Speech and Opinion Recognition from a Conversation

Tavishi Priyam, AMJ Muthukumaran, Himanshu Vinayak

Abstract— Speech Recognition is an interdisciplinary technique used to convert spoken language into text. It is a sub domain of computational linguistics and can be implemented using Machine Learning and Deep Learning Algorithms. Opinion Mining or Sentiment Analysis is a process which enables identifying opinions expressed by an author in a piece of text computationally. This opinion refers to the polarity of the expressed opinion, i.e. positive or negative. Through this research work, we aim to combine these two natural language processing techniques and devise a system that can take speech as the input and determine the sentiment behind the speakers' words. The subject of the speech input may vary but the end goal is to recognize whether the attitude of the speaker towards the subject was positive or negative. The input will be converted to text and this text will then be classified using several different machine learning techniques. These include Naïve Bayes' Classifier, Support Vector Machine, Logistic Regression and Decision Trees. After classification, the results for the three classifiers will be predicted and compared. Future scope of the project includes creating an ensemble of these classifiers to get better accuracy and precision of determining the sentiment of the speaker.

Keywords— Sentiment Analysis, Machine Learning, Natural Language Processing, Opinion Mining, Speech Recognition

I. INTRODUCTION

Sentiment Analysis or Opinion Mining is a natural language processing technique that proves to be extremely beneficial in classifying textual data such as reviews. It involves training a model so that the machine can learn from the text as it goes and recognise the sentiment or the polarity of the text as a whole. There are numerous machine learning algorithms that help in classifying textual data. This technological advancement plays a great role in business development as it helps the companies understand the needs and opinions of customers towards their products.

Speech Recognition is a field wherein speech input is analysed and computationally converted to text. It is a sub domain of computational linguistics and is used extensively is the world of technology.

In recent times, a multitude of gadgets and devices have been developed that provide an enhanced interactive experience using speech recognition and artificial intelligence techniques. Some popular applications of speech recognition include voice assistants such as Siri, Google Assistant and Alexa. Individually, both these techniques have been the main focus of many researchers over the past few years and a lot of progress has been made in terms of the accuracy and applications of these techniques. In this research work, we attempt to integrate these techniques and expand the possibilities of using them.

II. LITERATURE SURVEY

Sentiment Analysis and Speech Recognition are areas that are currently being widely researched by computer scientists all over the world. Many researchers have delved into various machine learning algorithms that can be used to analyse and classify natural text. After extensively researching the existing works, a survey was conducted in order to gain insight on which algorithm works best for pre-processing and analysing textual data.

In the research work proposed by Shweta Rana and Archana Singh [1], a movie dataset containing 1000 positive and 1000 negative reviews is used. The text is processed, which includes conjuring content, content changes and cursor development followed by the removal of affixes using the Porter-Stemmer algorithm. For modelling, two different techniques are used- Linear Support Vector Machine (SVM) and Naïve Bayes' (NB) Classification. Both these machine learning algorithms produce decent results but better accuracy was observed using the former. This method proposed by Santosh Kumar K L, Jayanti Desai and Jharna Majumdar [2], involves three different classification techniques- Naïve Bayes Classifier, Logistic Regression and Senti WordNet. They concentrate on mining the reviews from major e-commerce websites and then creating a system which allows the user to choose the desired machine learning algorithm for classification of the text. In the research work [3], authors proposed a method following SACP framework. The data is collected from e-commerce websites like Amazon.com. Pre-processing is performed on the data to remove stop words, punctuation marks, whitespaces, digits and special symbols. Then, the relevant features are selected for modelling. An additional step involves examining correlation between the different features. Finally, sentiment analysis is performed using SVM. Authors Karthikayini T and N.K. Srinath [4], propose a new algorithm - SENTI for classification of text data based on overall rating of reviews and computation pf partial sentiment score. The data used are reviews extracted from a renowned e-commerce website.

Revised Manuscript Received on April 30, 2020.

***** Correspondence Author

Tavishi Priyam*, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur.

AMJ Muthukumaran, Assistant Professor, Department of Computer Science & Engineering at SRM Institute of Science and Technology, Chennai.

Himanshu Vinayak, Mobile Developer, Department of Computer Science & Engineering at SRM Institute of Science and Technology, Chennai.

