Report

Task: Text Classification
No of classes- 12

Procedures which can be followed during preprocessing:

1. Consider words
2. Consider k-grams
2. Consider characters (in other words 1-gram)

First case: Considering words doesn't seem to be a good choice here. I have tried **LSTM** model after using the keras tokenizer. The model performed very bad on the validation data.

Second case: For k-gram, many classification model(For eg- Naive Bayes) can work. (k>=3). It seems to be a good choice for this type of data.

Third case: We can try to learn about the context from the characters. This also seems to be a suitable approach.

I have focused on learning the context based on the characters.

Preprocessing:

Need to remove the stopwords. (Those words which have occured frequently in most of the documents) .
[Due to some reason I couldn't do this step. I didn't have access of GPU intially. Later I didn't get the time to do this simple step. Accuracy will surely increase if we do this step]

Other preprocessing works were done to convert the data in the model specified format.
First we need to convert the text to some numbers . The data was converted to a list of lists which contain the character information. Padding is used to make each input of the same length.

Total unique characters considered- 69
For each input text,maximum length(no of characters) considered - 500

Train-Test Split:

To check how the model works, train-test split was done in the train data. Later for prediction pupose in the final test data, whole train data was used.

Model: Convolutional Neural Network

Optimizer: ADAM
Activation function in ouput layer: Softmax
Loss Function: Categorical Cross-entropy

Like other typical CNN models, convolution, pooling, fully connected layers were used.
For NLP tasks **Conv1D** layer is used for obvious reasons.

During fitting batch of size 256 is used. We can vary the batch size.

Evaluation: Before prediction on the final test data we need to validate our result on some validation data. That is why train-test split was done initially before implementing the final done. Accuracy on the validation set was 70-75%

Prediction: A final model was fit on the whole training data. Final score on the test data- 0.715

Scope:

Removing the stop words is an important step in this task. The same procedure as above can be followed but after removing the stopwords. This will definitely increase the accuracy.

Another thing which can be done is to consider both words or k-gram and the characters together. We can concatenate both these input and then pass it to some model.