Call Intent Classification

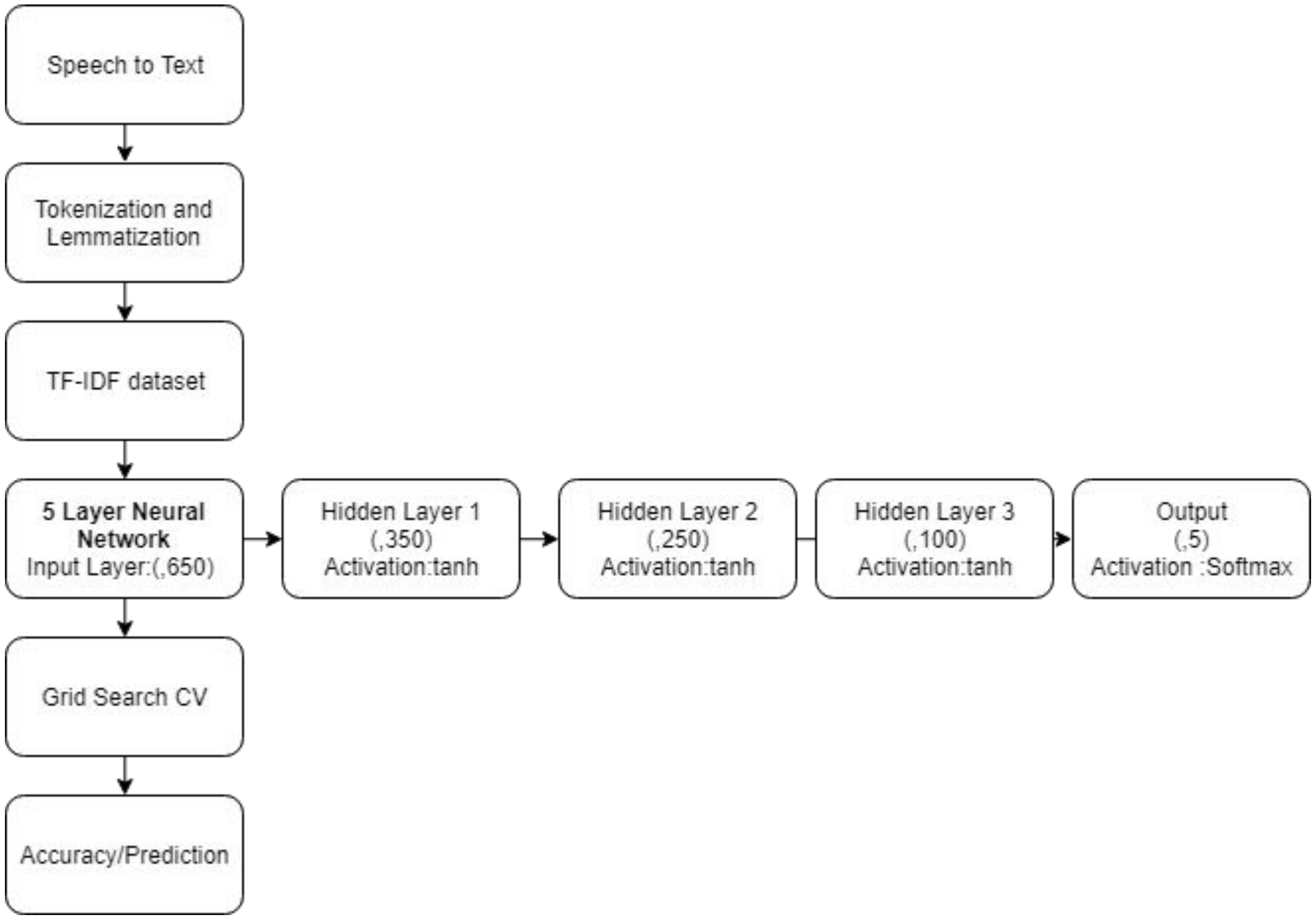
Approach

Source and Nature of data: Due to the lack of data, we asked some of our classmates to write about 10 lines for each category considering possible scenarios so as to get the most important features useful in classifying the conversations into categories.

Pre-processing approach: The sample audios were converted into text using Google Cloud Platform's Speech to Text API

Technology used: Neural Networks, Speech to Text, Lemmatization and Vectorization of text.

Solution Architecture



Speech to Text

Tokenization and Lemmatization

TF-IDF dataset

5 Layer Neural Network
Input Layer:(,650)

Hidden Layer 1
(,350)
Activation:tanh

Hidden Layer 2
(,250)
Activation:tanh

Hidden Layer 3
(,100)
Activation:tanh

Output
(,5)
Activation :Softmax

Grid Search CV

Accuracy/Prediction

Training Methodology

Performing tokenization/Lemmatization on the training sentences.

Use of the feature extraction/Vectorization algorithm(*Term Frequency – Inverse Document*) to encode the parsed tokens as integers or floating point values for use as input to a machine learning algorithm.

The data is trained using a 5 layer Neural Network consisting of 3 hidden layers.

The tanh activation is applied on the hidden layers and softmax function on the output layer for the purpose of classification.

The class which is assigned maximum probability by the softmax function is considered as the predicted class.

Testing Methodology & Accuracy attained during testing

Testing method:
We applied GridSearchcv to find the hyperparameters providing maximum accuracy.
Plotting the epoch Vs accuracy graph helped us find the optimum epoch and Batch size for obtaining minimum loss and maximum accuracy.

The first round of validation was performed on the X_Test set obtained by splitting our training data in the ratio 0.25.It gave us an accuracy of 97.67%

In the next step we converted the sample audios provided from Speech to Text using
Google Cloud API obtaining an accuracy of 60% (loss = 4.56) and
Microsoft Azure Speech API obtaining an accuracy of 80%

Steps to run the code

1. Create a new folder
2. Copy the Docker File and “Not_So_Bayesic.zip” into the folder
3. Extract the zip file in the same folder
4. Run the following command in the same folder
   1. docker build --tag= abcd .   (space and dot after abcd is a part of the command)
   2. docker run abcd

Summary

1. Hence, we have implemented a Neural Network model that classifies the recordings of calls into the given categories.
2. The accuracy of the entire model is affected due to the inaccurate conversion of Speech to Text API, the background noises and the clarity in the voice of the speaker.