NLP Fake News Detection

Improving Pre-Process function:-

Using the 90% training data and 10% of the validation data mean rank was found to be 4.5. Improving the pre-process function by removing stop words, lowercasing and punctuation marks through regex tokenizer obtained a mean rank of about 3.3125. Additionally adding porter stemmer which converts the tokenized words into their root form. Passing the modified pre-process function into training and validation data obtained a mean rank of 2.18 with a precision of 10 correct out of 16 and accuracy of 62%.

Improving Feature extraction

Used unigram,bi-gram, tri-gram found uni-gram to have a positive effect in improving the accuracy and mean_rank score. By applying unigrams obtained a mean rank of 1.97. Bi-gram and trigram are increasing the mean- rank score so considering uni-gram for linguistic feature extraction.

Uni-gram along with Pos-taggers obtained a mean-rank of 2.125, POS-taggers is increasing the mean-rank from 1.97 to 2.12 So, Pos-taggers is not providing a positive effect on the mean-rank.

Removing the most frequent words in the document using filter documents function. Most frequent words have higher document frequency; these words show very little effect in distinguishing the labels. Tried with max_df = 0.5,0.85,0.99 found 0.99 to be optimal. It worked positively on the mean_rank and obtained a mean_rank of 1.25 and a precision 13 correct out of 16 with an accuracy of 0.81.

Scikit-learn provides a method called K-best which extracts best features from the given data.The SelectKBest method selects the features according to the k highest score. It helps to remove unwanted data ultimately reducing the training time. After applying SelectKbest method the best mean rank was found to be 1.125 with a precision score 15 correct out of 16 with an accuracy of 0.9375.

Best mean rank obtained for the train data is 1.125 and Mean rank of **1.06** on the test data with a precision of 15 correct out of 16 and a test accuracy of 93%.

Analysing the similarity matrix

Modified preprocess and filtered feature matrix obtained a mean rank of 1.125 (15/16 are correct). At this point, we are displaying where there isn't a perfect match between the held-out character vector and the train character vector.

```
CHRISTIAN
Ranking for target 3 ROXY
*****
mean rank 1.125
mean cosine similarity 0.4202349024598916
15 correct out of 16 / accuracy: 0.9375
```

doc1	doc2	similarity
ROXY	CHRISTIAN	0.385565
ROXY	JANE	0.353831
ROXY	ROXY	0.349652

From the similarity matrix we can observe that all the target characters have the highest similarity to each other except 'Roxy'. We know that there must be a highest match between the 'Roxy' held-out vector and the 'Roxy' training character vector. From the output presented in the notebook we can observe that Roxy and Christian have the highest similarity score of 0.385 and Roxy and Roxy have a similarity score of 0.34. This explains to

us that the context of the lines spoken by Roxy from the held-out-data has a higher similarity with the context of the line spoken by Christian from the training data than the lines spoken by Roxy in the training data. Here we are extracting the words spoken by the respective characters and observing the common words between them. Extracting words from respective characters is clearly expressed in the notebook.

There is a highest similarity between roxy and christian because words spoken by these characters are mostly common words and they are in common. Some of the similar words spoken by these characters after applying ngrams are 'run', 'shop', 'ian', , 'lucki', 'marri', 'christian', 'fanci', 'think', 'bag', 'littl', 'bloke', 'might', 'proud', 'one', 'come', 'earth', 'shirt', 'thing', 'need', 'someth', 'today', 'top', 'grandios', 'tell', 'put', 'dinner', 'togeth', 'two', 'definit', 'bloke', 'dead', 'never', 'walford', 'got', 'lot'.

Add dialogue context and scene features

Creating the create_character_document_from_dataframe with the episode_scene column.

Here we are increasing the number of features by adding the context of the line spoken by the characters before and after the target character lines if they are in the same scene into character docs. Alternatively, we add the context of the spoken lines of the target character into the character docs. Mean rank obtained by using the episode_scene column as a feature obtained a mean score of 1.4325 and a precision of 10 out of 16 with an accuracy of 0.625.

Improve Vectorization method:

Improved create_document_matrix_from_corpus with TfIdf transformer where TfIdf-Term Frequency and Inverse Document frequency is a weighting scheme for common document terms.

L2 norm and smoothening the numerator and denominator by 1 of the IDF has slightly increased the mean -rank to 1.4375. Applying the Unigrams to the training corpus obtained a mean rank of 1.55, and bi-grams slightly increased the mean rank score. So, I'm keeping unigram features for further processing. Using the selectKbest method from sci-kit learn library with k (max num of features) as 1500 obtained a mean_rank of 1.75. On decreasing the K (number of characteristics) to 800 obtained a mean rank of 1.375 with a training accuracy of 68% and a precision of 11 correct of 16. Here modified pre_process and filtered feature matrix through min and max document frequency is used.

Best Mean Rank obtained for the train data is 1.375

Run on final test data

Training the train data with 400 character spoken lines and testing the test data of 40 character spoken lines. Here all the lines are preprocessed by removing stop words, lowercasing words and all the root words are obtained. These preprocessed words are filtered using the create document matrix from corpus. Extracting an episode-scene column by combining episode and scene from the test data.

Best Mean rank obtained from the modified test data is 1.375 with an accuracy of 69% and a precision of 11 correct out of 16.