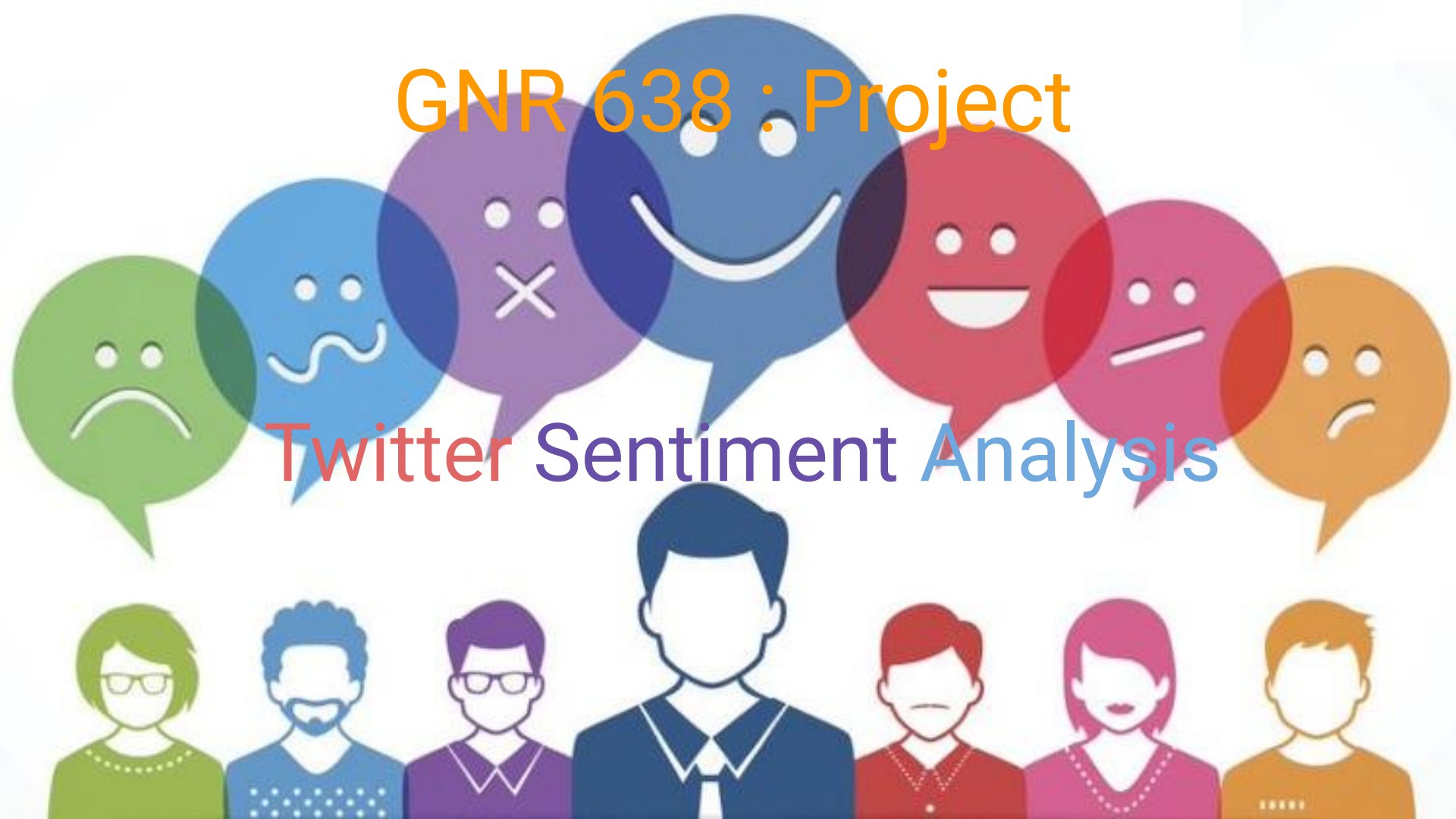
GNR 638 : Project



Twitter Sentiment Analysis

Background and Motivation

- Recent Communications:
  - Text-Messages
  - Tweeting
- People share sentiments about what is going on around the World on social Media.
- Issues with social Media:
  - Trolling
  - Hate Speech
  - Social Media Bullying
- Need of a system able to detect Negative Tweets to make social media better and bully-free place.

