

Emotion Detection from Text Using NLP With Prediction Probability

Submitted in partial fulfillment of the required for the award of

Master of Computer Science

By

Ragunathan S

REGISTER NO (42734015)



**DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF SCIENCE AND HUMANITIES**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A++" by NAAC

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 199

APR 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **RAGUNATHAN S (42734015)** who carried out the project entitled "**Emotion Detection From Text Using NLP With Prediction Probability**" under our supervision from Jan 2024 to Apr 2024.

S. Rajeshwary
Internal Guide

Ms. RAJESHWARY.S M.Sc., Assistant Professor.,

Rekha Chakravarthi
HEAD OF THE DEPARTMENT

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.,

Submitted for Viva voice Examination held on

20/04/2024

Internal Examiner

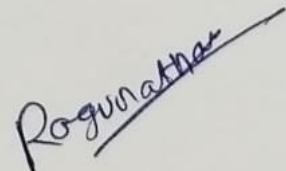
External Examiner

DECLARATION

I am, RAGUNATHAN S (42734015) hereby declare that the Project Report entitled “**Emotion Detection From Text Using NLP With Prediction Probability**” done by me under the guidance of Ms. S.Rajeshwary M.Sc., Assistant Professor, Department of Computer Science at SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, Jeppiar Nagar, Rajiv Gandhi Salai, Chennai-600119 is submitted in partial fulfillment of the requirements for the award of Master of Science degree in Computer Science.

DATE: 20/04/2024

PLACE: CHENNAI

A handwritten signature in blue ink, reading 'Ragunathan S', with a horizontal line drawn through it.

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. REKHA CHAKRAVARTHI M.E., Ph.D., Dean of Science and Humanities** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide **Ms. S.Rajeshwary M.Sc., Assistant Professor.,** Department of Computer Science for their valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff member of the Department of Computer Science who were helpful in many ways for the completion of the project.

RAGUNATHAN S

ABSTRACT

Emotional recognition in texts with prediction probability refers to the interdisciplinary field that leverages advanced natural language processing (NLP) techniques to discern and understand the emotional content embedded in written communication. This innovative approach combines the power of sentiment analysis, machine learning, and probabilistic Model to not only identify the emotional nuances within textual data but also predict the likelihood or probability of specific emotional states. The integration of emotional recognition with prediction probability represents a cutting-edge approach to understanding and interpreting emotions in textual data. Whether applied in business contexts for customer feedback analysis or in healthcare settings for mental health monitoring, this technology has the potential to enhance decision-making processes and improve our understanding of the intricate tapestry of human emotions expressed through written communication.

CHAPTER	TITLE	PAGE NO
	ABSTRACT	I
1.	INTRODUCTION	1
1.1	Overview of project	2
1.2	Aim & Objectives	1
1.3	Scope of Project	2
2.	LITERATURE SURVEY	3
3.	REQUIREMENTS, ANALYSIS, AND DESIGN	4
3.1	Text Based Emotion Detection Model	4
3.2	Emotion Model For Emotion Detection	5
3.3	Text Pre Processing Methods	5
3.4	Feature Extraction Analysis	6
3.5	Feature Extraction Methods	7
3.6	Approaches For Emotion Detection	9
4.	Real World Implementation & Usage	12
4.1	Conversational Agent Bots	12
4.2	Social Media	13
4.3	Review Systems	14
4.4	Product Improvement Analysis	15
5.	Future Directions in Text-Based Emotion Detection	16

5.1	Data Augmentation	16
5.2	Transfer Learning	17
5.3	Explainable Artificial Intelligence	18
5.4	Auto Encoders	19
5.5	Adversarial Machine Learning	19
6.	Flowcharts For Emotion Detection from Word NLP	20
6.1	Flowchart	20
6.2	Dataset Training Process Diagram	22
6.3	Dataset Training IDE Diagram	22
6.4	Dataflow Diagram	23
6.5	Entity Relationship Diagram	23
7.	Existing System & Proposed System	24
7.1	Existing System	24
7.2	Proposed System	25
8.	Modules Implementation	26
	Modules Implementation & Overview	27
9.	System Requirements	28
	System Requirements	28
	Screenshots & Output	29
10.	Conclusion & Future Enhancement	33
	Conclusion & Future Enhancement	33
	Reference	34
	Appendix – Source Code	35

