Natural Language Processing Assignment-II (Task-1)

Natural Language Processing Assignment-2 report submitted to Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

by

Shatansh Patnaik

(20MA20067)

Under the supervision of

Dr. Saptarshi Ghosh

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Spring Semester, 2024-25

February 4, 2024

Contents

Chapter 1

Implementation of Task-1

1.1 Evaluations for Training Data

1.1.1 Calculations of Accuracy, Precision, Recall and F1 Score for the Training Data

- The Accuracy for the training data: 98.1268448784794%
- The Precision for the training data: 98.15698556577934%
- The Recall for the training data: 98.1268448784794 %
- The F1 Score for the training data: 98.13385851813972 %
- The number of words for which Smoothing was used: 0

Based on the training data, we can infer the following:

- The accuracy of the model on the training data is 98.13%. This indicates that the model's predictions are correct for approximately 98.13% of the instances in the training dataset.
- The precision of the model on the training data is 98.16%. Precision measures the proportion of true positive predictions among all positive predictions made by the model. A high precision score suggests that the model has a low rate of false positive predictions.

- The recall of the model on the training data is 98.13%. Recall, also known as sensitivity, measures the proportion of true positives that were correctly identified by the model out of all actual positives in the dataset. A high recall score indicates that the model can effectively capture most of the positive instances.
- The F1 score of the model on the training data is 98.13%. The F1 score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. A high F1 score indicates both high precision and high recall.
- The number of words for which smoothing was used is 0. Smoothing is a technique commonly used in natural language processing to handle unseen words in a dataset. The absence of smoothing indicates that all words in the dataset were encountered during training.

1.1.2 Assumptions

- We have used add-one smoothing over the entire training set because some of the values were coming very close to zero.
- We have used the add-one smoothing during the calculation of word-tag probability matrix and not during the calculation of tag-tag probability matrix.
- While applying the Viterbi Algorithm, we have assumed that the occurence of each tag at the beginning of the sentence is equally likely.

1.1.3 Problems faced during the implementation of the algorithm

1.1.3.1 Data Sparsity

In a few cases of word-tag probability matrix computations, the entries of the matrix had very low values. So this leaded to invalid final probabilities according to Viterbi Algorithm.

1.1.3.2 Solution

To take care of this we have used add-one smoothing over the training set so that all the entires in the probability matrix have non zero values and the Viterbi algorithm can compute the ideal sequence of POS Tags optimally.

1.1.3.3 Initialization of Viterbi Algorithm

For Viterbi Algorithm to provide better results is is very important to assume proper assumptions for the base case of the algorithm.

1.1.3.4 Solution

While applying the Viterbi Algorithm, we have assumed that the occurence of each tag at the beginning of the sentence is equally likely.

1.2 Evaluations for Test Data

1.2.1 Calculations of Accuracy, Precision, Recall and F1 Score for the Test Data:

- The Accuracy for the training data: 86.91460055096418 %
- The Precision for the training data: 87.99260138411482 %
- The Recall for the training data: 86.91460055096418 %
- The F1 Score for the training data: 86.4908720924295 %
- The number of words for which Smoothing was used: 509

Based on the test data we have:

- The accuracy of the model on the training data is 86.91%. This indicates that the model's predictions are correct for approximately 86.91% of the instances in the training dataset.
- The precision of the model on the training data is 87.99%. Precision measures the proportion of true positive predictions among all positive predictions made by the model. A high precision score suggests that the model has a low rate of false positive predictions.
- The recall of the model on the training data is 86.91%. Recall, also known as sensitivity, measures the proportion of true positives that were correctly identified by the model out of all actual positives in the dataset. A high recall score indicates that the model can effectively capture most of the positive instances.

- The F1 score of the model on the training data is 86.49%. The F1 score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. A high F1 score indicates both high precision and high recall.

- The number of words for which smoothing was used is 509. Smoothing is a technique commonly used in natural language processing to handle unseen words or rare occurrences in a corpus. The presence of 509 words for which smoothing was used suggests that the model encountered 509 previously unseen words during training.

1.2.2 Assumptions

- We have used add-one smoothing over the entire test set because some of the word-tag probabilites were zero and this would lead the entire Viterbi Algorithm to give zero probability.

- We have used the add-one smoothing during the calculation of word-tag probability matrix and not during the calculation of tag-tag probability matrix.

- While applying the Viterbi Algorithm, we have assumed that the occurence of each tag at the beginning of the sentence is equally likely.

1.2.3 Problems faced during the implementation of the algorithm

1.2.3.1 Data Sparsity

In a few cases of word-tag probability matrix computations, the entries of the matrix had very low values. So this leaded to invalid final probabilities according to Viterbi Algorithm.

1.2.3.2 Solution

To take care of this we have used add-one smoothing over the training set so that all the entires in the probability matrix have non zero values and the Viterbi algorithm can compute the ideal sequence of POS Tags optimally.