Problem Statement

To implement Twitter Sentiment Analysis using different models like:

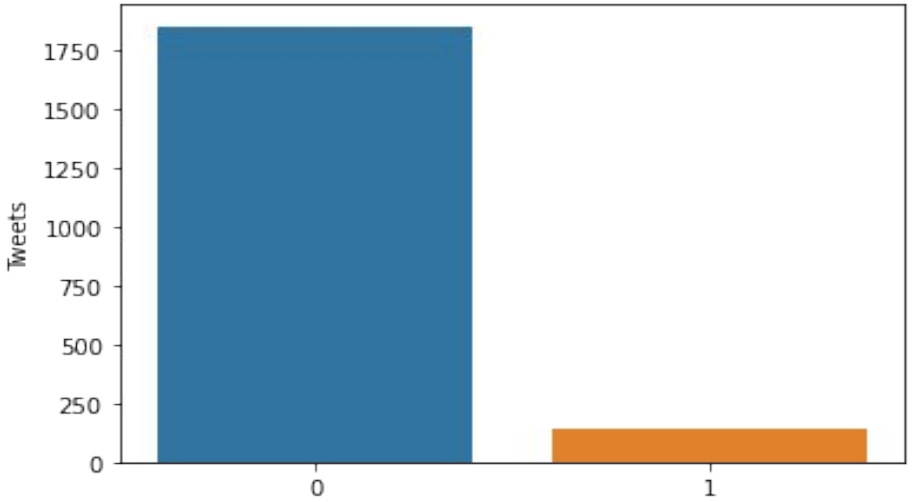
- ❏ Bag of Words Model
- ❏ TF-IDF Model
- ❏ Naive-Bayes Model
- ❏ RNN model Stacked with LSTM
- ❏ CNN-LSTM Model

And to compare the accuracies of the models to find out which model gives best results for Twitter Sentiment Analysis.
https://arxiv.org/pdf/1704.06125v1.pdf

Dataset

We used general Dataset “train_E6oV3lV.csv” which is easily available on the web. This dataset contains 31962 tweets with the columns being `['id', 'label', 'tweet']`. Label represents Sentiment of a Tweet. Label 1 means tweet is Positive and Label 0 represents Negative tweet.

Distribution of Tweets:



Dataset Preview

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation

Architecture of LSTM-CNN Model

Introduction

CNN - LSTM architecture involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction

Architecture

CNN layers on the front end

followed by LSTM layers with a

Dense layer on the output.