List of Figures

6.1	FLOW DIAGRAM	20
6.2	DATASET TRAINING SET PROCESS DIAGRAM	21
6.3	DATASET TRAINING IDE DIAGRAM	22
6.4	DATAFLOW DIAGRAM	22
6.5	ENTITY RELATIONSHIP DIAGRAM	23

CHAPTER 1: INTRODUCTION

1.1 Overview of Project:

Emotion detection from text using NLP with Prediction Probability is about teaching computers to understand feelings in written text. First, we gather a bunch of examples with labels like happy, sad, or angry. Then, we clean up the text and turn it into numbers computers can understand. Next, we train a computer model using this data to learn the patterns between the words and the emotions. After training, we check how well the model can guess emotions on new text it hasn't seen before. The model not only predicts emotions but also gives a score showing how confident it is in its prediction. If the model isn't accurate enough, we tweak it by adjusting settings and trying different techniques until it gets better. This technology has many uses, like understanding customer feedback, monitoring social media, or even helping with Product Improvement Strategy.

1.2 Aim & Objectives:

Aim:

The aim of the emotion detection from text using NLP with Prediction Probability project is to develop a system capable of accurately identifying and categorizing emotions expressed in written text using natural language processing techniques.

Objectives:

Developing an emotion recognition system involves several steps. First, gather a dataset with labeled text samples showing different emotions. Clean and format the data for NLP tasks. Extract meaningful features from the text using techniques like word embeddings. Train a machine learning model to recognize patterns between text features and emotions, and evaluate its performance using metrics like accuracy.

Enhance the system by providing probability estimates for each emotion class, offering nuanced insights. Fine-tune the model through optimization techniques like hyperparameter tuning. Finally, deploy the trained model as a web application for practical use, such as sentiment analysis or customer feedback analysis. This process ensures the model accurately interprets emotions in text, providing valuable insights for various applications.

1.3 Scope of The Project:

The project focuses on using NLP to detect emotions from text, with the added feature of prediction probability. This involves collecting and preprocessing text data, selecting and training appropriate models, and evaluating their performance. Future scopes include refining models for better accuracy, expanding datasets for diverse contexts, integrating real-time analysis for instant feedback, and exploring applications in mental health monitoring, personalized marketing, and chatbot interactions. Additionally, advancements in NLP techniques like transformer models and fine-tuning strategies will enhance emotion detection capabilities. Collaboration with psychologists and domain experts could further improve model interpretability and effectiveness in practical scenarios. Emotion detection from text using NLP with prediction probability offers promising avenues for understanding human sentiment across various domains and improving user experiences in diverse applications.

CHAPTER 2: LITERATURE REVIEW

AUTHOR	YEAR	LANGUAGES	RESULTS
Ortony, Andrew, Gerald L	1990	HTML & JAVASCRIPT	Cognitive Structure of Emotions
Donald Kirson, Philip	1987	HTML, CSS, JS, ECMA -3	Emotion Knowledge of Text Based Approach
Hearst & Marti	1992	JavaScript & Python	Automatic Acquisition of Hyponyms from Large Text Corpora
Brill Eric	1994	Python 1.2.4 & Language Modules	Some Advances in Transformation Based Part of Speech Tagging.
Ekman, Paul	1992	Python 2.4.5	Argument For Basic Emotion.

CHAPTER 3: TEXT BASED EMOTION DETECTION MODEL

3.1 Text Based Emotion Detection Model:

In the first step, the data is acquired from different text sources. Then text preprocessing is performed to clean the data. Mainly, the text from social media, product/customer reviews consist of slang words, emojis, short text, incomplete words, etc., which requires preprocessing. So preprocessing steps include tokenization, lowercasing, removing stop words and digits, removing punctuation marks, removing white spaces, removing hashtags (#), removing URLs, removing emoticons and emojis. Then applying stemming and lemmatization to get words into their root form. The next step is feature analysis which includes feature extraction and feature selection. In Feature extraction, essential features are extracted. If the feature set is large, the feature selection can be made to choose the most informative. The final step is classification and assigning data to one of the possible emotion classes. Classification techniques are used to recognize emotions. Commonly known classification techniques are machine learning, deep learning. In the literature, we can find, for example, Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), Naïve Bayes, many others. Deep learning does not require specific features. It learns from data itself. However, we need a relatively large data set to train generalization abilities. If the available data set is too small, its size can be expanded artificially by data reinforcement techniques.

