A Major Project Report

On

EMOTION DETECTION FROM TEXT

Submitted to Jawaharlal Nehru Technological University for the partial fulfillment of the requirement for the Award of the Degree in

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

Submitted by

V. RAMYA (18271A0527)
K. KALYANI (18271A0514)
K. HEMA (18271A0513)
V. AKHIL (18271A0503)

Under the Esteemed guidance of

Mr. M. RAVINDAR
Assoc. Professor (CSE Dept.)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

JYOTHISHMATHI INSTITUTE OF TECHNOLOGY & SCIENCE
(Approved by AICTE, New Delhi, Affiliated to JNTUH, Hyderabad)
NUSTULAPUR, KARIMNAGAR- 505481, TELANGANA, INDIA

2021-2022

CERTIFICATE

This is to certify that the Project Report entitled **"Emotion Detection From Text"** is being submitted by **V. Ramya (18271A0527), K. Kalyani (18271A0514), K. Hema (18271A0513), V. Akhil (18271A0503)** in partial fulfillment of the requirements for the award of the Degree of **Bachelor of Technology** in **Computer Science & Engineering** to the **Jyothishmathi Institute of Technology & Science**, Karimnagar, during academic year 2021-22, is a bonafide work carried out by them under my guidance and supervision.

The results presented in this Project Work have been verified and are found to be satisfactory. The results embodied in this Project Work have not been submitted to any other University for the award of any other degree or diploma.

Project Guide

M.RAVINDAR
Assoc. Professor
Dept. of CSE

Head of the Department

Dr. R . JEGADEESAN
Professor & HOD
Dept. of CSE

External Examiner

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our advisor, **Mr. M. RAVINDAR**, **Assoc. Professor, CSE Dept.,** whose knowledge and guidance has motivated us to achieve goals we never thought possible. The time we have spent working under his supervision has truly been a pleasure.

The experience from this kind of work is great and will be useful to us in future. We thank **Dr. R. Jegadeesan, Professor & HOD** CSE Dept**.** for his effort, kind cooperation, guidance and encouraging us to do this work and also for providing the facilities to carry out this work.

It is a great pleasure to convey our thanks to our principal **Dr. K. S. Rao,** Principal, Jyothishmathi Institute of Technology & Science and the College Management for permitting us to undertake this project and providing excellent facilities to carry out our project work.

We thank all the **Faculty** members of the Department of Computer Science & Engineering for sharing their valuable knowledge with us. We extend out thanks to the **Technical Staff** of the department for their valuable suggestions to technical problems.

Finally Special thanks to our parents for their support and encouragement throughout our life and this course. Thanks to all our friends and well wishers for their constant support.

DECLARATION

We hereby declare that the work which is being presented in this dissertation entitled, "**Emotion Detection From Text**", submitted towards the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering**, **Jyothishmathi Institute of Technology & Science**, Karimnagar is an authentic record of our own work carried out under the supervision of **Mr. M. Ravindar, Associate Professor, Department of CSE,** Jyothishmathi Institute of Technology and Science, Karimnagar.

To the best of our knowledge and belief, this project bears no resemblance with any report submitted to JNTUH or any other University for the award of any degree or diploma.

V.RAMYA (18271A0527)
K.KALYANI (18271A0514)
K.HEMA (18271A0513)
V.AKHIL (18271A0503)

Date:
Place: Karimnagar

Table of Contents

ABSTRACT

Human emotions can be expressed in many ways through facial expressions, speech, actions and in textual form. Emotions play an important role in determining the human's state and influence their lives in decision making, behavior and interaction. With the evolution of many advances in the technology, researchers are able to detect the emotion of humans through facial expressions, speech and text. Many advanced and efficient research work has been done in the field of emotion recognition through facial expressions and audio, but still there are not great works on emotion detection from text. Detecting emotions from text has wide applications in stock markets, business, decision-making, analyzing the view of people on any topic through social media conversations, tweets, blogs and articles.

Our work is focused on detection of emotions from text. We propose different approaches for detecting emotions through text. We implemented different approaches liked lexicon based approach, supervised machine learning approach with naïve bayes algorithm and unsupervised machine learning approach with semantic similarities. After the consideration of the cons of all the approaches and the accuracies derived from our work, it is evident that unsupervised machine learning approach with semantic similarities is the best approach with an accuracy of 79.9%.

Key Words: Emotion Detection, Navie Bayes, Semantic Similarity

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 INTRODUCTION

NLP or Natural Language Processing is a field of Artificial Intelligence that enables the machines to peruse, comprehend and get information from the dialects used by humans [1]. It is a field that centers around the association between information science and human language, and is scaling to heaps of ventures. Natural language preparing (NLP) is a subfield of phonetics, software engineering, data building, and artificial intelligences about the connections among PCs and human (normal) dialects, specifically how to program PCs to process and investigate a lot of regular language information.

NLP is trending and shows immense enhancements in the access to information and the expansion in computational force, which are permitting professionals to accomplish significant outcomes in zones like social insurance, media, fund and HR, among others.

In basic terms, NLP speaks to the programmed treatment of human language like audio or textual information, and in spite of the fact that the idea itself is captivating, the genuine incentive behind this innovation originates from the applications. NLP can assist you with loads of errands and the fields of use simply appear to increment on a day by day basis. NLP empowers the acknowledgment and detection of diseases dependent on electronic health records and patient's own discourse. This ability is being investigated in health conditions that go from cardiovascular illnesses to melancholy and even schizophrenia. For instance, Amazon Comprehend Medical is an assistance that utilizes NLP to identify disease conditions, drugs and treatment results from patient notes, clinical preliminary reports and other electronic health records.

Organizations can determine what customers are saying about a service or product by identifying and extracting information in sources like social media. This sentiment analysis can provide a lot of information about customers choices and their decision drivers. Associations can figure out what clients are stating about a help or item by distinguishing and removing data in sources like internet based life. This notion examination can give a great deal of data about clients decisions and their choice drivers.

Emotion detection

The process of recognizing the state of feeling of humans is called Emotion detection. Guessing the emotion of human through visually looking at them is easier. But recognizing the emotion by the usage of technology or some software is a typical task. Usage of technology to recognize emotions is a prominent field in the area of research. Most of the developed technologies regarding emotion detection are for facial expression recognition, speech and textual information. Though it is a challenging task, natural language processing combined with machine learning show great results in detecting emotions. By the use of automated emotion recognition, businesses can take better decisions by analyzing the customer emotions and feelings towards their products and services expressed through textual processing and automated video analytics. It reduces the expenditure and makes life better.

Emotion Detection from text

Emotion detection from text is a trending topic in the research area. It is related to sentiment analysis. In sentiment analysis only the positive, negative and neutral feelings from the text are detected where as in Emotion detection the type of the feeling towards the topic is also detected i.e., emotions of the text are also recognized. The various emotions that can be expressed include happiness, sadness, surprise and anger.

Emotion Detection from speech

Emotion recognition from speech is a field to recognize various emotions incorporated in the regular conversations of humans. It is generally termed as SER, Speech Emotion Recognition. Through tone of the voice and intensity we convey many emotions. The pitch and tone of the voice is how the emotion of the human is understood by the other living beings, humans and animals. Recognition of emotions from speech is tougher compared to text as the annotating data is a challenging task.

Emotion Detection from Images

Emotions can be recognized from facial expressions of humans which serves in AI. It is done by analyzing the real time expressions on the face. A face can represent several number of emotions like sadness, happiness, disgust, anger, fear, shock, surprise etc.. For every emotion the actions and features keep varying. For the

recognizition and detection of emotions from the facial expressions, we consider the key elements. Some features are calculated for the three dimensional face and these coefficients are used as key features.

Discrete Emotion Categories

Humans have different kinds of emotions. Of those, few emotions are considered as the basic emotions of humans. Emotions are discrete and can be divided into different types because they vary from human to human depending on the emotional state of the human and the biological processes.

Emotions are not fixed and they have evolved from time to time with the evolving nature of human based on their needs, characters and situations. Emotions play prominent role in the survival and they affect the cognition and behavior of the individual.

Theorists have researched to detect the basic emotions by broadly categorizing the secondary emotions. Paul Ekman and his colleagues in 1992, researched on these emotion categories and found that anger, fear, happiness, sadness, surprise and disgust are the six basic emotions of every human. Ekman explained these as the basic emotions and various emotions are attached to each of these categories and they vary in the degree of intensity of the emotion. Each emotion is a discrete category, not just a individual state.

A Psychologist Robert Plutchik broadly categorised the emotions into eight categories. They are joy, trust, fear, surprise, anger, anticipation, sadness and disgust. He also created a wheel of emotions to clearly demonstrate the connectivity and relationship among the emotions. The basic emotions are just categorised into eight emotions but the wheel of emotions shows different degrees of the emotions, creating a wide spectrum of emotions.

Parrott in 2001 worked on the emotions and classified them into primary and secondary emotions. He constructed a tree structure of all emotions. All the primary and secondary emotions are structured in the form of a tree, with the first layer representing the 6 basic primary emotions joy, love, sadness, fear, surprise and anger. These basic emotions are further classified into secondary emotions. Each emotion category has several other emotions varying in their degree and intensity. And these secondary emotions are classified further into tertiary emotions.

1.2 EXISTING SYSTEM:

There are many approaches for emotion detection from text. Emotions can be expressed and detected easily from facial expressions and speech, but it is a challenging task to analyse the textual information for identification of emotion.
It has gained attention in the recent days due to its various applications.

One of the basic approaches is keyword spotting approach which includes finding the keywords from the corpus of the dataset and finding the frequency of occurrence. This helps to find the emotion key words from the textual dataset based on the pre-defined words for each emotion. This approach uses the synonyms and antonyms from the database such as 5 wordnet, SentiWordnet etc.. In the lexical database there are words for every emotion category such as happy, sadness, anger, fear, disgust etc.. The bootstrapping approach uses a collection of words with their synonyms and antonyms to predict the semantic orientation of the words, adjectives. In WordNet, the organization of words is done into clusters based on the polarity and the words are oriented accordingly. For a word to assign its orientation, the antonym and synonym sets are searched in the WordNet. The whole process starts with the tokenization and then followed by extraction of emotion related keywords. After checking the polarity, the negation check done and then the emotion category is detected.

Disadvantages:

1. **Ambiguity of the words is not resolved**

The meaning of the word may differ according to the situation. Words can have different meaning with respect to the context used. Some words with various synonyms could have discrete emotions in some cases. Resolving the ambiguity of the word is not possible through keyword extracting approach.

2. **Emotions cannot be detected for sentences without keywords**

As this approach wholly depends on the presence and extraction of keywords, the detection of emotions from sentences without keywords is not possible. Emotion of such sentences are considered neutral.

3. **Ignorance of linguistic information**

Emotion of the sentence can be influenced by the semantic structure of the sentence. Depending on the perspective of the person, the emotion varies. For example, "I kicked him" and "He kicked me" represent different emotions based on the person's perspective. With the ignorance of such linguistic rules and information, the emotion is not predicted accurately.

Another approach is lexical based approach, keyword based approach is extended into this method by using probabilistic affinity for each emotion category. It is based on arbitrary words, not just on the keyword representing the emotion. This works based on lexical affinity and also considers the semantic and syntactic structure of the sentence. For example, a word cried is considered as a negative word and the lexicon based approach may fix the probability as 70 percent to the word indicating negative effect as in cases "I cried for him yesterday" or "I cried over my weaknesses". But it may not be negative in every context such as "she cried out of shock" which means she is not sad.

1.3 PROBLEM STATEMENT:

Human emotions can be expressed in many ways through facial expressions, speech, actions and in textual form. Emotions play an important role in determining the human's state and influence their lives in decision making, behavior and interaction. With the evolution of many advances in the technology, researchers are able to detect the emotion of humans through facial expressions, speech and text. Many advanced and efficient research work has been done in the field of emotion recognition through facial expressions and audio, but still there are not great works on emotion detection from text. Detecting emotions from text has wide applications in stock markets, business, decision-making, analyzing the view of people on any topic through social media conversations, tweets, blogs and articles.

Our work is focused on detection of emotions from text. We propose different approaches for detecting emotions through text.

1.4 PROPOSED SYSTEM:

1.4.1 PROPOSED SYSTEM

We propose a unsupervised machine learning method for the detection of emotion. We implemented different approaches for detection of emotion and after analyzing the disadvantages and the accuracies of those approaches. Our work is carried out on ISAER dataset to detect the 7 basic emotions joy, surprise, disgust, anger, fear, shame and guilt.

We experimented with lexicon approach, and a supervised and unsupervised machine learning model to detect emotions. Naivebayes classifier of textblob is used as a supervised machine learning model and used gensim to create a unsupervised model so that it does not require any pre-labeled training data. Gensim provides various methods to find the semantic similarity between the words thus considering all the semantic relations between the words. By the consideration of semantic relation between the words it gives a perfect model to predict the emotions with more accuracy.

Advantages:

- It is more feasible to implement as it does not require large sets of pre-labeled training data.
- It gives good accuracy as the semantic structure of the sentence is considered

1.4.2 OBJECTIVES

The objective of the work carried out by us is to determine an efficient approach for detection of emotions from text based on the accuracy of the approach and by considering all the advantages and limitations of the approaches. There are many existing solutions based on supervised models and other approaches. We work with supervised and unsupervised methods to determine which model is more efficient and accurate as the semantic meaning of the words differs according to the context.