Result

CNN extracts better features thus the accuracy increases

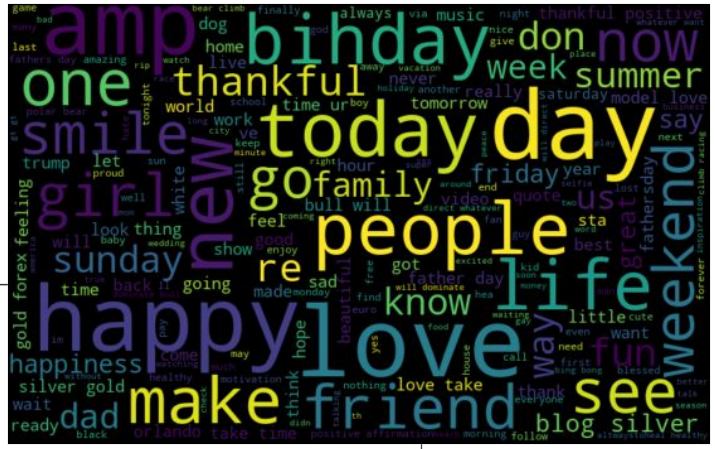
Proposed Modifications

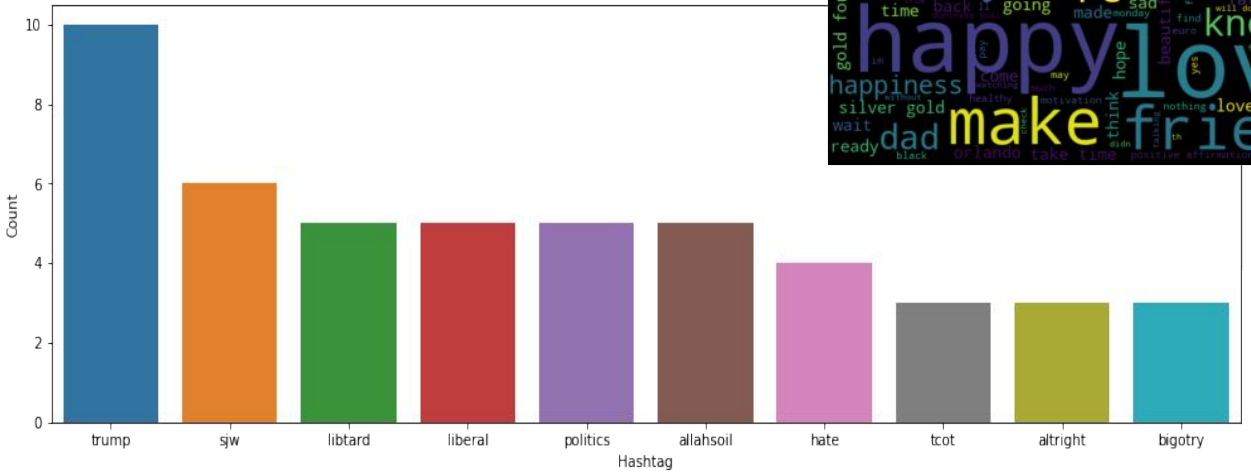
- Preprocessing:
  - Visualisation of Common Words using CloudWord
  - Visualisation of common words used in Racist/Sexist and Non Racist/Sexist Tweets using CloudWord and
  - Plotting frequency of these common Words in Bar-Plot.

Proposed Modifications

- Preprocessing:

Common Words
in all Tweets





Common words in
Negative Tweets

Proposed Modifications

- Different Models implemented to check Accuracy along with CNN-LSTM Model:
  - Bag of Words Model:
    - After converting tweets into vector of word Counts, Splitted dataset into 7:3 and fit training data to the model Predicted sentiments of a test set and used f1 score as a metric to find accuracy.
  - Naive Bayes Classifier
    - Build on the Vocabulary and frequency based on the Data. Build Counter function to make frequency and train data set bag of words set.Prediction function returned probability of Positive and Negative Class
    - OutputDecision function predicted whether tweet is Racist or not. And used accuracy_score metric to find accuracy.

Proposed Modifications

- Different Models Used to check Accuracy along with CNN-LSTM Model:

  - RNN model stacked with LSTM:
    - LSTM model has multiple LSTM layers
    - Output of Sequence of Vectors of previous LSTM layer is used as input to next LSTM layer.
    - For this we used a smaller dataset with 2000 samples to find accuracy.Used accuracy_score metric to find accuracy.

Result

Accuracy of all the models along with their process/Architecture:

- ★ Bag of Words Model
- ★ TF-IDF Model
- ★ Naive-Bayes Model
- ★ RNN model Stacked with LSTM
- ★ CNN-LSTM Model

BAG OF WORDS MODEL

Accuracy : 94.2%

Converted a collection of Tweets into a vector of term counts using CountVectorizer. Only the counts of words matter. disregarding grammar and even word order but keeping multiplicity

trained model using Logistic Regression.

TF-IDF MODEL

Accuracy : **94.37%**

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Collection of Tweets into a vector of term counts is carried out using TfidfVectorizer. Terms with higher weight scores are considered more important

NAIVE BAYES Classifier

Accuracy : 67.61%

Naive Bayes classifier uses the Bayes Theorem. It predicts membership probabilities for each class as the probability that given data point belongs to a particular class. We assume conditional independence between features. The class with the highest probability is considered as the most likely class.

RNN Sequence Model stacked with LSTM

Accuracy : 93.3%

An LSTM model comprised of multiple LSTM layers. Each LSTM layer outputs a sequence of vectors which will be used as an input to a subsequent LSTM layer. This hierarchy of hidden layers enables more complex representation of our text data, capturing information at different scales.