3.2 Emotion Models for Emotion Detection:

Studies of emotions date back to Greek mythology. Ancient Greek literature exhibits a wealth of emotions. According to Greek literature, emotions were organized into four basic categories: metus (fear), aegritudo (pain), libido (lust), and Laetitia (pleasure). After that, in the 19th century, Darwin's emotion theory stated emotions and expressions of emotions in humans and animals. In psychology, emotions are categorized into basic emotions and complex emotions (that is, emotions that are hard to classify under a single term, such as guilt, pride, shame, etc.). However, there is no universally accepted model of emotions. There are two prevalent theories in this field: the discrete emotion model and the dimensional model. And the third model has based on appraisal theory: the componential model.

1) Discrete Emotion Model – It is also termed a categorical emotion model. As stated by, discrete emotion theory, some emotions are distinguishable based on neural, physiological, behavioral, and expressive features regardless of culture. The discrete emotion model states that emotions can be placed into basic distinct classes or categories. Different emotions correspond to different neurological subsystems in the brain. Ekman's model (Ekman Paul et al. 1999) categorizes emotions into six basic categories. These basic emotions are happiness, sadness, anger, disgust, surprise, and fear. Plutchik's model (Plutchik 1980) states eight

emotions in opposite pairs as joy vs. sadness, trust vs. disgust, Anger vs. fear, and surprise vs. anticipation.

2) Dimensional Emotion Model - This model assumes, emotions are dependent and a relation exists between them. The dimensional model is based on the hypothesis that all emotions result from a common and interconnected neurophysiological system. Dimensional emotion models define a few dimensions with some parameters and specify emotions according to those dimensions. Two or three dimensions are used in most dimensional emotion models—‘valence’ (indicates the positivity or negativity of an emotion), ‘arousal’ (indicates the excitement level of emotion), and ‘dominance’ (indicates the level of control over an emotion). (Russell et al. 1980) presents a circular two-dimensional model prominent in dimensional emotions representation called the circumplex of effect. (Plutchik 1980) presents a 2-dimensional wheel of emotions showing valence on the vertical axis and Arousal on the horizontal axis.

3) Componential Emotion Model - This approach is an expansion of the dimensional approach. It is built on the appraisal concept (Scherer et al. 2005). According to this paradigm, emotion is derived through an analysis of circumstances if a person can be met with it. The outcome is determined by a person's background, objectives, and possibilities for action.

