Bhaskar Yuvaraj

TEXT CLASSIFICATION USING NLP

The text classification is done for the tweets to classify as tweets either associated with terrorism or not. The data was been analyzed and trained as per the following steps:

1. The data was imported to python platform (Jupyter Notebook)
2. The libraries used for this project are:
    - Pandas
    - Numpy
    - Nltk
    - Sklearn
    - Matplotlib
    - Re
    - Preprocessor
    - Transformers
    - Tensorflow

3. The basic data analysis was done to find for any missing values and data types of the data-frame.
4. The data is extracted from given data which had labels to use it for training.
5. The tweets in the training data contains of unwanted characters like #,*,@,:. These were cleaned and the stop words were removed. Stop words are common words like is, the, a which frequently occurs in a sentence. An functions were built to perform these operations. [text_cleaning(), get_text()]
6. I have checked for the count of the labels (0s and 1s) in the train data. It was found to be equally distributed of 2000 each and hence up-sampling/down-sampling is not required.
7. In this step, I had decided to work on two different algorithms: SGD classifier and Bert model. Just an overview of these two algorithms below:
    - Nj SGD classifier: This algorithm uses two approach, countVectoriser and Tfidf transformer. The cleaned data has to be converted to numerical form to apply in the model. The Bag of words does splitting of words into tokens and counts the frequency of the words in a sentence and gives output as matrix with text in rows and vocab in the column. The Tfidf transformer assigns the weights for the frequency of each word. Hence Sci-kit learn pipeline with a SGD classifier is used for training.
    - Bert model: This model uses Bert Tokenizer which splits the words into a list of tokens which are already available in the vocab in Bert base. The important parameters for the model is
        - Input ids (batch size and sequence length)
        - attention masks: In this step the masking is done to the text by padding to make it equal in length.
        - labels is prepared.

The tokenizer encodes the text which are understood by the model. For Bert, it is recommended to start with pre trained model to avoid over fitting and further training the model on smaller set which is known as fine tuning. The training data was divided to test, train data with test size of 30%.

8. The training data was divided to test, train data with test size of 30%.
9. The SGD classifier model was implemented to return the f1 score of 88.3%.
10. I came across Bert model, which is trending with high accuracy. It uses the concept of deep learning to increase the accuracy. So wanted to try out the accuracy by implementing it to this data. I went ahead and implemented it by splitting the training model to test and train with test size of 20%.
11. The loss and optimizer function were added to the Bert model in order to reduce the loss function and increase the accuracy
12. The model was trained with batch size of 300 with 6 epochs which sums up to just half of training data which yields the accuracy with f1 score of 78.32%.
13. Due to time constraints with training the entire training sample data, I was able to train only 50% of train data to yield a score of 78.32% with loss function in reducing rate. I am expecting an accuracy of greater than 90% upon training with batch size of 300 and epoch of 12. Which would consume greater time of around 7-8 hours
14. Considering all these I choose SGD classifier with 88% accuracy and predicted the labels for test data.
15. I referred few blogs like Swatimeena, Kaggle, vickdata regarding the algorithm models.

Conclusion:

The project was done with implementing two different algorithm to check for high accurate results. SGD was chosen with f1 score of 88%. I also believe that Bert also would yield high accuracy in this case if the train data is completely trained. With 50% training of the model with time constraints it yielded an f1 score of 78% and I expect it to provide an accuracy of greater than 90% upon complete training.