It is observed that the proposed algorithm SENTI outperforms the other two classification techniques in terms of accuracy and precision.

In [5], equal focus is given to term weighting and classification techniques. Initially, a dataset comprising of 1000 reviews is selected.

The text is pre-processed and lexicon is created. As an additional step, weights are assigned to each word of the sentence depending on how much that word has contributed towards the polarity of the sentence/author. This is done using TF-IDF algorithm.

Following this, the sentiment classification is performed using three different classification techniques – Decision trees, K-Nearest Neighbours and Support Vector Machine. Without lexicon, SVM has the highest accuracy of 91.00% while with lexicon, KNN shows the highest accuracy of 92.67%.

This research performed by Siti Ernawati, Eka Rini Yulia, Frieyadie and Samudi involves working on a dataset comprising of 200 fashion reviews.

These reviews are pre-processed and a new method of feature selection using Genetic algorithm is proposed. This is done in order to increase the accuracy of simple NB classifier. Finally, the results are evaluated for NB classifier with and without the genetic algorithm.

It is observed that the proposed method of combining both algorithms provides a higher accuracy of 87.50% whereas a standalone NB classifier provides accuracy of merely 68.50%.

The researchers of [6] and [7] both propose ensemble classification methods. An ensemble classifier is one that combines the results of multiple individual classifiers. It is observed that ensemble classifiers perform better than simple classification techniques.

In [6], the authors experimented with an ensemble of Naïve Bayes, Maximum Entropy and SVM and achieved 90%accuracy.

Additionally, in [7], Baseline was also added to the previously mentioned classifiers.

The ensemble outperforms all the individual classifiers in terms of accuracy and precision.

Based on observations from all the papers, it seems that Support Vector Machine and Naïve Bayes Classifier are the two machine learning algorithms that provide the best results when working with textual data.

III. PROPOSED WORK

The methodology proposed in this paper involves devising a system that takes speech from a conversation as the input. The speech input is then converted to text using speech recognition tools.

Subsequently, the text is passed into the trained classification model.

The output will be *positive* or *negative* based on the polarity of the input.

Figure1 depicts the basic architecture of the entire system. The detailed architecture includes the various techniques used to convert speech to text and classify the text.

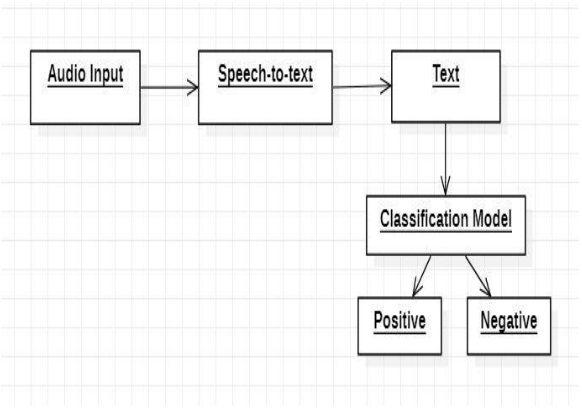

Figure 1. Architecture

For the conversion of the speech input to text, the python package Speech Recognition is used. This package offers a wide range of speech recognition APIs such as CMU Sphinx, Wit.ai and Google Cloud. In this research, we have worked with CMU Sphinx and Google speech API. CMU Sphinx has an added advantage since it is able to function without an active internet connection.

For the classification of sentiment, a variety of different machine learning algorithms are used and a comparative study is performed on the results. The algorithms we are working with are Support Vector Machine (SVM), Naïve Bayes Classifier (NB), Logistic Regression and Decision Trees.

These models haven been chosen after studying the effects of each individually on textual data. The system has been divided into two main modules- Speech Recognition and Sentiment Analysis. These models are further divided into sections.

A. *Speech Recognition*

In order to perform the speech to text conversion, the input physical signal is converted to an electrical signal using a microphone. This file is saved in the .mp3 format. The python module *pydub* allows manipulation of audio files. Using the library *AudioSegment* from this module, we convert the mp3 file to wav format.

Once converted, we have all the speech input ready to be converted to text. The python package Speech Recognition is the resource being used to convert the *wav* file into text. It allows access to a wide range of speech recognition APIs. For our purpose, we make use of the CMU Sphinx tool. It is an open source speech recognition tool developed at the Carnegie Mellon University.