2. LITERATURE SURVEY

Title: Text-Based Intelligent Learning Emotion System

Year: 2017

Authors: Mohammed Abdel Razek, Claude Frasson

Description:

The methodology proposed by them involves two steps- using 40% of the data to train the machine, extracting features according to the appraisal method, creation of emotion dominant meaning hierarchies, training the classifier with the preprocessed training data and predicting on the test data using the classifier. The architecture contains training stage and classification stage with the training happening at the server. Dominant meaning trees are prepared from the ISEAR dataset, that tree contains all the 7 emotions joy, fear, sadness, shame, guilt, anger and disgust. The classifier uses this information from the tree. In the proposed approach, the classifier does not use the pre-labeled examples for classification of text, which is tie taking process. It uses the data from the constructed dominant trees. The construction of the tree is the important step, the top dominant words representing an emotion should be identified to be included in the hierarchy. These words are detected based on the frequency of the words. For a new sentence to be classified, the emotion detection algorithm calculates the values using the dominant hierarchies and returns the index of the emotion with the highest value. They followed the Cohen's Kappa method to test the accuracy of the classifier. Their work proved that SVM gave better accuracies for classification of emotion.

Title: Emotion detection from text

Year: 2018

Authors: V VRamalingam, A Pandian, Abhijeeth Jaiswal, Nikhal Bhatia

Description:

This paper proposes a method based on machine learning for emotion detection using supervised machine learning algorithms like support vector machine. The emotion categories used are Anger, Disgust, Sadness, Joy, Fear. There are many approaches for recognizing emotions but every approach has it's own defects and advantages, so it is preferred to use hybrid methods as they give high accuracies. The work is not based

wholly on the machine learning based approach, but it is a hybrid method combining both key word based approach and learning based approach. Key-word based approach includes detection of emotion based on synonyms and antonyms from the wordNet which is created manually. Their work is done by text processing followed by extraction of keywords from the dataset, checking the emotion of the keywords with the wordNet created (files of the emotion words). Each emotion has a separate file with all the words related to that particular emotion. In the learning based approach, they extracted tweets using Graph API and the applied SVM algorithm on the training dataset. The whole work is done on the text obtained from converting a speech into text using Speech Recognition Package, then processing the text by tokenization and using the word Net to find the emotion related. The output is shown graphically with x-axis showing the five emotions Anger, Disgust, Sadness, Fear, Joy and the y-axis shows the amount of that respective emotion detected in the sentence.
The future enhancement of the work is to increase the efficiency of the proposed system and include more number of emotions for recognition.

Title: Unsupervised Emotion Detection from text using Semantic and Syntactic Relations

Year: 2017

Authors: Ameeta Agrawal, Aijun An

Description:

This research paper focuses on the detection of emotions through unsupervised methods. 11 There are many supervised methods, but they require large sets of annotated data which is hard to find. So, they proposed an unsupervised method which uses the semantic relationship between the words to detect the context of the words used, thus finding the emotion of the text. It does not require a word net to detect emotion, so it is more flexible and not dependent on a few set of words. The methodology starts with preprocessing of text, extracting relevant affect bearing words, calculating the semantic relativity between the words. The semantic scores of the words are aggregated thus analyzing at the sentence level to detect the emotion of the sentence. They employed PMI for calculation of semantic relations between the words which is a simple yet powerful method than the previous version LSA. Further experiments proved that stemming improved the accurary. This may be because the roots of words which belong to a specific emotion category get grouped together and

this latent clustering enables more accurate frequency counts. They had applied the model extensively on various datasets ISEAR Four emotions classification, ISEAR six emotions classification, Alm six emotions and Amag blog dataset, thus proving the effectiveness of the proposed model. The drawback of this model is the semantic relations are depending on the corpus of the text and we can overcome this by enhancing the derivation of semantic similarity from multiple sources.

Title: Comparative Analyses of BERT, RoBERTa, DistilBERT, and XLNet for Text-based Emotion Recognition

Year: 2020

Author: Francisca Adoma Acheampong , Henry Nunoo-Mensah,Wenyu Chen.

Description: This paper aims to find the efficiency of the BERT,RoBERTa,DistilBERT, and XLNet by using ISEAR dataset.The experimentsand results obtained from the four models are comparativelydiscussed concerning their accuracy, precision, and recall on the ISEAR dataset.The accuracy of four models are 0.7431,0.7299,0.7009, and 0.6693 for RoBERTa,XLNet, BERT, and DistilBERT. The F1 score of BERT RoBERTa, DistilBERT , XLNet are 0.6,0.65,0.52.0.63. The model RoBERTa has the highest accuracy compared to other model.The precision, recall, and F1-scores further proved the efficacy of RoBERTa over the other candidate models in recognizing the emotion on the ISEARdata

3. REQUIREMENTS & DOMAIN INFORMATION

3.1 REQUIREMENT SPECIFICATIONS:

Requirement Specifications describe the articraft of Software Requirements and HardwareRequirements used in this project.

3.1.1 Software Requirements:

Operating System: Windows10

Tool: Anaconda python with Jupyter Notebook

3.1.2 Hardware Requirements:

Processor: Pentium IV

Hard disk: minimum 80 GB

RAM: minimum 2 GB

3.2 DOMAIN INFORMATION:

Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python is simple and easy to learn. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Since there is no compilation step, the edit test debug cycle is incredibly fast. Debugging Python programs is easy and a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. The Proposed System works on python and above.

Jupyter notebook:

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. The jupyter notebook app is a server-client application that allows editing and running notebook documents via a web browser. The Jupyter Notebook app can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet. In addition to displaying/editing/running notebook documents, the Jupyter Notebook App has a "Dashboard" (Notebook Dashboard), a "control panel" showing local files and allowing to open notebook document and shutting down their kernels.

NLTK

NLTK is the abbreviation for Natural Language Toolkit. It is the most powerful library in NLP used for text processing NLTK stands for Natural Language Toolkit. It is a very popular library for developing python programs that help us to work with natural language. This toolkit is a powerful library in NLP. It contains various packages which provide various functionalities that helps us to make machines understand the language used by humans to communicate and respond to it with the natural language. This serves as a good interface for many corpora and lexical resources like wordnet and for many libraries used for text processing tasks like tokenization, stemming, parts of speech tagging and classification.

Nltk.corpus

NLTK.corpus is a package that contains number of corpusreader classes. These are helpful in using various sets of corpora by accessing them. Every corpus may have different formats and there are classes designed to handle such corpora. Each class is designed to handle a particular type of corpus. A set of corpus reader instances are automatically created by the package to help in easy access of corpora in the nltkpackage. The package has many types of files including xml and many other formats. Accessing method is same for all types of corpora.

Nltk.corpus.Stopwords

There are many words used in English language which help to frame the sentence, but they do not contribute much to the meaning of the sentence. Moreover these words consume the time of processing and learning. So they are removed from the dataset in the text processing stage itself. They can be ignored without disturbing the meaning of the text. The nltk.corpus package has many datasets, it contain the stopwords file which is useful for removing those from the datset.

RE module

RE module of python is used for regular expressions. A regular expression describes a pattern that can be used to find words, strings and set of strings from the text. It can be done easily by framing a sequence of special characters. Python usage with regular expressions is easier to find words with specified syntax. Many characters are used to represent a pattern and they have special meanings in the regular expression. If those characters are to be searched for, we can use raw strings to avoid ambiguity such as r'expression'.

Numpy

Numpy stands for Numerical Python. Numpy is a library that deals with scientific computing, n-dimensional arrays.. It is a fundamental library in python used for computing. It can be used to define arbitrary data types. It is a great container of generic data. Due to the special features of this package it can be easily integrated and used along with various databases.Numpy is used along with Pandas for easier data manipulation and computations. Numpy is used for operations of n-dimensional arrays. These arrays contain n number of rows and columns. These arrays can contain values of same data type and the sequences start with zero. It contains many ways of accessing and processing of arrays Numpy features help developers to perform various operations, complex and scientific computing at much ease. Some of the operations it can perform are the following-

- Mathematical computations with multi dimensional arrays.
- Operations related to Fourier transformations, linear algebra.
- Random number generation.

Pandas

Pandas is a popular open-source python library used for data manipulation. It is built on the top of Numpy library.Pandas library provides various data structures and various ways of accessing those data structures. It offers various operations to be performed for manipulation of numerical data and time series data. It is highly used for importing, analyzing and manipulating the data. Pandas is very fast, efficient and productive for users. It is popular due to it's high performance. Pandas is used with some other libraries in conjunction. It is mostly used along with numpy for performing operations on dataframes. Plotting can be done by the use of data generated by pandas. This library is used for loading different data with different formats such as heterogeneously typed columns, tabular data and time series data.

- Data from different sources of different formats can be uploaded.
- Handling of missing data in large files becomes much easier.
- It is fast and efficient in manipulation of data
- The size of the data can be mutated by insertion and deletion of rows and columns

TextBlob

TextBlob is a newPython library in Natural Language Processing .It is a powerful toolkit that stands on the shoulders of giants like NLTK and provides all the functionalities for text mining, text analysis and text processing. It is used for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks. This is built on the top of NLTK and the interface is much easier to use. It works faster and is efficient than nltk(Natural Language tool kit). It is efficient in NLU (Natural Language Understanding). It is the main concept behind Human Computer Interaction(HCI). It is most prominently used in chatbots, AI robots. It performs NLP tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.It contains very simple functions to tokenize the dataset at a single step, to extract different words belonging to a particular parts of speech etc… One of the unique functionalities of textblob is correction of spellings. With the use of a simple function, we can easily correct the spellings of all words in a document.

Sklearn

Scikit-learn is a free machine learning library for Python. It has powerful tools for data mining and data analytics. This library is used for modeling the data. Unlike Numpy

and Pandas it doesn’t focus on loading , manipulating and operating on the data. It features various algorithms for usage. It provides various supervised and unsupervised algorithms. It provides functionalities for Regression, Classification, Clustering, Model 22 Selection and Preprocessing. It is built on the top of natural language processing libraries like nltk, pandas, Numpyand matplotlib. It provides various machine learning algorithms like support vector machine, random forests, and k-neighbours and also used to estimate the performance of classifiers for supervised algorithms. This library supports Python numerical and scientific libraries like NumPy and SciPy. It is better to import the algorithms from the library rather than implementing them from scratch as it doesn’t reach the efficiencies and the accuracies desired. Not only the algorithms, this library provides various models for clustering, supervised models, models for data analysis, cross-validation, dimensionality reduction, Ensemble methods, Parameter Tuning, Feature Extraction, Feature Selection, Manifold Learning.

Gensim

Gensim is aopen source Python library for natural language processing, unsupervised topic modelling, document indexing and similarity retrieval with large corpora.It can be implemented in Python and Cython. This library is used for topic modeling using modern statistical machine learning.Gensim isused to handle large text collections using data streaming and incremental online algorithms. It is a good package for text processing, building topic models and to work with models such as Word2Vec, FastText, which are word vector models. Gensimlet’s the users to handle large sized text files without even loading the entire file into the memory of the model..This feature of gensim makes it unique from many other machine learning software packages as they focus only on in-memory processing.

3. SYSTEM METHODOLOGY

4.1 ARCHITECTURE OF PROPOSED SYSTEM:

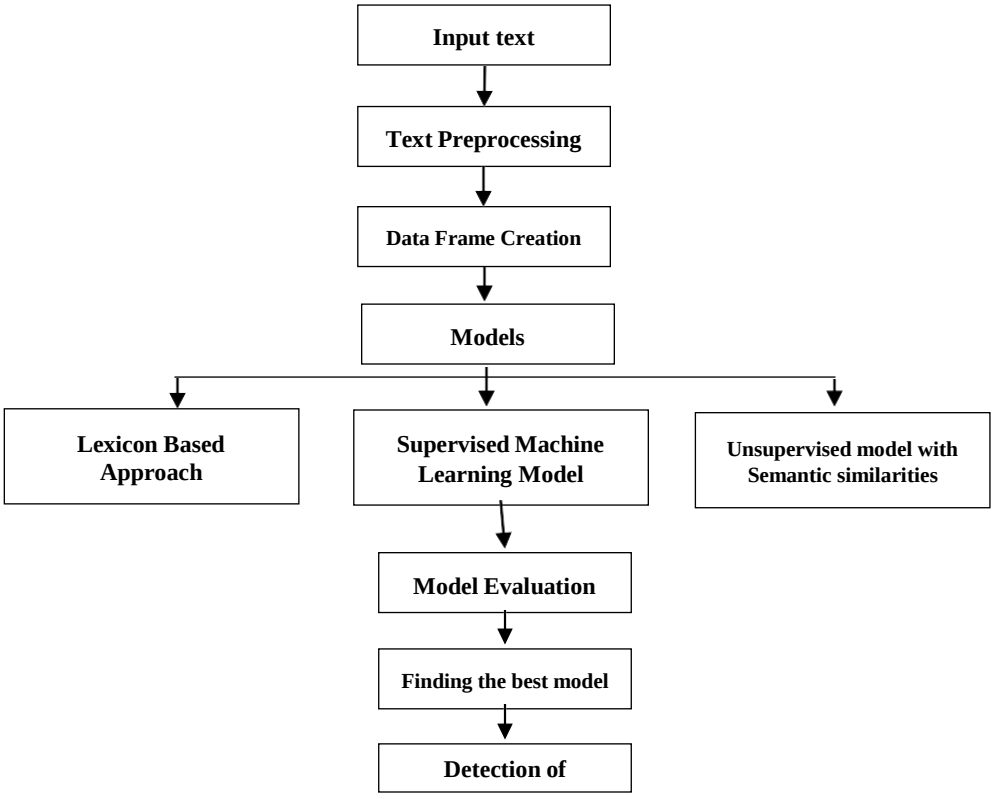

Fig 4.1.1Architecture of proposed system

The work is carried out by the collection of dataset as the first step. Data used for the experiments is ISEAR dataset. It contains sentences representing seven different kind of emotions joy, surprise, anger, sadness, shame, disgust and guilt. People express their emotions on social media through text. The emotion of people towards a particular topic can be understood by detecting the emotion of sentences used by people on social media platforms. The model built by us is tested using the International Survey on Emotion Antecedents and Reactions dataset (ISEAR). It is collected by collected by Klaus R. Scherer and Harald Wallbott. The sentences in the dataset are preprocessed through text processing techniques with the help of libraries pandas, numpy an nltk. After the preprocessing, the data frame is created from the preprocessed text for further usage. The dataset is undergone different approaches lexicon based by the use of the emotion keywords for different emotion categories. The dataset is divided into training and testing data along with the output labels for the

application of a supervised learning model and without output labels for the application of unsupervised learning model. The training data is used for the learning of the model and is tested on the testing data. The accuracies are calculated to find the best approach for detection of emotions.