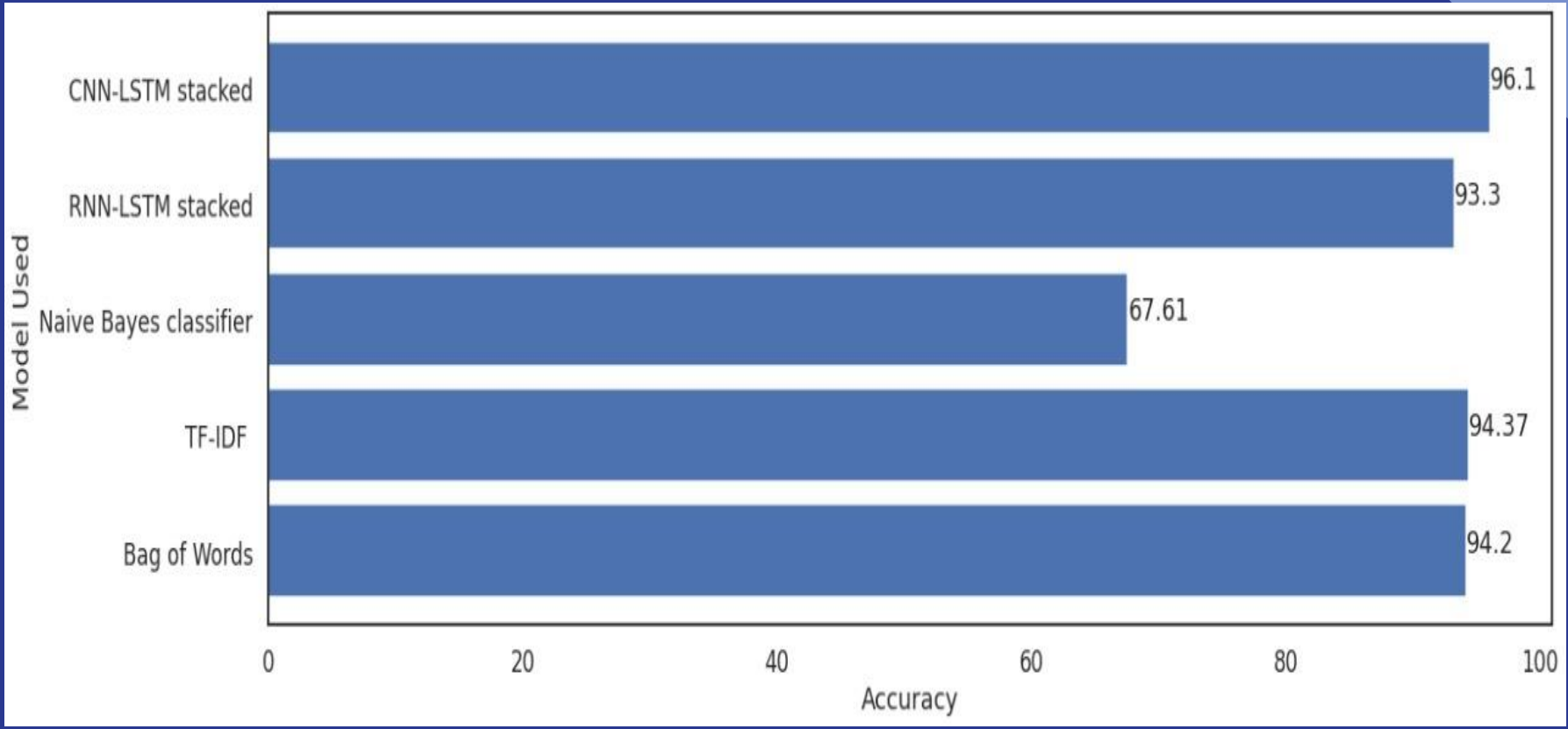
RNN stacked with LSTM model gives accuracy of 93.3%.