The first version of Sphinx made use of Hidden Markov Acoustic Models along with n-gram statistical language model. The subsequent versions have been modified to produce better results. We make use of this tool by invoking the *recognize_sphinx* command in python.

Under the Speech Recognition package, we can also access the Google speech API using *recognize_google*. It makes use of deep-learning neural network algorithms to convert the audio input to text. We make use of this tool to convert the speech to text.

Making use of the two different tools allows greater accuracy and increased adaptation to varying accents and dialects.

Once the input has been converted to text, we save the text in a python variable and plug it as input to the trained classification models.

B. *Sentiment Analysis*

The sentiment analysis module involves three different sections that deal with pre-processing the text, training the classification models and testing the data.

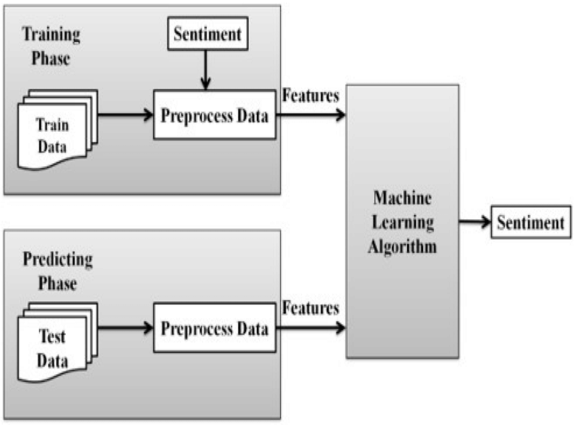

Figure 2. Architecture – Sentiment Analysis

1) *Pre-processing and visualising the data*

The dataset we are working with to train the classification model is available on Kaggle under the name 'Amazon Fine Food Reviews'. It consists of 568,454 reviews about products available on the popular e-commerce website. This dataset is divided randomly into two parts- training and test data. The test data is 20% the size of the original dataset. Each entry consists of 10 different features including ProductId, UserId, ProfileName, Score, Text and Summary. These features facilitate the process of classifying the text into *positive* or *negative.* However, not all the fields are required to perform the classification. Thus, we use the concept of Dimensionality Reduction to get rid of the variables not required for our study.

Following this, the textual data is cleaned using tools from the Natural Language Toolkit available in python. This includes converting all uppercase characters to lowercase, removal of duplicates, removal of unwanted characters such as hashtag, emoticons, etc.

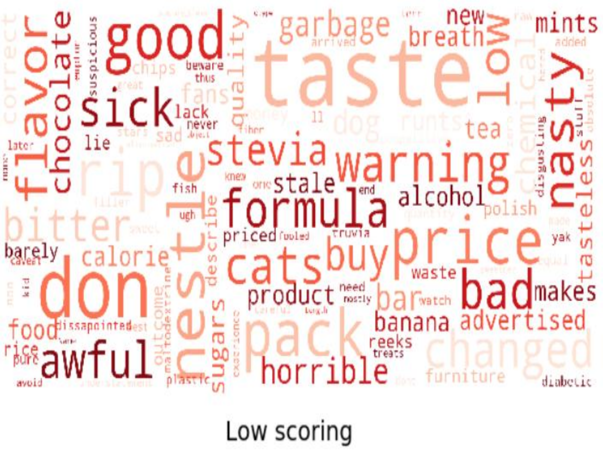

Figure 3. Words present in the negative reviews

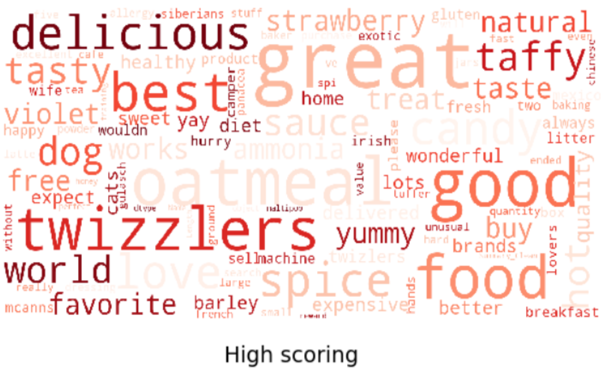

Figure 4. Words present in the positive reviews

In order to visualise the data, we propose making use of Word Clouds. A word cloud is a pictorial representation of the words present in the input text. They can used to identify the study the words present in the dataset. We make use of the word clouds to study the positive and negative scoring words as depicted in Figure 3 & 4.