Modules

1. Data collection and Text preprocessing
2. Detection of emotion through different approaches
3. Evaluation and choosing the best approach

1.Data collection and Text preprocessing

Data used for our work is International Survey on Emotion Antecedents and Reactions dataset (ISEAR) dataset. It is collected by collected by Klaus R. Scherer and Harald Wallbott. The dataset contains 7830 sentences with pre-labeled outputs. It features the sentences with 7 emotion categories joy, surprise, shame, guilt, sadness, disgust and anger. The data is collected and preprocessed using the functions of nltk library. The preprocessing tasks performed are tokenization, stemming, parts of speech tagging. Later it is converted to data frame on which models are applied to detect emotions.

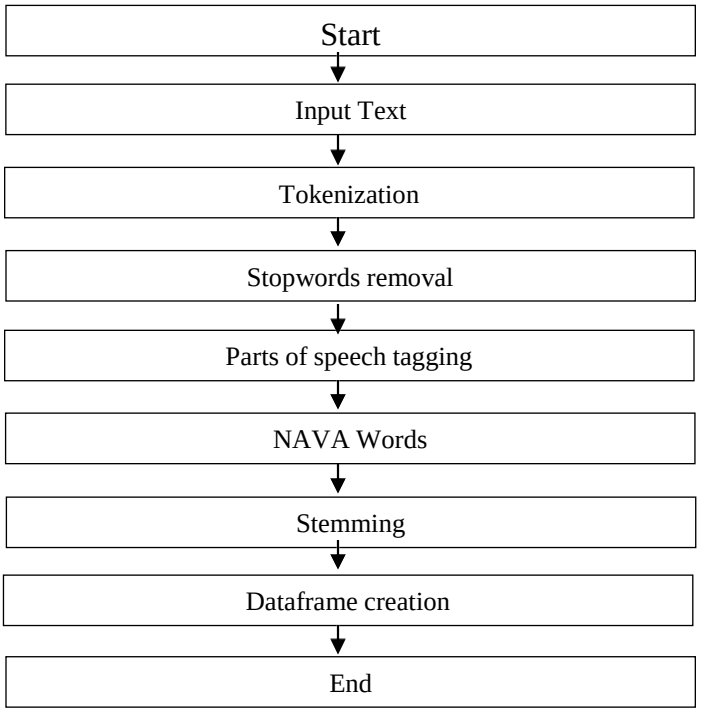

Fig 4.1.2 Pre Processing

2. Detection of emotions through different approaches

We applied three different approaches for building a system for emotion detection. In the lexicon based approach, emotion wordset is created from the manually created emotion keyword files. Check for the lexicons in the dataset by checking for the words in all sentences and creating a list of sentences and words, which act as sentence vectors and word vectors. Create a text vector by appending all the words of the word vectors. A word vector contains the words in the sentence with all the possible emotions of it as the values of the words. This text vector is used to classify the sentences and detect the emotions based on the lexicons. By applying sum and average functions to the classified words, the accuracy is calculated.

In the second approach, the dataset is divided into training and testing data and Naïve bayes classifier is applied on the training data. The trained classifier is tested on the test data. This is a supervised machine learning model. We calculate the accuracies for the model.

3. Evaluation and choosing the best approach

After the application of the algorithms, the model is tested with the test data and the emotions are detected for the new sentences. The accuracies of the models are calculated to choose the best approach. However, the unsupervised machine learning model to detect emotions using semantic similarities proved to give higher accuracies compared to the other two models. The other models are found to have disadvantages using them. In lexicon based approach the sentences without the keywords are not detected with appropriate emotions and is not suitable to all types of applications. The supervised machine learning model approach requires large sets of data with pre-labeled outputs which is hard to find as manual classification of large number of sentences is not possible

4.2 ALGORITHM

1. Lexicon approach

Creation of Emotion Word Set

After the creation of data frame and defining the emotion labels, an emotion word set is created. Based on the emotion labels, the word set is created. All the words of the

sentences in the dataset are checked with the representative words and the emotion detecting words are stored into the word set. This emotion word set is used for further classification of lexicons.

Checking for lexicons

Use of a dictionary of emotion representing keywords is one of the primary ways to deal with emotion identification and it includes figuring the emotion from the semantic direction of word or expressions that happen in a book. We require words speaking to all the feelings, these are made physically utilizing the feeling word set and the rundown of words speaking to a specific emotion are put away in their separate documents. By and large, this technique incorporates the portrayal of the considerable number of sentences as a lot of words. From this portrayal, the emotion representing words are recognized dependent on the feeling word set. Mathematical functions, for example, sum and average are applied efficient to make the last prediction with respect to the sentence's emotion. Be that as it may, it doesn't think about negation.

Classification based on lexicons

Check for the lexicons in the dataset by checking for the words in all sentences and creating a list of sentences and words, which act as sentence vectors and word vectors. Create a text vector by appending all the words of the word vectors. A word vector contains the words in the sentence with all the possible emotions of it as the values of the words. These words and the possible emotions are in the form of a dictionary. The combination of all the word vectors of a sentence is represented as a sentence vector. The text vector is the combination of all the sentence vectors. This text vector is used to classify the sentences and detect the emotions based on the lexicons. By applying sum and average functions to the classified words, the accuracies are calculated for every piece of text. The total accuracy is calculated by averaging all the accuracies of all the sentences. The disadvantage of this approach is it does not perform the negation check or intensification.

2. Hybrid approach of Machine Learning with NLP

Artificial Intelligence is involved in natural language processing and analytics of text along with machine learning algorithms. AI is used for comprehending textual data. These reports can be just about whatever contains content: internet based life remarks, online audits, study reactions, reviews, finance related, clinical, legitimate and administrative records. AI for NLP and content examination includes a lot of factual systems for distinguishing grammatical features, elements, estimation, and different parts of content. The strategies can be viewed as a model that is then checked by applying it to other content. It likewise could be a lot of algorithms that work across huge arrangements of information to extract meaning, which is known as unsupervised algorithm.

Text input is processed by the application of low-level textual functions at the initial stages. Unstructured data is converted to structured data by applying these functions at the starting stages, thus they are the base for information extraction. In the further steps, the mid-level functions are applied. These mid-level functions are used to extract the meaning from the textual document. Followed by the mid-level functions, the high level functions are applied to extract the emotion of the content at the sentence level and document level.

3. Supervised machine learning model

Splitting the dataset

There are two stages for the working of a supervised machine learning model- training stage and testing stage.

The algorithm learns from the training dataset which is already pre-labeled. It contains both the input and the output contents. The algorithm learns the rules and patterns from the pre labeled data and applies it on the test data. The aim behind the pre-labeled training set is to feed the algorithm with the “ground truth” data and learn from it.

The test data is used to know and test the algorithm how well it has learnt from the training data. The training data cannot be reused to test the algorithm as it is already aware of the output. Here is a simple flowchart that represents the process of training the algorithm and application of various functions on test and train data.

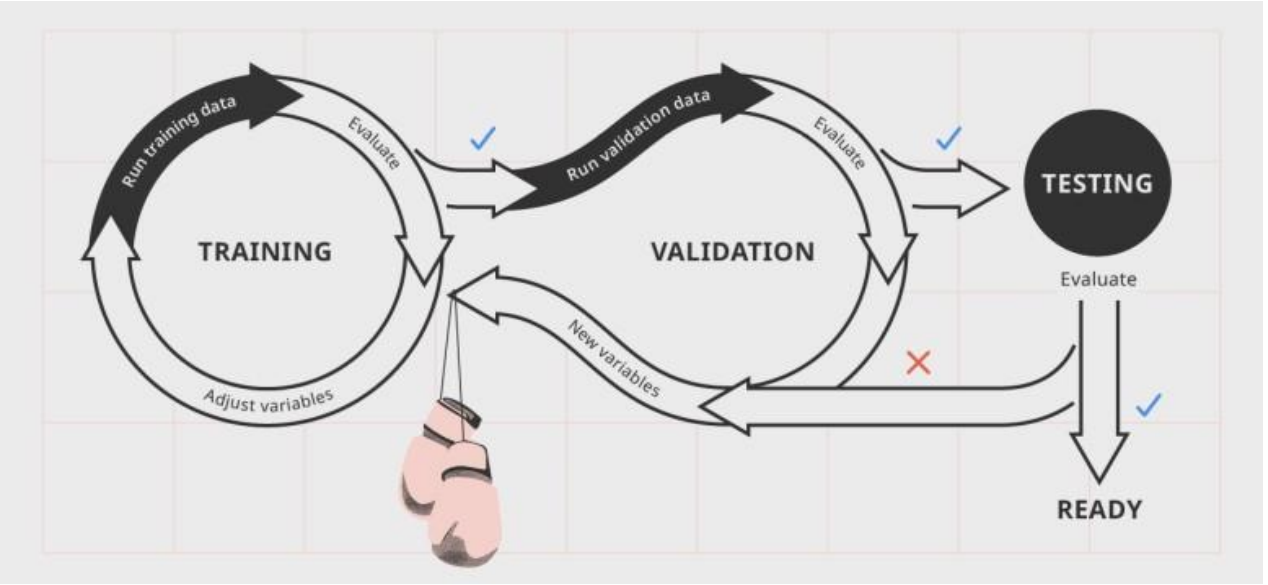

Fig 4.2.1 Splitting the dataset

We can use train_test_split method of sklearn.preprcoessing module to split the dataset or it can be done manually by the slicing of data frame. The dividing ratio has a huge impact on the accuracy of the model. Usually the training data size should be greater than the testing data size such that algorithm gets more data to learn from and various special cases too, thus achieving good accuracy.

We have divided the dataset in the ratio 90-1 for good accuracy of the model. Of all the7638 samples of our dataset 6830 samples are given to the training set and the remaining 1018samples i.e., 10% of the dataset is given to the testing set.

Application of Naïve Bayes Classifier

Naïve Bayes classifier is based on Bayes theorem. It works on simple probabilistic analysis in machine learning to determine the probability of the new data. The application of the algorithm is based on some assumptions. The feature of one class is independent of other feature.

The attributes of a class are independent of attributes of other class. Naïve bayes does the probability calculation of each attribute. This is a faster and efficient method. Conditional probability is the probability of a class over the probability of another attribute. We can obtain the probability of a instance of the dataset, the multiplication of the conditional probabilities for each attribute for a given class. By the calculation of probabilities of the instance, the class of the instance is detected. The class with the highest probability is considered as the output.

Algorithm:

P(A/B) = P(B/A) * P(A) / P(B)

1. A is called the **proposition** and is called the **evidence**

2. **P(A)** is called the prior probability of proposition and **P(B)** is called the prior probability of evidence

3. P(A/B) is called the **posterior**

4. P(B/A) is the **likelihood**

Naïve Bayes Classifier of textblob is used in text analytics, text parsing, detection of sentiment from text and emotion detection from text.

Fig 4.2.2 Working of Supervised machine learning model

4. Unsupervised Machine Learning model:

Unsupervised Machine Learning model takes the input data but it doesn't have any corresponding output labels. In this model there is no need to supervise the model. The goal of unsupervised learning is to model the underlying structure to learn more about the data and find unknown patterns in data by drawing inferences from the dataset. Here the machine is neither trained nor supervised by using training data but the algorithm acts on the information that is neither classified nor supervised without any guidance unlike the supervised algorithm does. They are left to their own to produce the interesting structure of the data. Since it doesn't have training data so the model categorizes based on the patterns or similarities. Unsupervised is of types i.e., Clustering and Association. Some of the algorithms that use unsupervised learning model are principal component analysis, singular value decomposition, k-means clustering etc.

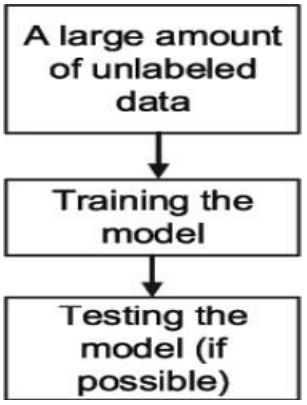

Fig 4.2.3 Unsupervised machine learning model

Here we have divided the dataset into unlabeled data i.e., training and testing data in 90:10 ratio to achieve good accuracy. And here we used semantic similarity model in which it finds the similarity between two sentences using cosine similarity by using gensim package and and Latent Semantic Indexing (LSI) model ,here genism which is a NLP package that does topic modelling for humans which is used for text processing and to work with word vector models and LSI model is used for topic modelling. Latent dirichlet Allocation and Latent Semantic Indexing (LSI) models are used for topic modelling.

Application of Gensim package to find Semantic similarity

1. Here the first step is to create a dictionary for the sentences list , the dictionary maps the tokens or words to its unique id's. So in order to create a dictionary for a list of sentences first we need to divide the sentences into words and we can give these words as input to convert into dictionary.