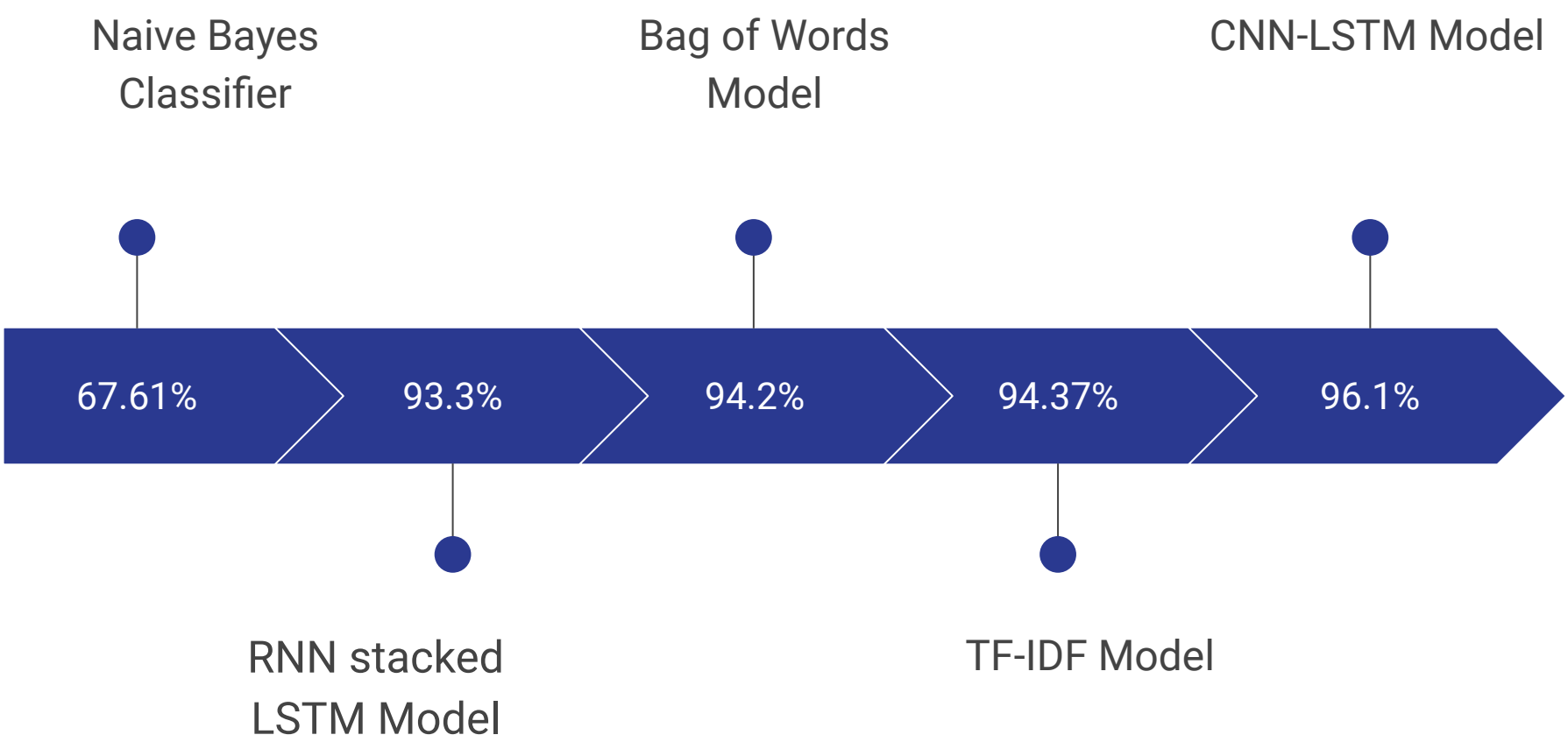
CNN-LSTM MODEL

Accuracy : 96.1%

Feature extraction is done using Convolutional Neural Network followed by LSTM to give sequence prediction. Better feature extraction of CNN gives highest accuracy of 96.1%.

Result


CNN-LSTM stacked
RNN-LSTM stacked
Naive Bayes classifier
TF-IDF
Bag of Words
96.1
93.3
67.61
94.37
94.2
Model Used
0
20
40
60
80
100
Accuracy


Naive Bayes Classifier
Bag of Words Model
CNN-LSTM Model
67.61%
93.3%
94.2%
94.37%
96.1%
RNN stacked LSTM Model
TF-IDF Model

The team and Contributions

Shubham Namdev Kamble	Abhishek Anand	Koushikey Chhaparia
180020105	18D070001	204310004
Cleaning and pre-processing of data , presentation making, Video Making	Bag of words , TF-IDF model , Naive Bayes classifier, Video Making	RNN stacked LSTM Model , CNN-LSTM Model, Video Making , Video Editing

Thank you