Before training the model for determining the sentiment, we assigned weights to individual words by creating vectors. These weights depend on how important the word is to the entire textual data as a whole. This9 is done using the concept of *Term Frequency- Inverse Document Frequency (TF-IDF).* Both the frequencies are computed as shown below.

$$TF(t) = (Number\ of\ times\ term\ t\ appears\ in\ a\ document) /(Total\ number\ of\ terms\ in\ the\ document$$

$$IDF(t) = log\_e(Total\ number\ of\ documents /Number\ of\ documents\ with\ term\ t\ in\ it)$$

2) *Training the Classification models*

A variety of different machine learning techniques are being used to classify the training data: Naïve Bayes Classifier, Logistic Regression, Support Vector Machine and Decision trees.

Naïve Bayes Classifier works on the foundation of Bayes' Theorem and uses probability of the classification to compute the final outcome. It works on the assumption that each feature is independent and contributes equally to the outcome. For our study, we use the Multinomial and Bernoulli NB Classifier from the *Scikit-learn* package in python. The multinomial NB is used to model the count of a feature whereas the Bernoulli NB is only used to test whether the feature is present or not. The formula used to compute probability using NB classifier is Logistic Regression is a statistical technique adopted into machine learning. It performs a predictive analysis and describes the data. The analysis is done by representing one dependent and one or more independent variables graphically. We have used the Logistic Regression algorithm under Scikit-learn package.

Support Vector Machine is a simple machine learning algorithm that involves finding a separating hyperplane for the two different classes. Many different hyperplanes are possible but the algorithm aims to find the one with the maximum margins. Support Vectors are the datapoints that lie close to the hyperplane and determine its final position and orientation. In this research, we make use of the svm algorithm under the Scikit-learn package. Decision Trees are widely used classification algorithms. The root of the tree is at the top and it works its way down to the bottom, classifying the features at every level. These trees are prone to overfitting and thus need to be stopped at the required level or pruned later.

3) *Testing the data*

For testing our trained classification models, we make use of the 20% data in the test set along with a collection of textual data converted from speech. To compute the metrics of testings, we use the *accuracy_score* function of the Scikit-learn tool. It is a simple function that can be used to test the accuracy of the multilabel classifications. It computes the accuracy classification score based on the outcome of the test data. The accuracy can be manually computed using the formula

$$ACCURACY = \frac{TP + TN}{TOTAL}$$

TP- True Positives, TN- True Negatives

Following this, we create a function that can used to test any other textual data. The input to this function is the output from the previously converted text from speech. This function allows the user to either type our or plug in any textual data and run the classifier of their choice. The output of the function is the classification- *positive* or *negative* along with the probability percentage of the polarity.

IV. EXPERIMENTAL RESULTS

The training models produced different results with the test inputs. Highest accuracy of 96.05% was achieved using the SVM model. The results produced by the Logistic Regression were also high in accuracy. The NB classifier resulted in low accuracy and precision as compared to the other models. Table1 depicts the various classification metrics of the models.

Table1: Classification Metrics

Classifier	Accuracy	Precision	Recall
Multinomial NB	0.92	0.93	0.92
Bernoulli NB	0.88	0.89	0.89
Support Vector Machine	0.96	0.96	0.96
Logistic Regression	0.95	0.96	0.96
Decision Trees	0.94	0.95	0.95

To make the understanding and visualization of the classification metrics easier, we plot a Receiver Operating Characteristic (ROC) curve. It is a plot made to evaluate and compare different predictive models. The graphical representation is between True Positive Rate (TPR) and False Positive Rate (FPR). These are computed as

$$TPR = \frac{TP}{TP+FN} \qquad FPR = \frac{FP}{FP+TN}$$

Here, TP refers to the true positive and FP refers to the false positives.

