**IMPLEMENTATION DOCUMENT**

Steps took in implementing this project

1. Create a conda environment

Created a conda environment “m\_cooper” and installed all the required python libraries

1. Importing Libraries

Imported all the required libraries, classes and functions for further process

1. Load Data set

Load the context.csv using pandas and analyze the data for each features, check shape, null values if present, unique classes in target column, number of statements in each category,

1. Created a function “text\_preprocess” to clean the statement data by removing stopwords and symbols(non alpha numeric characters)
2. Tokenize the statements and add padding to balance number of words in each statements to fixed number(MAX\_SEQUENCE\_LENGTH=150)
3. Created independent(X) and dependent(Y) feature tensors using padding sequences
4. Split the data set to train and test sets
5. Created a Keras Sequential model and added the embedding layer
6. Added LSTM layer with 100 units and dropout ratio of 0.2 to avoid overfitting
7. Added an output layer with Dense layer with 11 units(11 is the total number of classes), with activation function as “softmax” as it is a multiclass classification
8. Compiled the model with categorical crossentropy as the loss and adam as the optimizer. Checked the model summary.
9. Train the model for 10 epochs with early stoping by monitoring validation loss and with 32 as batch size. 10% of training set will be used as validation set.
10. Checked the performance of the model created
11. Ploted graphs of loss and accuracy of the model for both training and test set
12. For predicting topic for a new statement:

Get the statement, tokenize the statement and then add padding to make a padded sequence and used it for predicting. Predicted class will be the class with high probability

1. Save model as a pickle file
2. Create a Flask app with an index.html for user interface where user can enter statement and get the predicted result