• To create the dictionary:

```
mydictionary = corpora.Dictionary(wordsofsentences)
```

• To get info. About our dictionary

```
print(mydictionary)
```

• To show the word to id mapping

```
print(mydictionary.token2id)
```

• Here we can save the dictionary file to our system as mycorpus.dict by using the command

```
mydictionary.save("mycorpus.dict")
```

• we can also assign our word to id mapping to a variable

gensimtoken=mydictionary.token2id

2. The second step is to create a bag of words model for the generated dictionary.we create the bag of words model by passing the tokenized set of words to the command mydictionary.doc2bow()

• Here the texts are converted to bag of words model and assigned to a variable as below Mycorpusmm=[mydictionary.doc2bow(word)for word in training_data]

• The corpus is saved to the system as .mm file(matrix market) as below

corpora.MmCorpus.serialize('mycorpus.mm',mycorpusmm)

• Here the .mm file can have random access

gensim_c = corpora.MmCorpus('mycorpus.mm')

3. The third step is to create a TFIDF matrix using gensim. The Term Frequency – Inverse Document Frequency(TFIDF) is a bag-of-words model .It down weights tokens (words) that appears frequently across the datasets. It is found by multiplying a local component i.e., term frequency (TF) with a global component i.e., inverse document frequency (IDF) and then normalizing the obtained result to the unit length. hence the words that come frequently across the document will be down weighted. Training the corpus with models.TfidfModel(). And then applying the corpus to square brackets of the trained tfidf model above.

• Calculating the TFIDF weights

Tfidf_model = models.TfidfMo

del(gensim_c) mycorpus_tfidf = tfidf_model[gensim_c]

4. The datasets can be loaded as below

dictionary = corpora.Dictionary.load('mycorpus.dict')

5. Here we create the topic models using LSI

• First we create lsi model by using existing module models.LsiModel() and passing TFIDF weights of the corpus and dictionary of texts and then we pass the TFIDF weights of the corpus to the square brackets of lsi model and thus we get the lsi model for our corpus

lsi_model = models.LsiModel(mycorpus_tfidf, id2word=dictionarymycorpus_lsi = lsi_model[mycorpus_tfidf]

6. Now, we create the index file to determine matrix similarities and save the index file and lsi file.

```
myindex=similarities.MatrixSimilarity(lsi_model[gensim_c])
lsi_model.save('mycorpus.lsi')
myindex.save('mycorpus.index')
```

7. The last step is calculating the similarity by using different similarity metrics such as calculating matrix similarity and thus finding the semantic similarities.

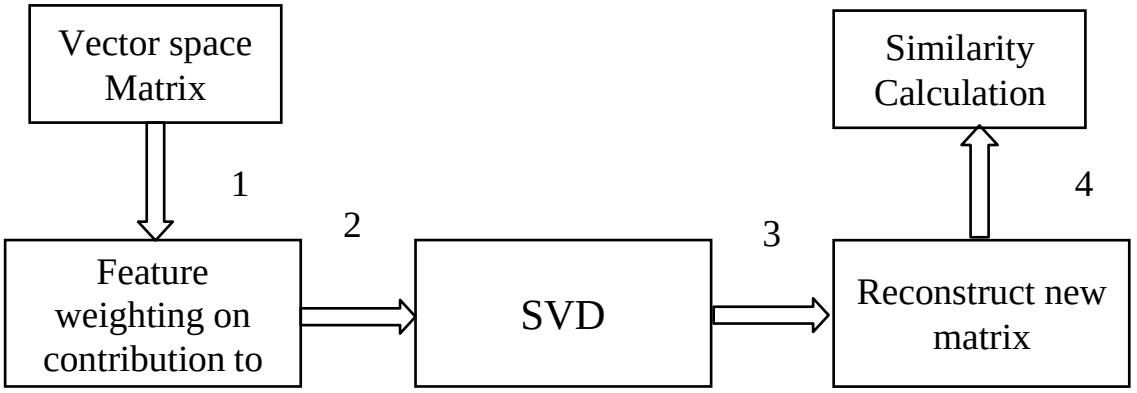

Fig 4.2.4 Working of LSI Model

4.3 SYSTEM DESIGN

4.3.1 UML Diagrams

The Unified Modeling Language (UML) is a universally useful visual displaying language that is utilized to determine, imagine, develop, and report the curios of a product framework. It catches choices and comprehension about frameworks that must

be developed. There are various types of diagrams in UML, each used for a different purpose.

4.3.1.1 Use Case diagram

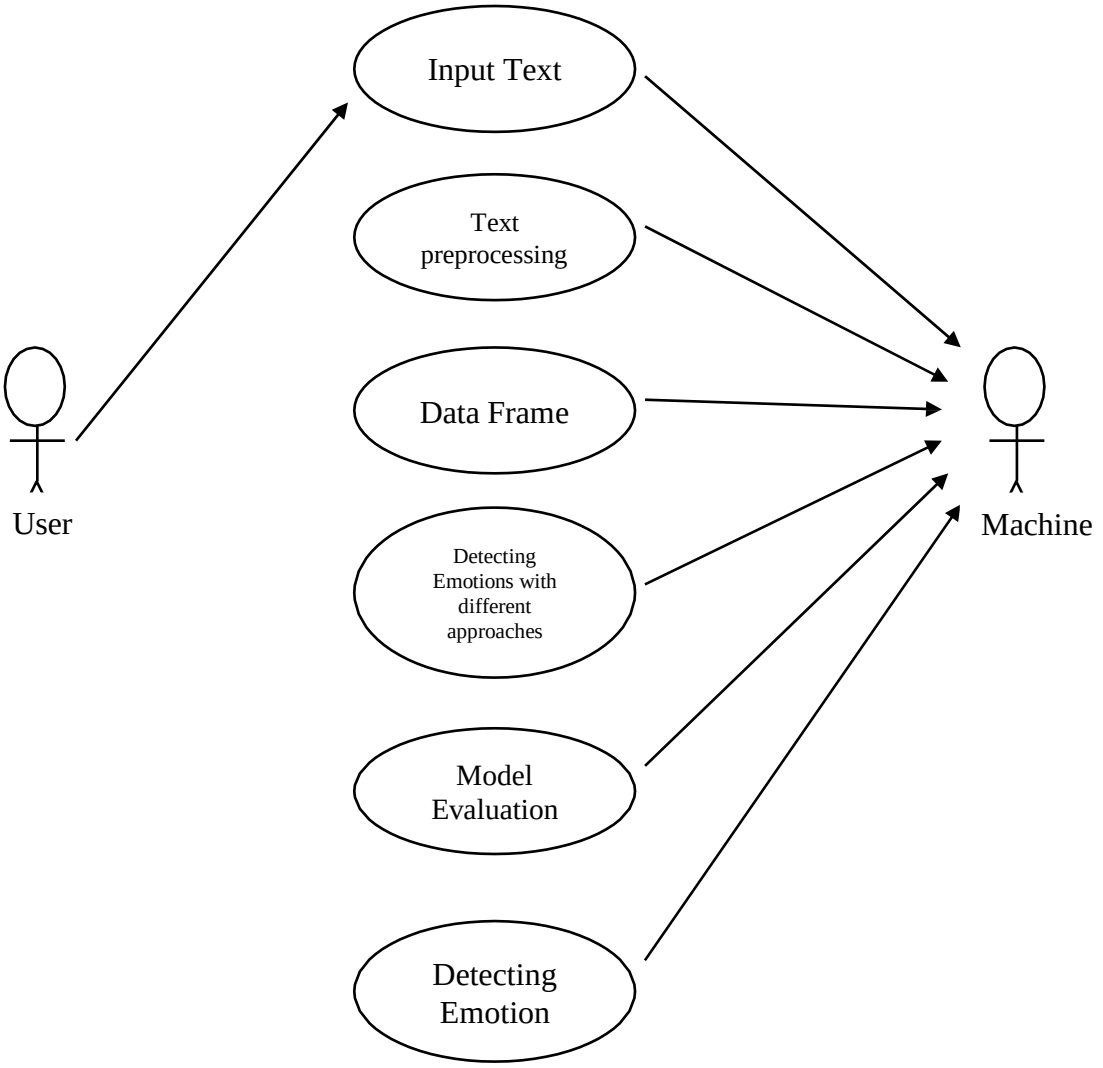

Fig 4.3.1.1 Use Case diagram

The dynamic behavior of the system is represented using Use Case diagram. It uses actors and use cases to describe the performance and functions of the system. A use case diagram is a dynamic or behavior diagram in UML. The activities, functions and services are represented by the use cases and the operators of the functions are the actors and they have the irrespective roles based on the actions performed and under the defined rules.

4.3.1.2 Activity Diagram

This diagram shows the process of execution of the system and the behavior of the system. It is a set of activities that represents the flow by which actions are taking place in the system. Each activity represents the action performed in every step.

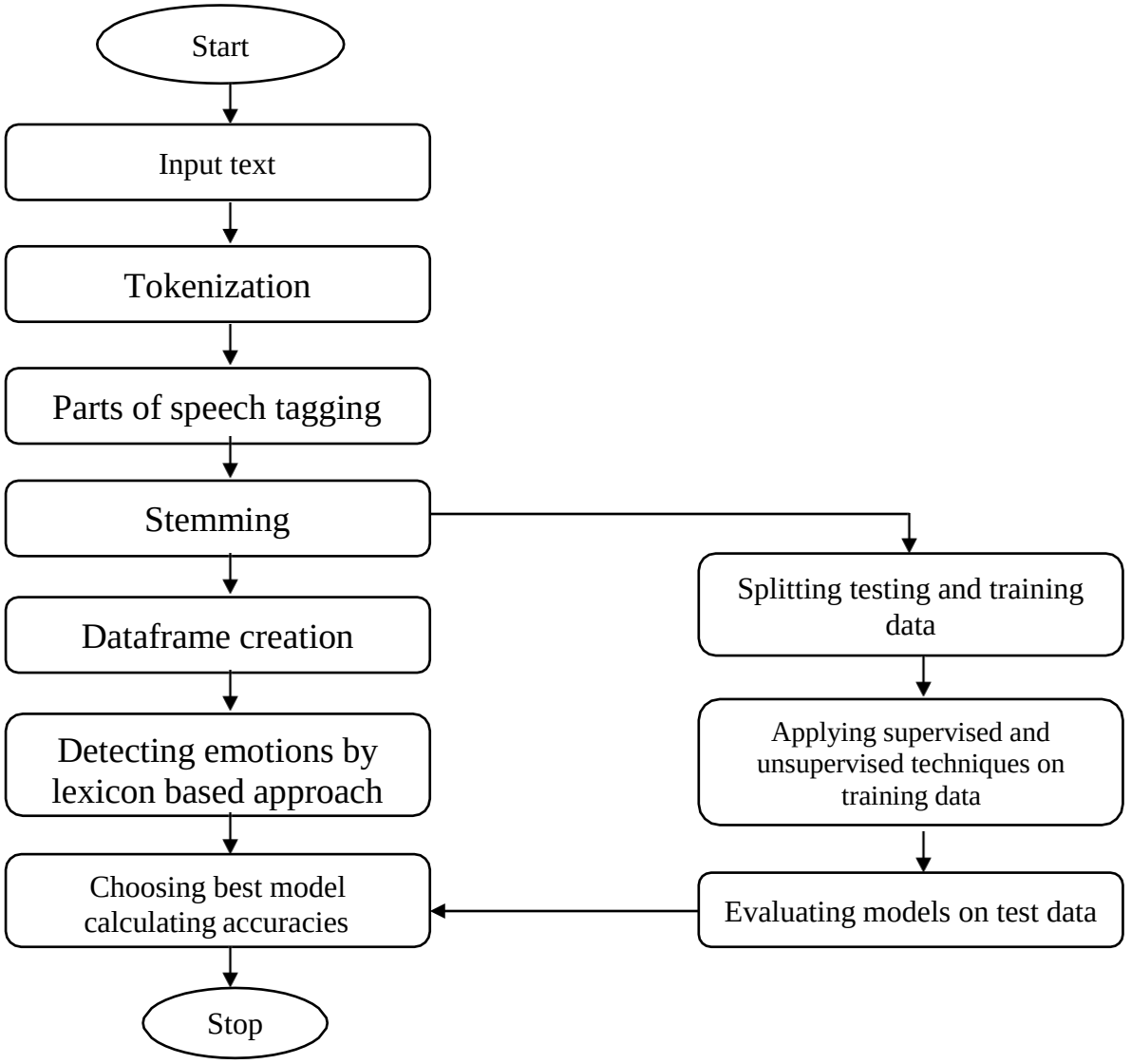

Fig 4.3.1.2 Activity Diagram

5. EXPERIMENTATION AND ANALYSIS

Jupyter Notebook platform is used for implementation of project. The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning, and much more.

5.1 EXPERIMENTATION

Dataset

joy	On days when I feel close to my partner and other friends.
fear	Every time I imagine that someone I love or I could contact a
anger	When I had been obviously unjustly treated and had no possibility
sadness	When I think about the short time that we live and relate it to
disgust	At a gathering I found myself involuntarily sitting next to two
shame	When I realized that I was directing the feelings of discontent
guilt	I feel guilty when when I realize that I consider material things
joy	After my girlfriend had taken her exam we went to her parent's
fear	When, for the first time I realized the meaning of death.
anger	When a car is overtaking another and I am forced to drive off the
sadness	When I recently thought about the hard work it takes to study, and
disgust	When I found a bristle in the liver paste tube.
shame	When I was tired and unmotivated, I shouted at my girlfriend and
guilt	When I think that I do not study enough. After the weekend I
joy	When I pass an examination which I did not think I did well.
fear	When one has arranged to meet someone and that person arrives
anger	When one is unjustly accused of something one has not done.
sadness	When one's studies seem hopelessly difficult and uninteresting.
disgust	When one finds out that someone you know is not at all like one
shame	When one has been unjust, stupid towards someone else.
guilt	When one has neglected or been unjust to a good friend.
joy	Passing an exam I did not expect to pass.
fear	When I climbed up a tree to pick apples. The angle of the ladder
guilt	When excuses are necessary and I get out of doing it myself.
joy	When I had my children.
fear	When my 2 year old son climbed up and sat on the 7th floor
anger	When my partner was attacked and lost three teeth.
sadness	When I see children on T.V from areas devastated by drought and
disgust	When I nearly walked on a blindworm and then saw it crawl away.
shame	When I saw my 18 year old son grab an oxygen mask as he had
guilt	I experience a sense of guilt as my middle son cannot express

Fig 5.1.1 Dataset Description

In the dataset the first column contains the emotion and the second column contains the sentence.