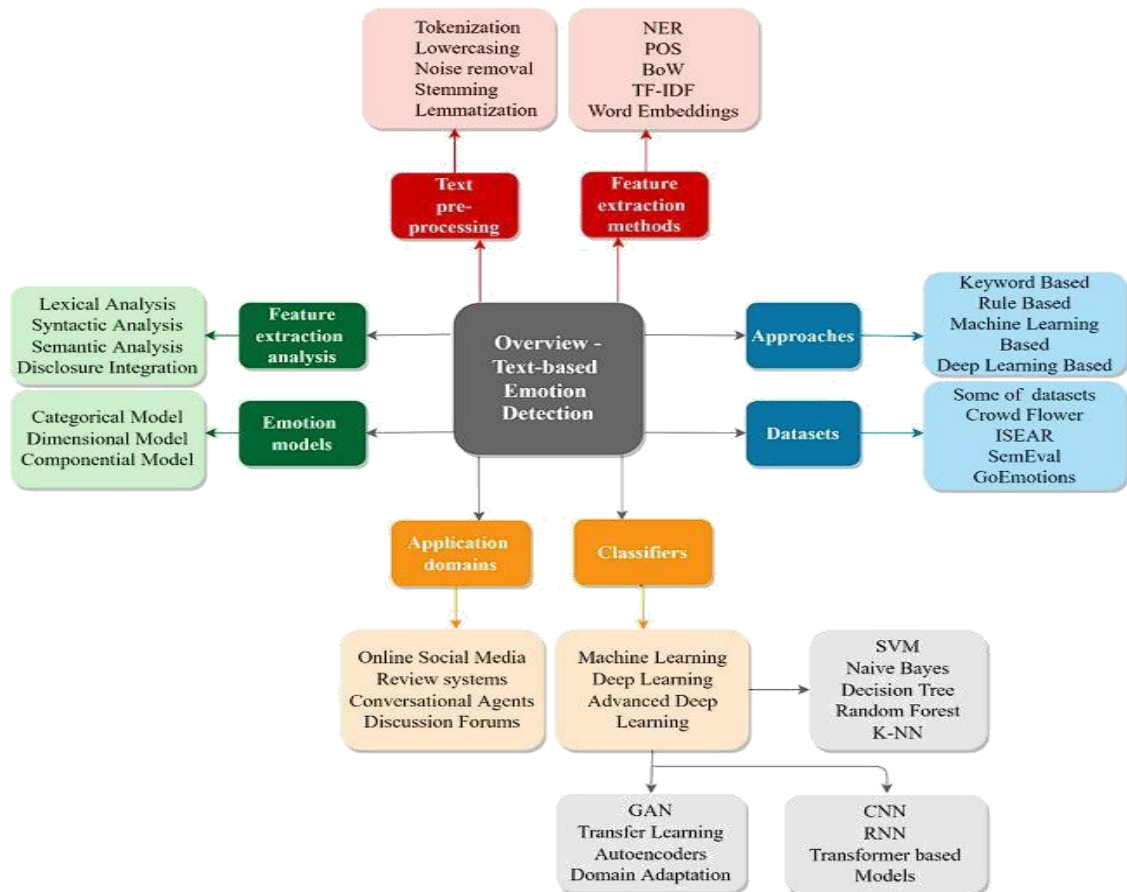
3.3 Text Pre-Processing Methods:

The initial step in text-based emotion detection prepares the data to the form appropriate for any machine or deep learning system. Many deep learning and machine learning algorithms require numerical data to perform any classification or regression task. Therefore, text preprocessing and feature extraction are vital steps for text classification problems. This segment introduces methods of text cleaning, tokenization, normalization, and allowing for informative featurization. Mainly, the text from social media, customer/ product reviews consist of short text, emojis, slang words, incomplete words, etc., which make preprocessing a necessity. Text preprocessing has the following steps:

1) Data cleaning – It includes lowercasing, punctuation, and digits removal, stop words removal, hashtag & HTML tag removal, expanding contractions, etc.

2) Tokenization/Segmentation – Splitting text/strings into tokens representing words.

3) Normalization - Stemming and lemmatization include truncating a word to its root form.



3.4 Feature Extraction Analysis:

Commonly, documents and texts are unstructured data. Therefore, this section gives a brief theoretical idea about feature extraction elements before diving into feature extraction methods. Following are some elements that need to be considered while extracting features in the text. It shows different feature extraction analyses used in text-based emotion detection.

1) Lexical Analysis – In linguistic analysis, the whole text is divided into paragraphs, sentences, and words. It helps to identify and analyze the word structures. Tokenization is one of the examples of linguistic analysis.

2) Syntactic Analysis – It involves analyzing the grammar of words in a sentence and showing the relationship among the words. Parts-of-Speech tagging (POS), parsing are some examples of syntactic analysis.

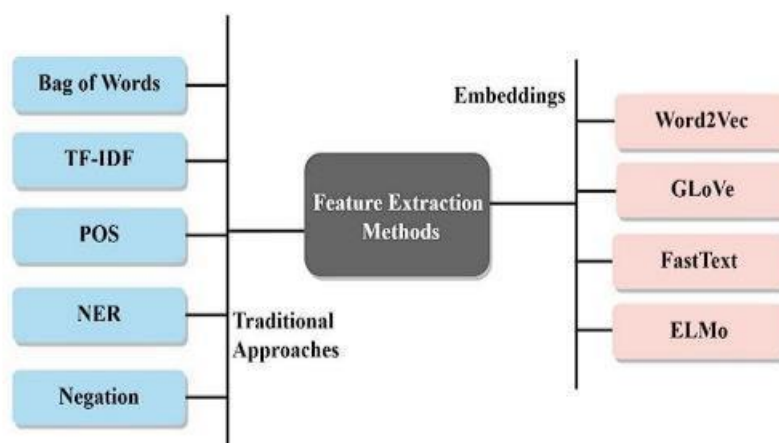
3) Semantic Analysis – Semantic analysis determines the actual meaning of words and evaluates the text's significance. Named Entity Recognition (NER) is an example of semantic analysis.

