the test data is with bigrams at 2 iter is 0.86743209831

Sentence Recall on the test data is with bigrams at 2 iter is 0.89651239008

Sentence F1_score on the test data is with bigrams at 2 iter is 0.7612089653
```

3rd fold

```
tag Precision on the test data is with bigrams at 3 iter is 0.43556732119008

tag recall on the test data is with bigrams at 3 iter is 0.40677843290871691

tag f1_score on the test data is with bigrams at 3 iter is 0.419825
```

```
Sentence Precision on the test data is with bigrams at 3 iter is 0.915533890123
```

```
Sentence Recall on the test data is with bigrams at 3 iter is 0.87432211021
```

```
Sentence F1_score on the test data is with bigrams at 3 iter is 0.9235
```

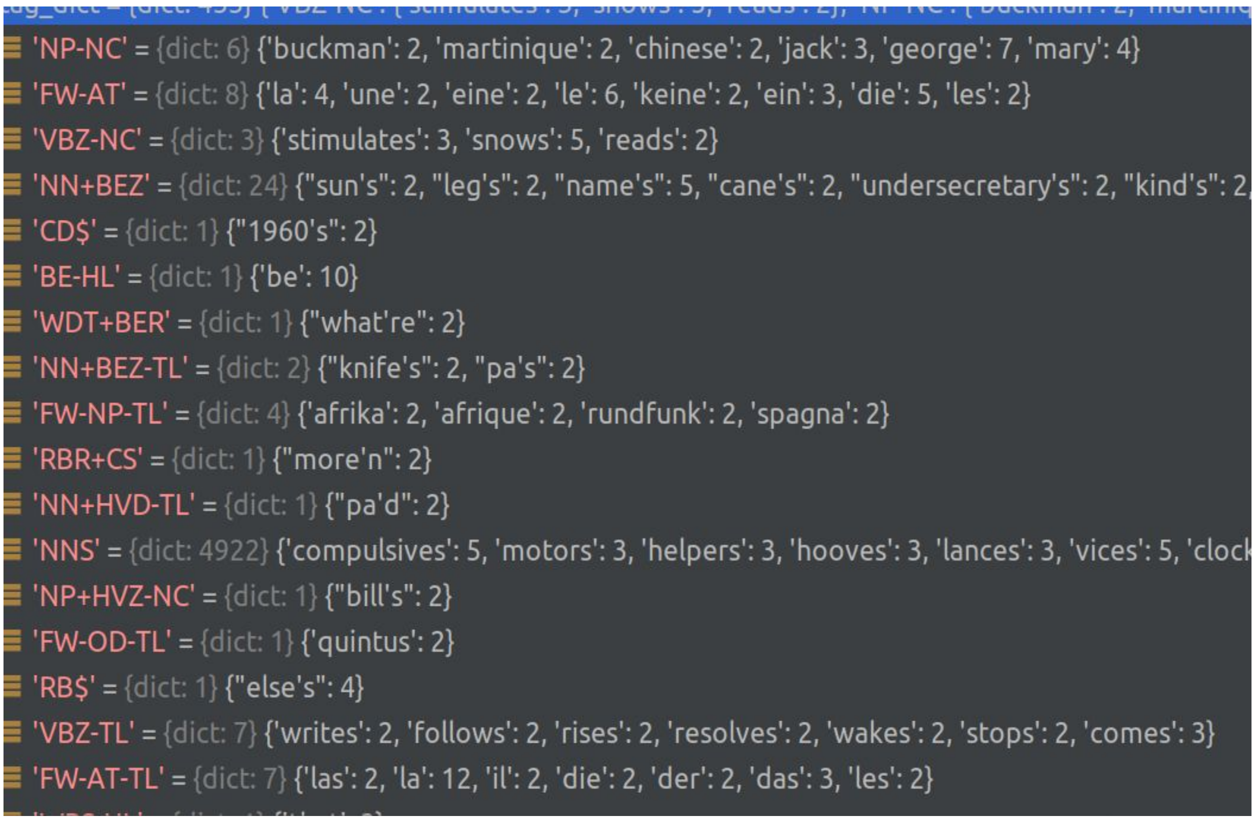
Confusion Matrix

```
[[327, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 99, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 13, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 83, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 241, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

Data Stats

```
Sentences : 55145
Tags: 366
words :41012
```

Question 2



'NP-NC' = {dict: 6} {'buckman': 2, 'martinique': 2, 'chinese': 2, 'jack': 3, 'george': 7, 'mary': 4}
'FW-AT' = {dict: 8} {'la': 4, 'une': 2, 'eine': 2, 'le': 6, 'keine': 2, 'ein': 3, 'die': 5, 'les': 2}
'VBZ-NC' = {dict: 3} {'stimulates': 3, 'snows': 5, 'reads': 2}
'NN+BEZ' = {dict: 24} {"sun's": 2, "leg's": 2, "name's": 5, "cane's": 2, "undersecretary's": 2, "kind's": 2,
'CD$' = {dict: 1} {"1960's": 2}
'BE-HL' = {dict: 1} {'be': 10}
'WDT+BER' = {dict: 1} {"what're": 2}
'NN+BEZ-TL' = {dict: 2} {"knife's": 2, "pa's": 2}
'FW-NP-TL' = {dict: 4} {'afrika': 2, 'afrique': 2, 'rundfunk': 2, 'spagna': 2}
'RBR+CS' = {dict: 1} {"more'n": 2}
'NN+HVD-TL' = {dict: 1} {"pa'd": 2}
'NNS' = {dict: 4922} {'compulsives': 5, 'motors': 3, 'helpers': 3, 'hooves': 3, 'lances': 3, 'vices': 5, 'clock
'NP+HVZ-NC' = {dict: 1} {"bill's": 2}
'FW-OD-TL' = {dict: 1} {'quintus': 2}
'RB$' = {dict: 1} {"else's": 4}
'VBZ-TL' = {dict: 7} {'writes': 2, 'follows': 2, 'rises': 2, 'resolves': 2, 'wakes': 2, 'stops': 2, 'comes': 3}
'FW-AT-TL' = {dict: 7} {'las': 2, 'la': 12, 'il': 2, 'die': 2, 'der': 2, 'das': 3, 'les': 2}

The tags which will be mostly classified wrong are the combined tags. The reason for this is the imbalance frequency of these tags in the dataset. These tags have very low frequency in the dataset and thus classifier fails to learn them correctly.

Question 3

Set of tags = $\{a_1, a_2 \ldots a_n\}$

Observation = $\{b_1, b_2 \ldots b_n\}$

$$P(A/B) = \frac{P(a_1, a_2, a_3 \ldots\}}{\{b_1, b_2, b_3 \ldots\}}$$

$$P(A/B) = P(a_1, B) \cdot P(a_2|B) \cdot P(a_3|B) \ldots$$

as knaive bayes states

$$P(X/Y) = \frac{P(x) \cdot P(Y/x)}{P(Y)}$$

Prior

likelyhood

marginal

we can ignore this as this is constant

for HMM on length 2 i.e., current state will depend on previous two states.

tags

words

$$P(XY) = P(X) \cdot P(Y/x)$$

$P(x)$ = prior / transitioning prob

$$P(x) = \prod p(x_i / x_{i-1}, x_{i-2})$$

$$P(X/Y) = \prod_{i \geq 2} P(Y_i / X_i) * P\left(\frac{X_i}{X_{i-2} \cdot X_{i-1}}\right)$$