The area under the ROC curve (AUC) is a measure which helps in determining the efficiency of the model as a whole. AUC closer to 1 means that the separation of classes is good and clear, whereas an AUC value closer to 0 represents a model with poor separation. Figure 5 depicts the ROC curve for the different classifiers.

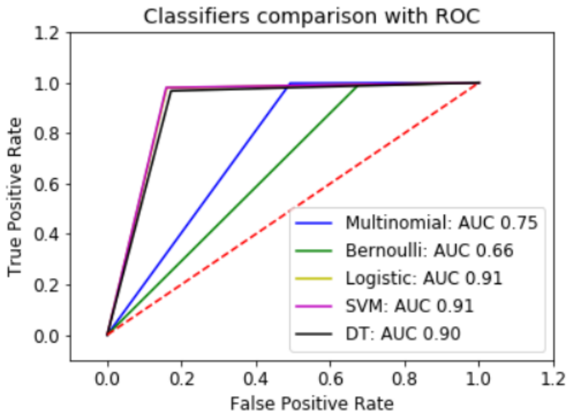

Figure 5. ROC Curve

As observed from the curve, SVM and Logistic Regression have the highest AUC values. Lowest separation is shown by Bernoulli NB Classifier. Overall, SVM and Logistic Regression are the most efficient and the best performing classifiers.

V. CONCLUSION

We have developed a system that takes speech as input and classifies the input as positive or negative by converting the speech to text and performing a sentiment analysis on the converted text. Various classifiers are used to determine the polarity of the text. We have performed a comparative study on the different classifiers used and recorded the results of the same. Support Vector Machine and Logistic Regression have proven to be the best classification techniques with the data we have used. Future scope of the project involves increasing the accuracy of the classifiers by creating an ensemble classifier which is basically a combination of all existing classifiers.

REFERENCES

1. Shweta Rana and Archana Singh, “Comparative Analysis of Sentiment Orientation Using SVM and Naïve Bayes Techniques” – in the 2016 2nd International Conference on Next Generation Computing Technologies, IEEE.
2. Santosh Kumar K L, Jayanti Desai and Jharna Majumdar, “Opinion Mining and Sentiment Analysis on Online Customer Review” – in the International Conference on Computational Intelligence and Computing Research, IEEE 2016.

3. Zeenia Singla, Sukhchandan Randhawa and Sushma Jain, “Statistical and Sentiment Analysis of Consumer Product Reviews” – in the 8th International Conference on Computing Communication and Networking Technologies, IEEE 2017
4. Karthikayini T and N.K. Srinath, “Comparative polarity analysis on Amazon product reviews using existing machine learning algorithms” – in the 2nd International Conference on Computational Systems and Informational Technology for Sustainable Solutions 2017, IEEE
5. Ardianda Aryo Prakoso, By Winantesa Yananta, Arif Fitra Setyawan and Muljono, “A Lexicon-based Sentiment Analysis for Amazon Web Review” – in the International Seminar on Application for Technology of Information and Communication, IEEE 2018
6. Neethu M S and Rajasree R, “Sentiment Analysis in Twitter using Machine Learning Techniques” – in the 4th International Conference on Computing Communication and Networking Technologies, IEEE 2014
7. Monisha Kanakaraj and Ram Mohana Reddy Guddeti, “Performance Analysis of Ensemble Methods on Twitter Sentiment Analysis using NLP Techniques”- in the Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)

AUTHOR PROFILE

Tavishi Priyam, an aspiring data scientist, currently pursuing her B. Tech in Computer Science and Engineering from SRM Institute of Science and Technology, Kattankulathur. Her areas of interest for research include Machine Learning, Natural Language Processing, Data Science and Big Data Analytics.

AMJ Muthukumaran pursuing his PHD in Computer Science and Engineering and his M. Tech in Computer Science and Engineering in 2007. Currently, he is working as Assistant Professor in Department of Computer Science & Engineering at SRM Institute of Science and Technology, Chennai. His research interests are in software engineering, with an emphasis on improving software reliability and dependability. Published many research papers in Indian and International journals.

Himanshu Vinayak, pursuing his B. Tech in Computer Science and Engineering from SRM Institute of Science and Technology, Chennai. Currently is working in Mobicip Technologies Pvt. Ltd., Banglore as a mobile developer. His research interests include software engineering and mobile applications