4) Disclosure Integration – Disclosure integration gives a sense of the context. It involves building the relationship and meaningfulness between two consecutive sentences or phrases. Dependency parsing is one of the examples of disclosure integration.

5) Pragmatic Analysis – The general interpretation of language is the focus of pragmatic analysis. It is concerned with leveraging real-world knowledge to derive meaningful interpretations of human language.

3.5 Feature Extraction Methods:

The unstructured text must usually be converted to structured language using a mathematical model or algorithm, as shown in figure 7. To begin, the data must be cleansed to remove any irrelevant characters or words. Formal feature extraction approaches can be used once the data has been cleansed. Feature extraction algorithms are required to convert the text into a matrix (or vector) of features. This transformation task is generally called feature extraction of data. Feature extraction techniques are shown in figure 8. Text feature extraction techniques have two approaches: the traditional approach based on syntactic word representations and the modern approach based on feature learning techniques.



1) Traditional Approach (Syntactic Word Representations) – It includes the following methods –

a) Bag of Words (BoW) — This method is also known as the n-gram method. BOW counts the frequency of words in the text and generates a sparse vector that represents present words as one and missing words as 0 in the text. Machine learning models use these vectors as input. N-grams are groups of words that appear in close proximity to one another and are merged into a single vector of features. Common representations of N-grams are 1-gram, 2-gram, 3-gram.

b) Term Frequency-Inverse Document Frequency (TF-IDF) – The TF-IDF metric indicates how significant or relevant a term is in a document. It identifies words or sets that frequently appear in a single document yet are uncommon throughout the text corpus. TF-IDF is defined as follows: $TF-IDF = TF * IDF$ $TF = \text{Term Frequency} = \text{no. of term in a document} / \text{total terms in the document}$. $IDF = \text{Inverse Document Frequency} = \text{Total No. of documents} / \text{no. of documents has term}$. So, TF gives the measure of how frequently the term occurs. And IDF gives the measure of how important the term is.

c) Part of Speech Tagging (PoS) – The POS is the word category that showcases its role and helps the NLP work. It includes categories like nouns, verbs, adjectives, and adverbs. **d) Named Entity Recognition (NER)** – In the disambiguation, the entities are identified in a sentence, such as the famous person, brand, etc. In the named entity recognition, the entity is identified and categorized as per the date, organization, person, time, location, etc

2) Modern Approach (Feature Learning Techniques) – Embeddings - Word embeddings are techniques in which features are learned, and each word from the dictionary is represented to N-dimensional integer. These feature vectors are based on the context of the text dataset. To convert words into intelligible input, many word embedding methods have been developed. Word2Vec, GloVe, FastText, and ELMo are some common deep learning embedding approaches that have been successfully applied.

- **Word2Vec** – (T. Mikolov et al. 2013) introduced the "word to vector" interpretation. It is an upgraded architecture of word embedding. This approach utilizes continuous bag-of-words, skip-gram, and lightweight neural networks consisting of two hidden layers to generate a feature vector for every word. It gives you immediate access to the vectors that represent words. A simple continuous bag of words (CBOW) model looks for words that might appear near one other, whereas skip-gram looks for words that might appear near each other. This approach is a strong method for detecting correlations in a text dataset as well as word resemblance.

- **Glove** – GloVe (Global Vectors for Word Representation) by (Pennington et al. 2014) is another technique for creating word embeddings. It captures a global word-to-word conjunction matrix from a dataset to construct word embeddings. The idea is to establish a context for words i and j in terms of whether or not they occur near N -words apart. The ratio of the co-occurrence probability of two words is encoded in the encoding vector, which is known as the count-based technique. However, this method does not employ the entire corpus. It collects global statistics and improves word representation learning on tasks including word comparison, word similarity, and named entity recognition.

Fast Text - Facebook's AI research team created the FastText library. It is capable of learning and calculating word representations as well as sentence classification. It is a novel technique that solves the morphology issue by allocating a different vector to each word. As a result, each word is represented by an n-gram, which is a collection of characters.