Data used for the experiments is ISEAR dataset. It contains sentences representing seven different kind of emotions joy, surprise, anger, sadness, shame, disgust and guilt. People express their emotions on social media through text.

Loading the dataset:

We can work on different types of datasets like json, csv, tsv, excel etc…Pandas is a great library to work with datasets.The Python API provides the module CSV and the function reader () that can be used to load CSV files. Pandas library contains the method read_csv to read csv files. While importing the dataset, the dataset should be present in the current working directory or the filename along with the path of the file should be given as an argument to the read_csv method of pandas.

Before importing the dataset, check the current working directory and set it to the directory where the data is available.

Text Preprocessing

Data preprocessing is an essential step in a NLP application. The very first step is to process the text. It is an important step before building the model. Text preprocessing is conversion of textual data tomore machine understandable form for learning. The efficiency of the model depends on how effectively the text is preprocessed.Analysis of the textual data can be done easily only when processing is done. Python programming is used for processing and the libraries of python can make it easier with some inbuilt functions.

On textual data many steps have to be applied to convert it from textual format to numerical format with which we can work. This is an important aspect for proceeding into further steps. Only then the machine learning techniques can be applied on the data. Text preprocessing can be different for different applications. All applications do not need the same steps to be performed. They vary depending on the type of the problem.

In general, the pre-processing task can be classified into four different steps. Cleaning of data, Annotation, Normalization and Analysis of the data.

• Cleaning of data involves removal of unnecessary special characters, stopwords, converting all the words to a single case.

• Annotation includes tagging parts of speech to the words and structural markup.

• Normalization of text consists of reduction of words by linguistic rules or other standardized methods. It includes Stemming, Lemmatization of text i.e. converting every word to its basic form.

• Analysis of data involves analyzing the data through generalizing for feature analysis. It can be done through visualization, wordcloud etc..

Broadly classified, the text preprocessing steps include tasks such as tokenization of data, removal of special characters, removal of stopwords, parts of speech tagging, stemming, Lemmatization and vectorization of the data through word2vec conversion , countvectorization etc..Many pre-built tools and libraries are available for processing and analyzing the text.

The nltk(Natural language tool kit) library of python is very efficient to handle the preprocessing tasks. It has in-built functions to perform the tasks through simple steps. Textblob is another python library which is powerful than nltk. It is built on the top of nltklibrary and handles the preprocessing steps efficiently and effectively than nltk. It has more functions and unique features compared to nltk.

Tokenization

Tokenization is the first step of preprocessing of text. It is the process of converting the sentences into a list of words. This is done by splitting the sentences into words and meaningful phrases. The result of tokenization is a list data type with words separated by commas. Not only the words, but the punctuations are also splitted. These splitted parts are called tokens. A token is a sequence of characters that give some meaning.These tokens are very useful for finding such patterns as well as is considered as a base step for stemming and lemmatization. It can be done by using the pre-defined functions of the libraries like nltk, textblob. It requires simple steps for conversion.

Stopwords Removal

Stopwords are the most commomly used words in the language for the construction of the sentence. They frame the sentence meaningfully. But for the analysis of text, these

stopwords can be ignored. In general the serach engines do not consider the stopwords for searching and they retrieve the related searches for the query based on the key words of the sentence. Stop words are ignored as they do not affect the search results of a query. These are ignored as they take up the space in the database and much time for processing. Stopwords are a list of words already available in the nltk library as a file which can be downloaded or imported from the library. From the list of tokenized words, the stopwords can be removed.

Parts Of Speech Tagging

Pos tagging is grammatical tagging, to remove the ambiguity among words. The tokenized words, after the removal of stopwords are tagged with the respective parts of speech. The result is a list of tuples, with each tuple containing the word with the tag in the form of(word,tag). Tag represents the part of speech the word belongs to and signifies whether the word is a noun, Pronoun possessive pronoun, verb, adjective, adverb. Not only the basic types, they are further classified into secondary categories and are represented by different tags.

Parts of speech tagging is done by Default Tagger class. Default tagging is the basic step in pos tagging. This DefaultTaggerclass takes a single argument ‘tag’. NN is the tag for a singular noun. DefaultTagger is most useful when it gets to work with most common partof-speech tag, that’s why a noun tag is recommended.

Penn Treebank used some tags used for representation of the parts of speech to which the word belongs to. Some of the most used tags are

NN – Noun, singular or mass

NNS – Noun,plural

JJ – Adjective

RB – Adverb

RBR – Adverb, comparative

VB – Verb, base form

VBD – Verb, past tense

Getting NAVA words

By tagging parts of speech to all the words, we will be able to detect the type of words. Of all the words, we retrieve only those words which have some meaning and may add to the emotion of the sentence. Those words mainly belong to the parts of

speech like nouns, adjectives, verbs and adverbs .All other words are already stemmed, so there is no chance of missing the words which may represent emotions. The words with the tags NN(nouns), VB(verbs), JJ(adjectives), RB(adverb) are considered as the key words that may represent emotion.

It can be done using the tags as reference. This can be done by the use of regular expressions. The method ‘startswith’ can be used to check the words and are retrieved. Only NAVA N-Nouns, A-Adjectives, V-Verbs, A-Adverbs words are retrieved which are helpful in detecting the emotions. We store both the list of the tuples, nava words with their tags and nava words as a list

Stemming

Stemming is a linguistic morphology of reducing a word to its stem, base word by removing the suffixes, prefixeswhich inflect the word. Stemming is essential in the field of 26 naturallanguageprocessing(NLP) and natural language understanding(NLU).Stemming helps in reducing the ambiguity in understanding the context of the text. Recognizing, searching and retrieving more forms of words returns more results. When a form of a word is recognized it can make it possible to return search results that otherwise might have been missed. That additional information retrieved is why stemming is integral to search queries and information retrieval.

The words are stemmed to its root word. The stem word need not match the base word in dictionary. It is just a basic from of the base word, it can be equal to the actual word or even smaller than the word. There are many stemming algorithms for performing stemming.

Porter stemmer: This stemming algorithm is old and efficient. It is a simple algorithm and its main goal is to remove the endings to words so that they can be transformed to their basic from. This algorithm does not work efficiently for complex tasks. It is a gentle algorithm and easy to handle compared to all other stemming algorithms. It is a nice, basic algorithm to work with simple applications.

Data Frame Creation

A data frame is a two dimensional array with rows and columns. Each column contains variables and the rows contain values for the columns. Selecting the rows and

columns by the help of words and indices can be done easily through Data frame APIs. They can also be used for slicing the data, normalizing he data and perform several operations on the data. A data frame can be comprehended and understood easily. A data frame can consist of any number of datasets.

A Data Frame can be created using the pandas library by loading the datasets from existing storage, storage can be SQL Database, CSV file, and Excel file. Pandas data Frame can be created from the lists, dictionary, and from a list of dictionary etc. Pandas contain a method Dataframe() for the creation of data frame from a single dataset or multiple datasets. In data frame datasets arrange in rows and columns, we can store any number of datasets in a data frame. We can perform many operations on these datasets like arithmetic operation, columns/rows selection, columns/rows addition etc.

Seven emotions and their words

Shame

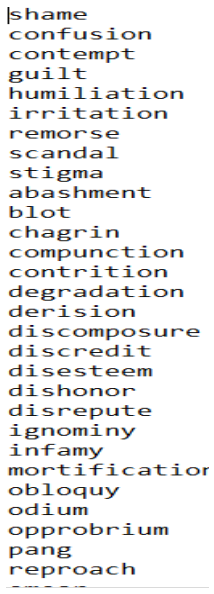

Fig 5.1.2 words of shame emotion

Sadness

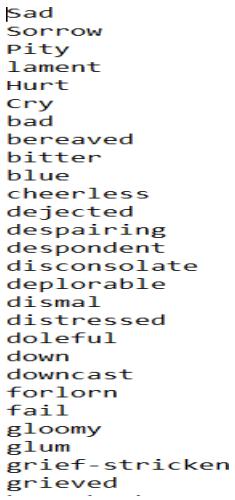

Fig 5.1.3 words of sadness emotion

Joy

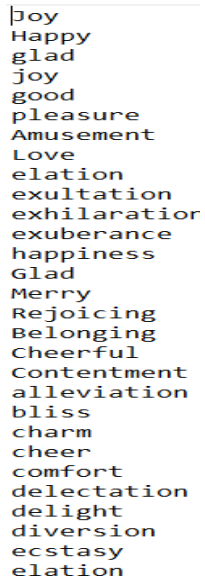

Fig 5.1.4 words of joy emotion

Guilt

guilt
culpability
disgrace
indiscretion
liability
regret
remorse
responsibilit
shame
sin
stigma
answerability
blameworthine
contrition
crime
criminality
delinquency
dereliction
dishonor
error
failing
fault
infamy
iniquity

Fig 5.1.5 words of guilt emotion

Disgust

disgust
sicken
nuseate
antipathy
dislike
distaste
hatred
loath
revulsion
repel
grossed
abhorrencesta
rabomination
detestation
hatefulness
nausea
objection
repugnance
revolt
satiation
satiety
sickness
surfeit
nauseation

Fig 5.1.6 words of disgust emotion

Anger

Angry
Irritate
Stupid
Annoy
Furious
rage
Frustrate
acrimony
animosity
annoyance
antagonism
conniption
dander
disapprobation
displeasure
distemper
enmity
tempestuous
turbulent
stormy

Fig 5.1.7 Words of anger emotion

Fear

fear
afraid
frighten
scare
venerate
terrify
abhorrence
agitation
angst
anxiety
aversion
awe
concern
consternation
cowardice
creeps
death
despair
discomposure
dismay

Fig 5.1.8 words of fear emotion

5.2 RESULTS

5.2.1 Screen Shots

Nava_tokens

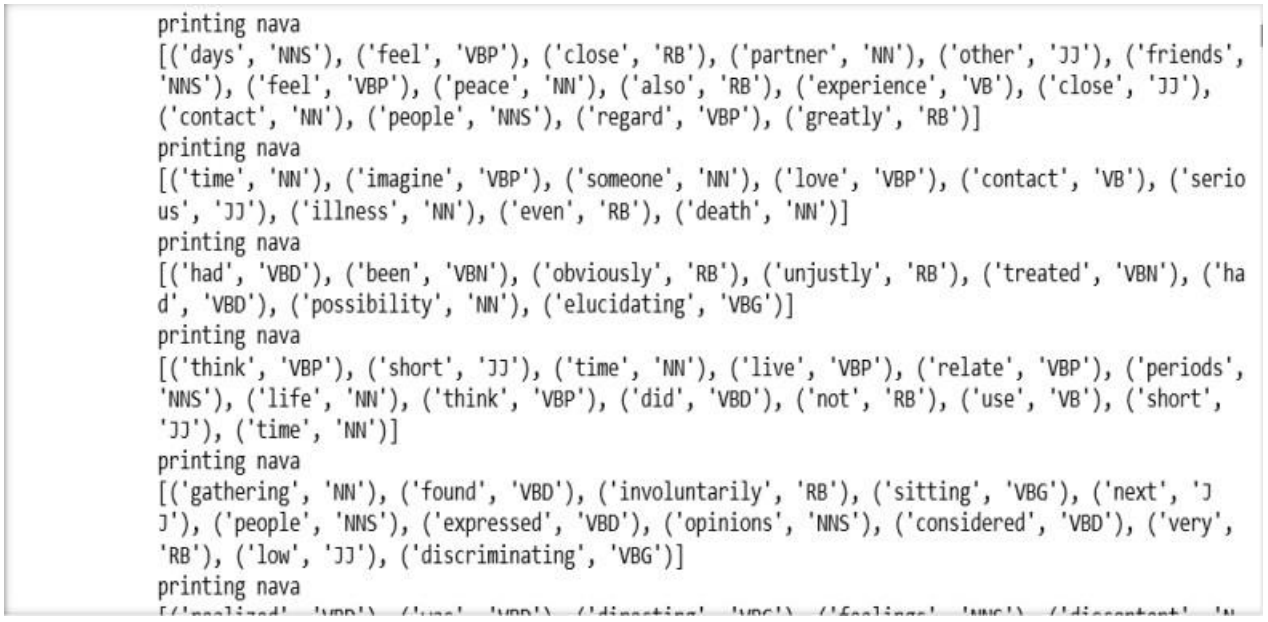

Fig 5.2.1.1 Fig Nava_ tokens

Nava_words

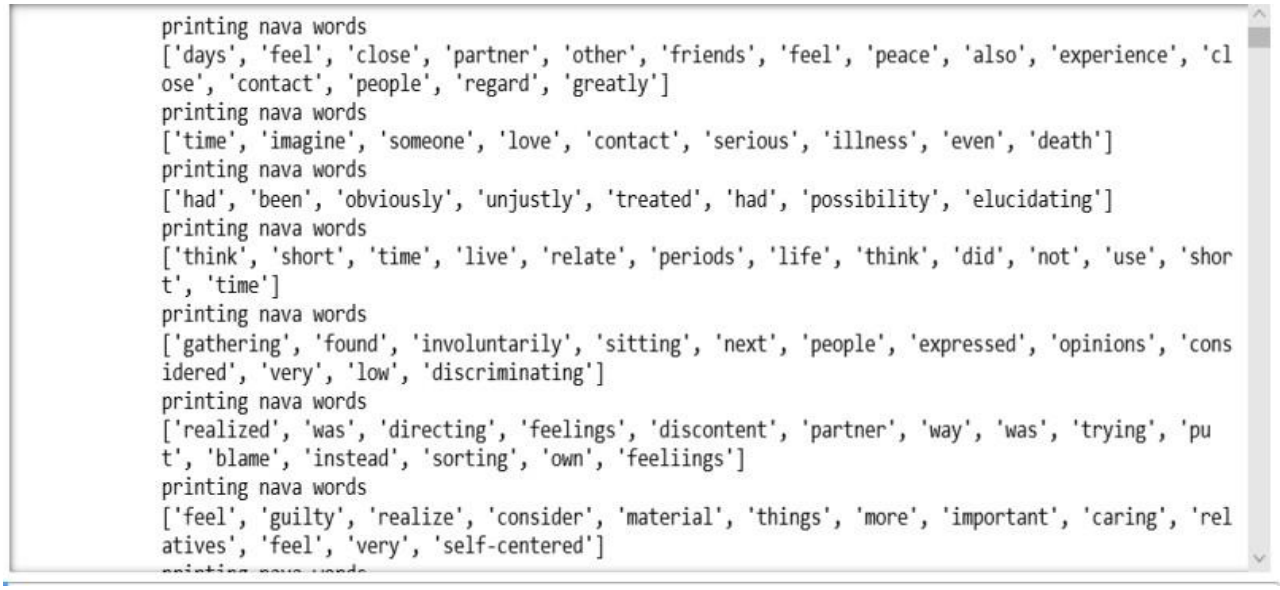

Fig 5.2.1.2 Nava _Words

Dataframe – df

```
In [12]: df
```

Out[12]:

