

REPORT ON NATURAL LANGUAGE PROCCESSING ASSIGNMENT

Implement Hidden Markov Model - Viterbi Algorithm
for POS Tagging with Bigram assumption

SUBMITTED BY:
MAHAVEER CHOUHAN
B140809CS
mahaveerchouhan424@gmail.com
KRISHNA KUMAR
B140731CS
vermakrishna621@gmail.com

DESIGN:

- 1.code checks for a training file and then extract and enlist all the words and corresponding tags
- 2.save words and tag in indexed a list
- 3.splitting the pair of word and tag
- 4.applying the bigram assumption
- 5.handling the end of sentence case (only sign we used)
- 6.making the training runtime databse
- 7.read from the test file
- 8.save words and tag in indexed a list
- 9.splitting the pair of word and tag
- 10.iterate through the whole training set for thr tag if found saved or else a default tag of noun is assigned
- 11.splitting the pair of word and tag
- 12.the loop checks for hits of original with predicted
- 13.we print the 1-error-ratio to get the accoracy

Implementation

The language chosen for implementation was Python.

Input : two datasets (training and test) in word/tag format obtained from the Brown Corpus

Output :

- Accuracy(the ratio of hits to total tags)
- Actual Tag set (extracted from the input)
- Estimated Tag set (Predicted from the algorithm)

Experimentation Procedure

We took the whole brown corpus to train and used some randomly made sentences taken from the corpus to test.

RESULT:

Sr.No.	Train	Test	Accuracy
1	Brown Corpus	T1	0.92307692%
2	Brown Corpus	T2	0.71428571%
3	Brown Corpus	T3	0.66666666%

T1

```
Terminal
mah@cutu: ~/Desktop/nlp-final/1/edited
mah@cutu:~/Desktop/nlp-final/1/edited$ py Viterbi_POS.py
Fraction of errors (Viterbi): 0.0769230769231
Accuracy: 0.923076923077%
Tags suggested by Viterbi Algorithm: ['NNP', 'VBD', 'IN', 'VBD', 'NNP', 'VBN', 'TO', 'NNS', 'RB', 'CD', 'NN', 'IN', '.']
Correct tags: ['NNP', 'VBD', 'VBD', 'VBD', 'NNP', 'VBN', 'TO', 'NNS', 'RB', 'CD', 'NN', 'IN', '.']
```

T2

```
Terminal
mah@cutu: ~/Desktop/nlp-final/1/edited
mah@cutu:~/Desktop/nlp-final/1/edited$ py Viterbi_POS.py
Fraction of errors (Viterbi): 0.285714285714
Accuracy: 0.714285714286%
Tags suggested by Viterbi Algorithm: ['NNP', 'NNP', 'IN', 'NN', 'VBD', 'IN', 'PRP', 'MD', 'NNP', 'IN', 'DT', 'NNP', 'IN', 'NNP']
Correct tags: ['NNP', 'NNP', 'IN', 'NN', 'VBD', 'CS', 'PPS', 'MD', 'VB', 'IN', 'AT', 'NNP', 'IN', 'NNP']
```

T3

```
Terminal
mah@cutu: ~/Desktop/nlp-final/1/edited
mah@cutu:~/Desktop/nlp-final/1/edited$ py Viterbi_POS.py
Fraction of errors (Viterbi): 0.333333333333
Accuracy: 0.666666666667%
Tags suggested by Viterbi Algorithm: ['NNP', 'DT', 'NNP', 'NNP', 'CC', 'NNP', 'NNP', 'PRP$', 'NNP']
Correct tags: ['NNP', 'DT', 'NNP', 'VBD', 'CC', 'JJ', 'NNP', 'PP', 'NNP']
```

CONCLUSION:

The Implementation of Hidden Markov Model - Viterbi Algorithm for POS Tagging with Bigram assumption is done and the accuracy for different test sets were found.