CENG463 NLP HW1 REPORT:

PART1:

In the preproccesing:

First I tokenize the words in every line and apply PortersAlgorithm for stemming.

A screen shot of a computer code

Description automatically generated

In the feature selection part:

For every preprocessed documentation I apply TfidVectorizer method which transforms a collection of text documents into a matrix where each row corresponds to a document, and each column corresponds to a unique term in the entire collection.

A computer code with text

Description automatically generated

The classification results for the test data ("philosophy\_test.txt","sports\_test.txt","mystery\_test.txt",...) are as follows:



As seen in the picture above all the test txts are classified correctly. For the SVC predictions I encountered some error during the SVC.fit part in the train() function and I couldn’t solve the problem. Anyway, my implementation structure for the SVC classifier is ready in the pythpn file and I believe If this error can be solved then the classifications are predicted automatically for the SVC (I add the SVC structure as comment in the main part).

Also, the accuracy is %100 for the NaiveBayes Classifier which I tempted to think I misunderstand the homework description but I implemented it as I understand. Since all the classification results are true there are no false positive or false negatives, therefore I do not calculate performance metrics.

PART2:

1) 2.COLUMN(UPOS): Universal Part of Speech. It stands for the word's allocated universal part-of-speech tag. These tags adhere to the Universal POS tag set, a condensed collection of part-of-speech tags with a goal of being language-neutral.

3.COLUMN(FEATS): This column includes morphological characteristics or further details regarding the form of the word. It may contain information about number, gender, tense, and other language-specific characteristics.

All the columns are used in the implementation:

A screen shot of a computer code

Description automatically generated

Column1 is used to access the token, Column2 is checked if it is PUNCT but not direcly used(Only for checking purposes), Column3 is used to access the label of that token if Column2 is not PUNCT.

2) For set1 the performance metrics are as follows:

A close-up of a computer screen

Description automatically generated

For the set2 :

A screen shot of a computer code

Description automatically generated

The performance metrics are as follows:

A white background with black text

Description automatically generated

For the set3:

A screen shot of a computer code

Description automatically generated

The performance metrics are as follows:

A close-up of a computer screen

Description automatically generated

3) Positive aspects of LR: With LR, we might see shorter training periods and reduced computational costs. If there are no significant sequential dependencies in your data, LR may still function rather well.

Negative aspects of LR: Losing the ability to represent complex sequential patterns could impair our performance, particularly if our task benefits from taking the complete sequence into account.

CRF could be more suitable for our task because I think our task relies on capturing intricate sequential dependencies rather than simplicity and efficiency.