	0	1	2	3	4
0	joy	(day, feel, close, partner, other, friend, fee...	day feel close partner other friend feel peac ...	[day, feel, close, partner, other, friend, fee...	[joy]
1	fear	(time, imagin, someon, love, contact, seriou, ...	time imagin someon love contact seriou ill eve...	[time, imagin, someon, love, contact, seriou, ...	[fear]
2	anger	(had, been, obvious, unjustli, treat, had, pos...	had been obvious unjustli treat had possibl el...	[had, been, obvious, unjustli, treat, had, pos...	[anger]
3	sadness	(think, short, time, live, relat, period, life...	think short time live relat period life think ...	[think, short, time, live, relat, period, life...	[sadness]
4	disgust	(gather, found, involuntarili, sit, next, peop...	gather found involuntarili sit next peopl expr...	[gather, found, involuntarili, sit, next, peop...	[disgust]
5	shame	(realiz, wa, direct, feel, discont, partner, w...	realiz wa direct feel discont partner way wa t...	[realiz, wa, direct, feel, discont, partner, w...	[shame]
6	guilt	(feel, guilti, realiz, consid, materi, thing, ...	feel guilti realiz consid materi thing more im...	[feel, guilti, realiz, consid, materi, thing, ...	[guilt]
		(girlfriend, had, taken, exam	girlfriend had taken exam went	[girlfriend, had, taken, exam	

Fig 5.2.1.3 Dataframe

Sentences divided into tokens :sentoken

```
In [14]: sentoken

Out[14]: [[u'day',
          u'feel',
          u'close',
          u'partner',
          u'other',
          u'friend',
          u'feel',
          u'peac',
          u'also',
          u'experi',
          u'close',
          u'contact',
          u'peopl',
          u'regard',
          u'greatli'],
         [u'time',
          u'imagin',
          u'someon',
          u'love',
          u'contact'
```

Fig 5.2.1.4 sentoken

Sentences in string format :Senstr

```
In [16]: senstr

Out[16]: [u'day feel close partner other friend feel peac also experi close contact peopl regard greatli ',
          u'time imagin someon love contact seriou ill even death ',
          u'had been obvious unjustli treat had possibl elucid ',
          u'think short time live relat period life think did not use short time ',
          u'gather found involuntarili sit next peopl express opinion consid veri low discrimin ',
          u'realiz wa direct feel discont partner way wa tri put blame instead sort own feeli ',
          u'feel guilti realiz consid materi thing more import care rel feel veri self-cent ',
          'girlfriend had taken exam went parent place ',
          u'first time realiz mean death ',
          u'car overtak forc drive road ',
          u'recent thought hard work take studi want tri someth els read theoret book english did not unders
         tand ',
          u'found bristl liver past tube ',
          u'wa tire unmotiv shout girlfriend brought neg side charact are actual not import ',
          u'think not studi enough weekend think have been abl have accomplish someth time ',
          u'pass examin did not think did well ',
          u'one ha arrang meet someon person arriv late meantim start think have gone wrong e.g traffic acci
         d ',
          u'unjustli accus someth ha not done ',
```

Fig 5.2.1.5 Sentence into string format

Labels

```
In [15]: labels

Out[15]: ['joy',
          'fear',
          'anger',
          'sadness',
          'disgust',
          'shame',
          'guilt',
          'joy',
          'fear',
          'anger',
          'sadness',
          'disgust',
          'shame',
          'guilt',
          'joy',
          'fear',
          'anger',
          'sadness',
          'disgust',
```

Fig 5.2.1.6 Labels

Word_set

```
printing representative words or synonyms of emotion : joy
['joy', u'happi', 'glad', 'good', u'pleasur', u'amus', 'love', u'elat', u'exult', u'exhilar', u'exube
r', u'merri', u'rejoic', u'belong', u'cheer', u'content', u'allevi', u'bliss', 'charm', 'comfort',
u'delect', 'delight', u'divers', u'ecstasi', u'felic', u'festiv', 'frolic', 'fruition', u'gaieti', 'g
lee', 'humor', u'gratif', u'hilar', u'jubil', u'liveli', u'luxuri', 'merriment', u'ravish', u'refres
h', u'regal', u'revelri', u'satisfact', u'solac', u'treasur', u'bless', u'captiv', 'chipper', u'chirp
i', u'convivi', u'ecstat', 'gay', 'gleeful', u'gratifi', u'intox', u'jolli', u'joyou', u'laugh', 'lig
ht', u'live', u'mirth', u'overjoy', u'peac', u'peppi', u'perki', u'play', 'pleasant', u'pleas']
printing representative words or synonyms of emotion : fear
['fear', 'afraid', 'frighten', 'scare', u'vener', u'terrifi', u'abhorr', u'agit', 'angst', u'anxiet
i', u'avers', 'awe', 'concern', u'constern', u'cowardic', u'creep', 'death', 'despair', u'discomposu
r', 'dismay', u'disquietud', u'distress', 'doubt', 'dread', u'fainthearted', u'forebod', 'fright', 'f
unk', 'horror', u'jitter', u'misgiv', u'nightmar', 'panic', 'phobia', u'presenti', 'qualm', u'recrean
c', u'rever', u'revuls', 'suspicion', 'terror', u'timid', u'trembl', 'tremor', u'trepid', u'uneas',
u'uneasi', u'worri', 'aghast', u'anxiou', u'panicki', u'petrifi', 'shaken', u'startl']
printing representative words or synonyms of emotion : anger
[u'angri', u'irrit', 'stupid', 'annoy', u'furiou', 'rage', u'frustrat', u'acrimoni', u'animos', u'ant
agon', u'connipt', 'dander', u'disapprob', u'displeasur', u'distemp', u'enmiti', u'tempestu', u'turbu
l', u'stormi', u'exasper', u'furi', 'gall', u'hatr', 'huff', u'impati', u'indign', u'infuri', u'iras
c', 'ire', 'mad', 'miff', u'outrag', 'passion', u'peevish', u'petul', u'piqu', u'rankl', u'resent',
'slowburn', u'sore', 'stew', 'storm', 'tantrum', 'temper', 'tiff', u'umbrag', u'vexat', u'violenc',
u'tortur']
printing representative words or synonyms of emotion : sadness
['sad', 'sorrow', u'piti', 'lament', 'hurt', u'cri', 'bad', u'bereav', 'bitter', 'blue', u'cheerles
s', u'deject', u'despair', u'despond', u'disconsol', u'deplor', 'dismal', u'distress', u'dole', 'dow
n', 'downcast', 'forlorn', 'fail', u'gloomi', 'glum', 'grief-stricken', u'griev', 'heartbroken', 'hea
rtsick', u'heavyheart', u'languish', 'low', u'low-spirit', u'lugubri', 'lost', u'melancholi', 'morbi
d', u'moros', u'mourn', u'pensiv', u'pessimist', 'somber', u'sorri', u'troubl', u'weep', u'wist', u'w
```

Fig 5.2.1.7 word_set

```
d', u'moros', u'mourn', u'pensiv', u'pessimist', 'somber', u'sorri', u'troubl', u'weep', u'wist', u'w
oebegon']
printing representative words or synonyms of emotion : disguist
['disgust', 'sicken', u'nuseat', u'antipathi', u'dislik', u'distast', u'hatr', 'loath', u'revuls', 'r
epel', u'gross', 'abhorrencesta', u'rabomin', u'detest', u'hate', 'nausea', u'object', u'repugn', 're
volt', u'satiat', u'satieti', u'sick', 'surfeit', u'nauseat', u'nauseous', u'nauseou', u'abomin', u'a
w', u'creepi', u'gruesom', u'horrif', u'loathsom', u'nasti', u'objection', u'obnoxi', u'odiou', u'out
rag', u'scandal', u'shameless', u'shock', 'vile', 'vulgar', u'cloy', 'foul', u'stink', u'beastli',
u'fright', u'ghastli', u'grodi', u'hideou', 'horrid', u'icki', u'lousi', u'macabr', u'monstrou', u'ne
rdi', u'noisom', u'offens', 'rotten', u'scuzzi', u'sleazebal', u'sleazi', u'tortur', u'yecchi', u'yuc
ki']
printing representative words or synonyms of emotion : shame
['shame', u'confus', 'contempt', 'guilt', u'humili', u'irrit', u'remors', 'scandal', 'stigma', u'abas
h', 'blot', 'chagrin', u'compunct', u'contrit', u'degrad', u'deris', u'discomposur', 'discredit', 'di
sesteem', 'dishonor', u'disreput', u'ignomini', u'infami', u'mortif', u'mistak', 'obloquy', 'odium',
'opprobrium', 'pang', 'reproach', 'smear', u'stupefact', u'treacheri', u'pudenc', u'dastardli', u'dis
grac', u'embarrass', 'flagrant', u'heinou', u'immor', u'indec', u'outrag', u'reprehens', u'shock',
u'sin', 'vile', 'base', 'carnal', 'corrupt', u'debauch', u'diabol', 'drunken', 'immodest', u'impur',
u'infam', u'intemper', 'lewd', 'low', 'mean', u'mortifi', u'notori', u'obscen', u'opprobri', u'profli
g', u'reprob', 'ribald', u'unbecom', 'unclean', u'unabl', u'unworthi', u'villain', 'vulgar', u'wick']
printing representative words or synonyms of emotion : guilt
['guilt', u'culpabl', u'disgrac', u'indiscret', u'liabil', 'regret', u'remors', u'respons', 'shame',
'sin', 'stigma', u'answer', u'blameworthi', u'contrit', 'crime', u'crimin', u'delinqu', u'derelict',
'dishonor', 'error', u'fail', 'fault', u'infami', u'iniqu', u'laps', u'malfeas', u'malpractic', 'misb
ehavior', 'misconduct', 'misstep', u'offens', u'onu', u'penit', 'slip', u'solec', u'transgress', u'wi
cked', 'wrong', u'malefact', u'peccabl', u'convict', u'liabl', u'sorri', u'accus', 'caught', u'censu
r', u'chargeabl', u'condemn', u'damn', u'deprav', u'doom', 'err', 'evil', u'feloni', 'hangdog', u'imp
each', u'incrimin', u'iniquit', u'judg', u'licenti', u'offend', u'proscrib', u'reprehens', 'rueful',
u'sentenc', 'sheepish', u'wick']
```

Fig 5.2.1.8 word_set

Textvector

```
printing text vector
[[{u'peac': ['joy']}], [{u'love': ['joy']}, {u'death': ['fear']}], [], [{u'live': ['joy']}], [{u'lo
w': ['shame', 'sadness']}], [], [], [], [{u'mean': ['shame']}, {u'death': ['fear']}], [], [], [],
[], [], [], [{u'wrong': ['guilt']}], [{u'accus': ['guilt']}], [], [], [{u'stupid': ['anger']}],
[{u'good': ['joy']}], [], [], [], [], [{u'lost': ['sadness']}], [{u'lost': ['sadness']}], [], [],
[{u'bad': ['sadness']}, {u'guilt': ['shame', 'guilt']}], [{u'guilt': ['shame', 'guilt']}], [], [],
[{u'accus': ['guilt']}], [{u'fail': ['sadness', 'guilt']}], [{u'disgust': ['disguist']}], [{u'caugh
t': ['guilt']}], [], [], [], [], [], [], [{u'good': ['joy']}], [], [], [{u'frighten': ['fear']},
{u'frighten': ['fear']}], [], [], [{u'disgust': ['disguist']}], [{u'good': ['joy']}], [], [], [],
[], [], [], [], [{u'fail': ['sadness', 'guilt']}], [{u'storm': ['anger']}, {u'scare': ['fear']}],
[], [], [], [{u'intox': ['joy']}], [], [], [], [], [], [], [{u'good': ['joy']}], [], [{u'pleasant':
['joy']}, {u'happi': ['joy']}, {u'good': ['joy']}], [{u'lost': ['sadness']}], [{'mad': ['anger']}],
[{u'love': ['joy']}], [{u'bad': ['sadness']}], [{u'caught': ['guilt']}], [], [{u'happi': ['joy']}],
[{u'caught': ['guilt']}, {u'storm': ['anger']}], [], [], [], [{u'laugh': ['joy']}], [{u'caught':
['guilt']}], [], [], [{u'hurt': ['sadness']}], [], [], [], [{u'regret': ['guilt']}], [], [{'wrong':
['guilt']}], [], [], [], [], [], [], [{u'scare': ['fear']}], [{u'hatr': ['disguist', 'anger']}],
[{u'afraid': ['fear']}, {u'fail': ['sadness', 'guilt']}], [{u'terrifi': ['fear']}, {u'disgust': ['d
isguist']}], [], [{u'mistak': ['shame']}, {u'wrong': ['guilt']}], [], [], [{u'caught': ['guilt']}],
[{u'fail': ['sadness', 'guilt']}], [], [], [], [{'live': ['joy']}], [], [], [], [{u'troubl': ['sadn
```

Fig 5.2.1.9 TextVector

Lexicon accuracy

```
[ ] print( accuracy*100,"%")

    53.11526479750779 %
```

Fig 5.2.10 Lexicon Accuracy

Training data for NaiveBayes

```
print("printing train data")
print(data_se)
```

```
printing train data
[(['day', 'feel', 'close', 'partner', 'other', 'friend', 'feel', 'peac', 'also', 'experi', 'close',
'contact', 'peopl', 'regard', 'greatli'], 'joy'), (['time', 'imagin', 'someon', 'love', 'contact',
'seriou', 'ill', 'even', 'death'], 'fear'), (['had', 'been', 'obvious', 'unjustli', 'treat', 'had',
'possibl', 'elucid'], 'anger'), (['think', 'short', 'time', 'live', 'relat', 'period', 'life', 'thi
nk', 'did', 'not', 'use', 'short', 'time'], 'sadness'), (['gather', 'found', 'involuntarili', 'si
t', 'next', 'peopl', 'express', 'opinion', 'consid', 'veri', 'low', 'discrimin'], 'disgust'), (['re
aliz', 'wa', 'direct', 'feel', 'discont', 'partner', 'way', 'wa', 'tri', 'put', 'blame', 'instead',
'sort', 'own', 'feeli'], 'shame'), (['feel', 'guilti', 'realiz', 'consid', 'materi', 'thing', 'mor
e', 'import', 'care', 'rel', 'feel', 'veri', 'self-cent'], 'guilt'), (['girlfriend', 'had', 'take
n', 'exam', 'went', 'parent', 'place'], 'joy'), (['first', 'time', 'realiz', 'mean', 'death'], 'fea
r'), (['car', 'overtak', 'forc', 'drive', 'road'], 'anger'), (['recent', 'thought', 'hard', 'work',
'take', 'studi', 'want', 'tri', 'someth', 'els', 'read', 'theoret', 'book', 'english', 'did', 'no
t', 'understand'], 'sadness'), (['found', 'bristl', 'liver', 'past', 'tube'], 'disgust'), (['wa',
'tire', 'unmotiv', 'shout', 'girlfriend', 'brought', 'neg', 'side', 'charact', 'are', 'actual', 'no
t', 'import'], 'shame'), (['think', 'not', 'studi', 'enough', 'weekend', 'think', 'have', 'been',
'abl', 'have', 'accomplish', 'someth', 'time'], 'guilt'), (['pass', 'examin', 'did', 'not', 'thin
k', 'did', 'well'], 'joy'), (['one', 'ha', 'arrang', 'meet', 'someon', 'person', 'arriv', 'late',
'meantim', 'start', 'think', 'have', 'gone', 'wrong', 'e.g', 'traffic', 'accid'], 'fear'), (['unjus
```

Fig 5.2.1.11 Training data for Navie Bayes

Testing data for Naïve Bayes

```
print("printing test data")
print(testdata)
```

```
printing test data
[['be put class leader ', 'anger'], ['find life span china shorter west ', 'sadness'], ['girl dress
foreign univers ', 'disgust'], ['be insult public ', 'shame'], ['insult other peopl ', 'disgust'],
['found travel best friend ', 'joy'], ['walk home dark colleg ', 'fear'], ['find wa deceiv friend
', 'anger'], ['not do well examn ', 'shame'], ['find best friend wa deceiv ', 'anger'], ['fail exam
n did not work hard enough ', 'guilt'], ['not act promis ', 'shame'], ['find not ill not serious ',
'joy'], ['find health condit attend univers lectur ', 'fear'], ['bought someth bad shop refus chang
', 'anger'], ['rel death ', 'sadness'], ['saw peopl quarrel bu ', 'disgust'], ['run away fire ', 's
adness'], ['forgot give present littl nephew ', 'guilt'], ['first public speak ', 'joy'], ['go alon
dark ', 'fear'], ['saw young peopl fight seat bu ', 'disgust'], ['mother death ', 'sadness'], ['saw
peopl spit public ', 'disgust'], ['went lectur chines histori hear opium war ', 'shame'], ['fall lo
ve ', 'joy'], ['friend find love ', 'sadness'], ['fall love friend angri ', 'anger'], ['friend stil
l trust friend ', 'sadness'], ['someon be arrog ', 'disgust'], ['insult teacher front class ', 'sha
me'], ['fall love close friend ', 'sadness'], ['do unexpectedli well examn ', 'joy'], ['examn ', 'j
oy'], ['want borrow lectur note friend did not lend ', 'guilt'], ['find univers doe not have enough
foreign currenc buy yournal refer list ', 'anger'], ['someon told wa chosen english lectur wa good
friend class leader ', 'shame'], ['find chines money not buy foreign journal ', 'anger'], ['late le
ctur therefor miss ', 'guilt'], ['talk close friend ', 'joy'], ['ran park car bicycl ', 'fear'],
['insult reason ', 'shame'], ['did badli examn ', 'sadness'], ['saw man dress woman ', 'disgust'],
```

Fig 5.2.12 Testing data for navie bayes

All_words

```
printing all the words in dataset
['day', 'feel', 'close', 'partner', 'other', 'friend', 'feel', 'peac', 'also', 'experi', 'close',
'contact', 'peopl', 'regard', 'greatli', 'time', 'imagin', 'someon', 'love', 'contact', 'seriou',
'ill', 'even', 'death', 'had', 'been', 'obvious', 'unjustli', 'treat', 'had', 'possibl', 'elucid',
'think', 'short', 'time', 'live', 'relat', 'period', 'life', 'think', 'did', 'not', 'use', 'short',
'time', 'gather', 'found', 'involuntarili', 'sit', 'next', 'peopl', 'express', 'opinion', 'consid',
'veri', 'low', 'discrimin', 'realiz', 'wa', 'direct', 'feel', 'discont', 'partner', 'way', 'wa', 't
ri', 'put', 'blame', 'instead', 'sort', 'own', 'feeli', 'feel', 'guilti', 'realiz', 'consid', 'mate
ri', 'thing', 'more', 'import', 'care', 'rel', 'feel', 'veri', 'self-cent', 'girlfriend', 'had', 't
aken', 'exam', 'went', 'parent', 'place', 'first', 'time', 'realiz', 'mean', 'death', 'car', 'overt
ak', 'forc', 'drive', 'road', 'recent', 'thought', 'hard', 'work', 'take', 'studi', 'want', 'tri',
'someth', 'els', 'read', 'theoret', 'book', 'english', 'did', 'not', 'understand', 'found', 'brist
l', 'liver', 'past', 'tube', 'wa', 'tire', 'unmotiv', 'shout', 'girlfriend', 'brought', 'neg', 'sid
e', 'charact', 'are', 'actual', 'not', 'import', 'think', 'not', 'studi', 'enough', 'weekend', 'thi
nk', 'have', 'been', 'abl', 'have', 'accomplish', 'someth', 'time', 'pass', 'examin', 'did', 'not',
'think', 'did', 'well', 'one', 'ha', 'arrang', 'meet', 'someon', 'person', 'arriv', 'late', 'meanti
m', 'start', 'think', 'have', 'gone', 'wrong', 'e.g', 'traffic', 'accid', 'unjustli', 'accus', 'som
eth', 'ha', 'not', 'done', 'one', 'studi', 'seem', 'hopelessli', 'difficult', 'uninterest', 'find',
'someon', 'know', 'not', 'had', 'thought', 'instanc', 'friend', 'steal', 'thing', 'quit', 'unwarr',
```

Fig 5.2.13 All words in the dataset

Frequency distribution of all words

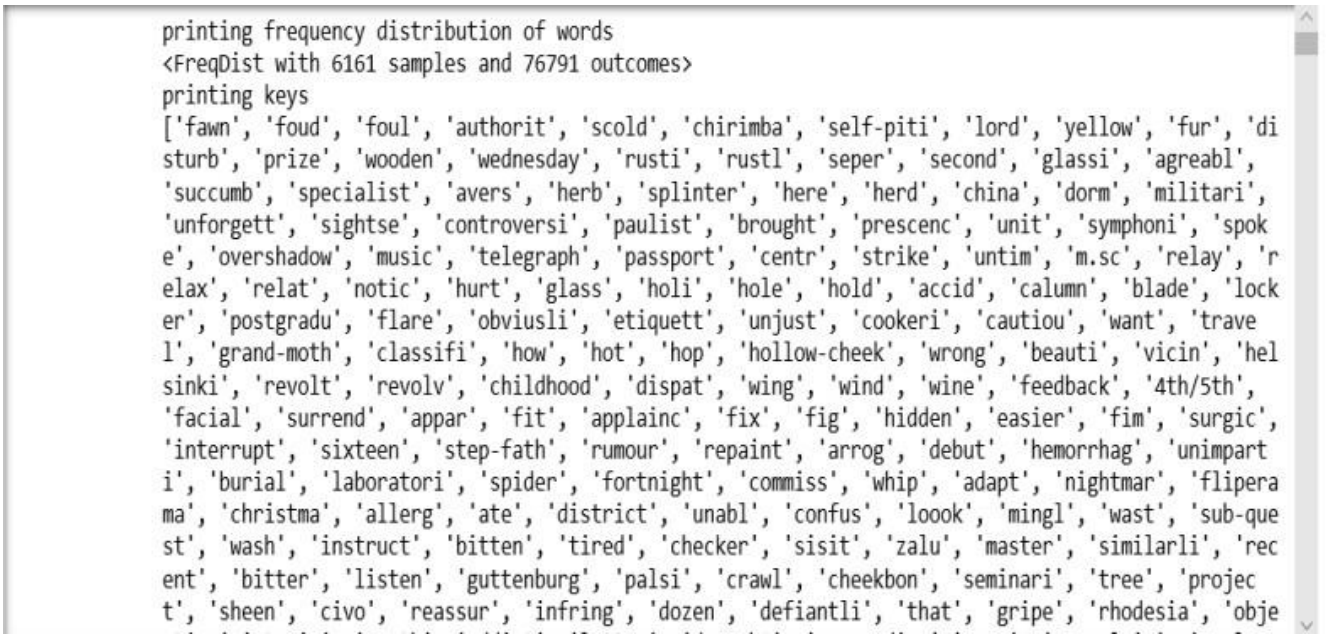

Fig 5.2.14 Frequency distribution of all words

Values of the keys

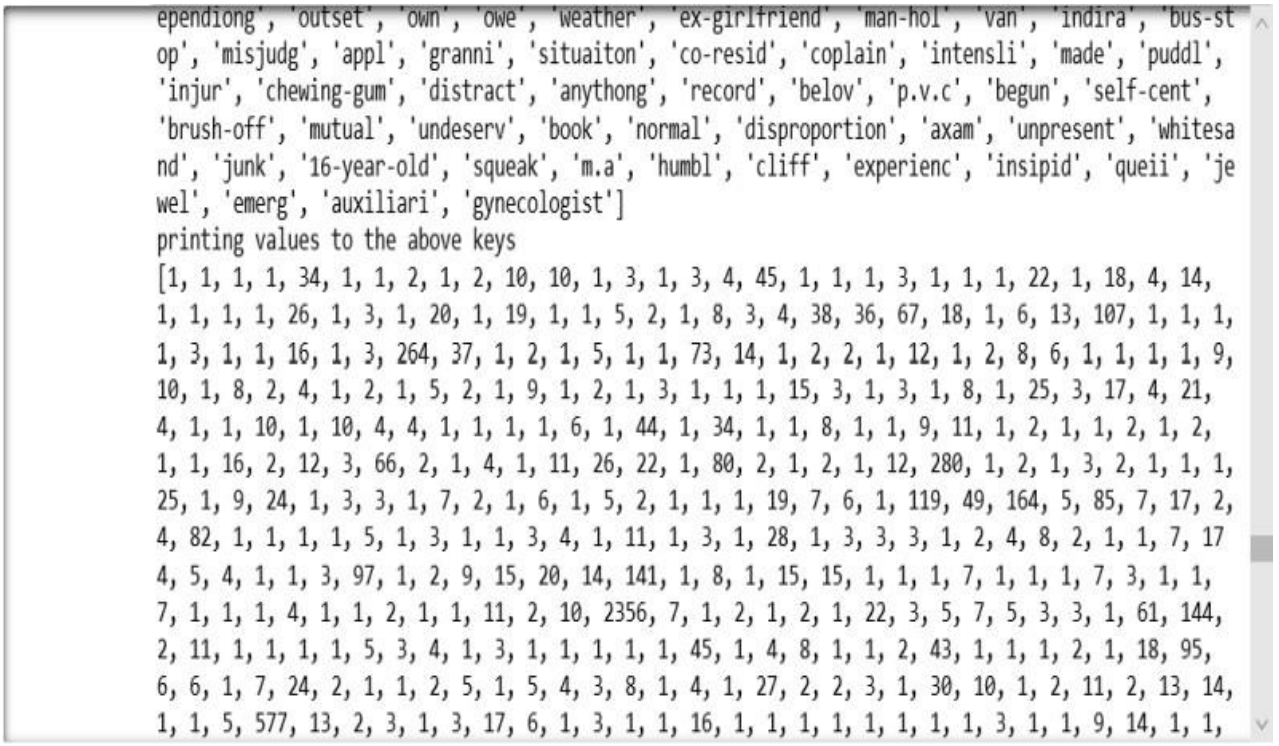

Fig 5.2.15 Values of the keys

Extracting features

```
extracting features
{'contains(kidney)': False, 'contains(clumsi)': False, 'contains(downhil)': False, 'contains(cereb
r)': False, 'contains(becam)': False, 'contains(fer)': False, 'contains(everybobi)': False, 'contai
ns(would-b)': False, 'contains(bibl)': False, 'contains(coffe)': False, 'contains(laboratori)': Fal
se, 'contains(case)': False, 'contains(cheris)': False, 'contains(heavyli)': False, 'contains(sla
p)': False, 'contains(auxiliari)': False, 'contains(tile)': False, 'contains(pullov)': False, 'cont
ains(diamet)': False, 'contains(withdraw)': False, 'contains(birthray)': False, 'contains(bunk)': F
alse, 'contains(templ)': False, 'contains(vice-versa)': False, 'contains(boycot)': False, 'contains
(expostul)': False, 'contains(convoc)': False, 'contains(rent)': False, 'contains(teeth)': False,
'contains(refriger)': False, 'contains(els)': False, 'contains(scrubber)': False, 'contains(spoi
l)': False, 'contains(fan)': False, 'contains(hospit)': False, 'contains(guidanc)': False, 'contain
s(alcool)': False, 'contains(comment)': False, 'contains(dnce)': False, 'contains(compass)': False,
'contains(copperbelt)': False, 'contains(statist)': False, 'contains(sorrow)': False, 'contains(rap
e)': False, 'contains(sproul)': False, 'contains(polythecn)': False, 'contains(thereaft)': False,
'contains(vet)': False, 'contains(unkind)': False, 'contains(import)': False, 'contains(preclin)':
False, 'contains(suck)': False, 'contains(tightli)': False, 'contains(drank)': False, 'contains(hor
ribl)': False, 'contains(switch)': False, 'contains(aborigin)': False, 'contains(rain)': False, 'co
ntains(misbehav)': False, 'contains(crake)': False, 'contains(perspect)': False, 'contains(h.s.c)':
False, 'contains(nonsens)': False, 'contains(snack)': False, 'contains(strictli)': False, 'contains
```

Fig 5.2.16 Extracting Features

```
a='my name is saic saic'
c=nltk.tokenize.word_tokenize(a)
b=nltk.FreqDist(c)
keys=b.keys()
values=b.values()
print(keys),
print(values)
```

```
['saic', 'is', 'my', 'name'] [2, 1, 1, 1]
```

Fig 5.2.17 keys and values

Naïve bayes Accuracy

```
[ ] Naive_accuracy = accuracy(testdata, classifier)
```

```
62.56544502617801
```

Fig 5.2.18 Naive bayes accuracy

Classification using naïve bayes

```
data_classification("am fed up with this")
```

```
'anger'
```

```
data_classification("i love sweets")
```

```
'joy'
```

```
data_classification("i don't like to dance public")
```

```
'shame'
```

```
data_classification("i ran away from that dark place")
```

```
'fear'
```

```
data_classification("i wouldn't have beat her")
```

```
'disgust'
```

Fig 5.2.19 Classification of naïve bayes

Labels of test data

```
print("printing predicted labels of test data using naive bayes")
for i in range(len(testdata)):
    print(testdata[i][0]),
    print(" -> "),
    print(data_classification(testdata[i][0]))
```

```
printing predicted labels of test data using naive bayes
be put class leader   ->  anger
find life span china shorter west   ->  sadness
girl dress foreign univers   ->  joy
be insult public   ->  shame
insult other peopl   ->  disgust
found travel best friend   ->  sadness
walk home dark colleg   ->  fear
find wa deceiv friend   ->  anger
not do well examn   ->  shame
find best friend wa deceiv   ->  anger
fail examn did not work hard enough   ->  guilt
not act promis   ->  shame
find not ill not serious   ->  sadness
find health condit attend univers lectur   ->  joy
bought someth bad shop refus chang   ->  guilt
rel death   ->  sadness
saw peopl quarrel bu   ->  disgust
run away fire   ->  sadness
```

Activate Windows
Go to Settings to activate

Fig 5.2.20 predicted labels of testdata using naïve bayes

Prediction of labels of test data using semantic similarity

```
printing predictions of test data using semantic similarity
0 be put class leader  -> shame
1 find life span china shorter west  -> shame
2 girl dress foreign univers  -> joy
3 be insult public  -> shame
4 insult other peopl  -> guilt
5 found travel best friend  -> fear
6 walk home dark colleg  -> sadness
7 find wa deceiv friend  -> guilt
8 not do well examn  -> shame
9 find best friend wa deceiv  -> anger
10 fail examn did not work hard enough  -> joy
11 not act promis  -> disgust
12 find not ill not serious  -> disgust
13 find health condit attend univers lectur  -> joy
14 bought someth bad shop refus chang  -> sadness
15 rel death  -> sadness
16 saw peopl quarrel bu  -> disgust
17 run away fire  -> anger
```

Fig 5.2.21 Predicted labels of testdata using semantic similarity

Accuracy using Semantic Similarity approach

```
[ ] print (semantic_accuracy*100,'%')

    79.94757536041939 %
```

Fig 5.2.22 Accuracy using semantic similarity

5.3 TESTING

Testing is the process of running the program with the goal of detecting errors present in the software. Testing is used to make the software free of errors and to perform accurately as expected. At the end of the testing phase, we will be able to obtain the software with zero errors. Software testing is an examination led to provide access to partners about the nature of the product. Software testing can give a mark, independent view on the product to allow the business to admit and gives a picture of dangers involved in programming execution.

Testing methods include the way towards executing a program or application with the purpose of discovering bugs in a program and confirms the application

product item is fit to launch into the market. Testing refers to the execution of a product's entity or framework segment to access at least one of the properties. When everything is done, these properties show the degree to which the segment or framework under test:

- meets the essential that lead its structure and advancement,
- reacts effectively to the different sources of information,
- plays out its capacities inside an adequate time,
- can be introduced and run in its expected platforms, and
- Accomplishes the general outcome of its partners as expected.

As the quantity of potential tests for even basic programming parts is basically limitless, all product testing utilizes some methodology to choose tests that are achievable in the accessible time and with desired assets. Hence, programming testing ordinarily (yet not only) endeavours to execute a program or application with the expectation of discovering programming bugs (mistakes or different imperfections). The process of testing is an iterative procedure as when one bug is fixed, it can light up other, more profound bugs, or can even make new ones.

5.3.1 TYPES OF TESTING

1. UNIT TESTING:

Unit testing is the basic type of testing. This testing is performed at fundamental level. Every unit of the software is tested to check it's functioning. It consists of the test experiments to ensure the appropriate functioning of the internal parts of the program and also to assure that program inputs produce considerable results. It is the process of verifying all the individual units of the program in the whole system. Testing of each unit is done before integrating into the application. Unit testing is done for each segment to assure its performance. These are used to test framework designs, application functionalities and business procedures. These tests are done to certify that business plan works according to the specified details and meets the requirements. It guarantees the outcome matches with the expected outcome provided the input is valid.

2. INTEGRATION TESTING:

Integration testing is done on the integrated programming segments which have already undergone unit testing. It is done to guarantee that the individual segments can perform well together as a program. It ensures the compatibility of the individual units with each other. The faults in incorporation of the individual units are discovered in the process. Even though all the individual units are tested before integration, they are tested wholly as a system to ensure the blend is appropriate. Integration testing is done at both component level and 24 system level. Component integration testing is done to exhibit the faults in blend of components, the interaction between the interfaces and components. System integration testing is done to discover the faults in the interaction between the systems incorporated. It can be performed by the developer or an individual. Test outcomes: All the experiments referenced above passed effectively. No deformities experienced.

3. FUNCTIONAL TESTING:

Functional testing is trying the product to ensure the proper functionality of the functions of the application according to the business requirements and prerequisites. Functional testing is focused on the following items.

- Valid Input: identified classes of valid input must be accepted.
- Invalid Input: identified classes of invalid input must be rejected.
- Functions: identified functions must be exercised.
- Output: identified classes of application outputs must be exercised
- Systems/Procedures: interfacing systems or procedures must be invoked.

Composition of functional test cases is done based on the requirements, key concepts and special cases.

4. TEST STRATEGIES IN MACHINE LEARNING

1. **Training testing split**: The train-test split is a technique for evaluating the performance of a machine learning algorithm. It can be used for classification or regression problems and can be used for any supervised learning algorithm. The procedure involves taking a dataset and dividing it into two subsets.

- Train Dataset: Used to fit the machine learning model.
- Test Dataset: Used to evaluate the fit machine learning model.

2. **K Fold validation test**: K-Fold cross-validation is a statistical method used to estimate the skill of machine learning models. . That k-fold cross validation is a procedure used to estimate the skill of the model on new data. There are common tactics that you can use to select the value of k for your dataset.

3. **K Fold stratified test:** Stratified kfold cross validation is an extension of regular kfold cross validation but specifically for classification problems where rather than the splits being completely random, the ratio between the target classes is the same in each fold as it is in the full dataset.

4. **Testing on balanced dataset:** Testing on balanced dataset is the one that contains an equal or almost equal number of samples from the positive and negative classes.

5. Testing on unbalanced dataset: Testing on unbalanced data refers to those types of datasets where the target class has an uneven distribution of observations, i.e one class label has a very high number of observations and the other has a very low number of observations.

6. Overfitting: Overfitting is a modeling error in statistics that occurs when a function is too closely aligned to a limited set of data points. Thus, attempting to make the model conform too closely to slightly inaccurate data can infect the model with substantial errors and reduce its predictive power.

- Overfitting is easy to diagnose with the accuracy visualizations you have available. If "Accuracy" (measured against the training set) is very good and "Validation Accuracy" (measured against a validation set) is not as good, then your model is overfitting.

7. Under fitting: Under fitting is a scenario in data science where a data model is unable to capture the relationship between the input and output variables accurately, generating a high error rate on both the training set and unseen data.

5.3.2 TEST CASES

SNO	Test case Id	Input	Expected output	Actual output	Test case pass/fail
1	T1	"I am happy"	Joy	Joy	Pass
2	T2	"I am fed up with this laptop"	Disgust	Disgust	Pass
3	T3	"I cried over my success"	Joy	Sadness	Fail

6. CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION:

The basic emotions joy, surprise, anger, disgust, shame and guilt are detected from the data using the ISEAR dataset for training. The work carried out by us proves that lexicon based approach and learning based method are not so efficient in building a model to detect emotions. Each method has it's own disadvantages. Detection of emotions using unsupervised machine learning approach proved to give more accuracy and less errors. It gave 78% accuracy in detecting the emotion of sentences given. It is due to the semantic similarity used for detection. It does not need labelled data, So it is efficient to build a model suitable for all types of datasets.

6.2 FUTURE SCOPE:

It can further be improved by working on different datasets to improve the efficiency of the approach. An advancement to our work includes detecting more number of emotions from the text, not just the basic emotions. Secondary category of emotions can be detected in further enhancement of the work.

REFERENCE

[1] V V Ramalingam1, A Pandian2, Abhijeet Jaiswal3 and NikharBhatia4"Emotion detection from Text" National Conference on Mathematical Te Publishing012027 doi :10.1088/1742-6596/1000/1/012027

[2] Razek, M.A. and Frasson, C. (2017) "Text-Based Intelligent Learning Emotion System of Intelligent Learning Systems and Applications ,9, 17-2-26

[3] Ameeta Agarwal, AijunAn "Unsupervised Emotion detection from text using semantic and syntactic relations" Proceedings of theIEEE/WIC/ACM International joint Conferences on Web Intelligence and Intelligent Agent Technology -Volume 01.

[4] SimpalKawade, Prof. K. C. Waghmare "A Survey on Identification of Emotion from Text Corpus"International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, Issue 3, March 2017

[5] Armin Seyeditabari, NargesTabari, WlodekZadrozny "Emotion Detection in Text: a Review", arXiv:2004.04225 [cs.CL]

[6] Lea Canales, Patricio Martínez-Barco "Emotion Detection from text: A Survey", Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC) DOI: 10.3115/v1/W14-690

[7] Shiv Naresh Shivhare1 and Prof.Saritha Khethawat2 "Emotion detection fom text", DataMining and Knowledge Management Process (DKMP2012)DOI: 10.5121/csit.2012.2237May 201

[8] Martin D. Sykora, Thomas W. Jackson, Ann O'Brien and Suzanne Elayan "Emotive Ontology: Extracting Fine-Grained Emotions From Terse,Informal Messages",International Journal on Computer Science and Information Systems Vol. 8, No. 2, pp.106-118

[9] Douijiyasmina, MousannifHajar, Al Moatassime Hassan "Using YouTube comments for text-based emotion recognition", International Conference on Ambient Systems, Networks and Technologies, 2016

[10] M.Naveen Kumar, R.Suresh "Emotion Detection Using Lexical Chains", International Journal of Computer Applications (0975 – 8887) Volume 57– No.4, November 2012.

[11] Francisca Adoma Acheampong , Henry Nunoo-Mensah,Wenyu Chen"Comparative Analyses of BERT, RoBERTa, DistilBERT, and XLNet for Text-based Emotion Recognition"