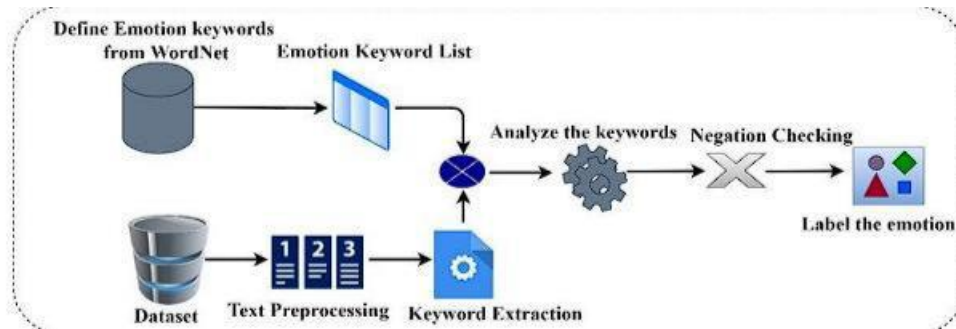
- **ELMO** – Embeddings from Language Models (Peters et al. 2018) is a type of contextual embedding that looks at the words around. It uses a bidirectional LSTM algorithm, which means that the words before and after it are used in both directions. This character-based language model is able to envision the previous or next word, as well as handle words that aren't in the dictionary. It also handles semantical and grammatical characteristics of words and how this differs around language contexts like to develop lexical ambiguity. It advances modern techniques in a broad range of demanding NLP tasks, involving textual context question answering, and sentiment analysis.

3.6 Approaches for Emotion Detection:

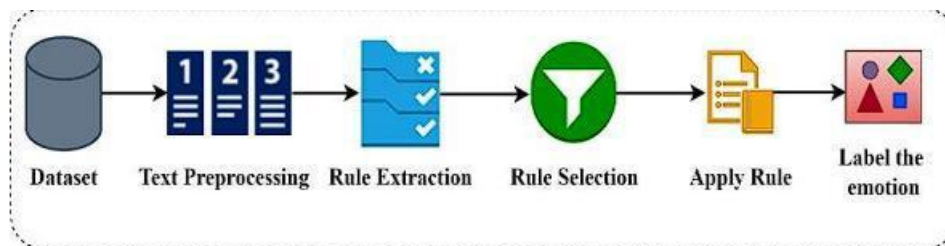
The most common approaches for identifying text-based emotions are keyword-based, rule-based, machine learning-based, deep learning-based.

- 1) **Keyword-based approach** – This technique focused on locating keyword occurrences in a given text and comparing them to the annotations registered in the dataset. Figure 9 outlines the method. The initial emotion keyword list is derived from conventional lexical resources WordNet or WordNet-Affect in this method. After that, the dataset is pre-processed. Then, keyword matching is performed between a predefined keyword list and emotion words from the text. After that the power of the emotion keyword is examined. After that, negation is examined to see whether there are any negation cues and what is the scope of, and lastly, the emotion tag calculation is done. In this, we studied (Tao J. 2004, Ma C et al. 2005, Perikos et al. 2013, Shivhare et al. 2015) which are based on a keyword-based approach.
- 2) **Rule-based approach** – This approach utilizes linguistic rules to recognize the emotions from the text. Figure 10 shows a general overview of the approach. The dataset is first subjected to text preparation. It involves data cleaning, tokenization, POS tagging, etc. Then rules for extracting emotions are built using a statistic, linguistic concepts. Then, probabilistic affinity is stored with each word. Later, the best of the rules has been chosen for the dataset to detect emotion labels.

Keyword Based Approach Diagram:



Rule -Based Approach Diagram:



Machine learning-based Approach — Machine learning-based methods allow systems to learn and improve on their own as a result of their experiences. Text is classified into several emotion classes using machine learning algorithms. There are two types of machine learning algorithms: supervised and unsupervised. Supervised machine learning algorithms are applied in the majority of the studies reviewed. Text preprocessing, such as tokenization, POS tagging, and lemmatization, is usually the first step in this technique. Then the text's useful features are extracted, and only the features having the highest information yield are chosen. The algorithm is then trained using the specified characteristics and emotion labels. finally, the system is utilized to predict emotions from unobserved data using a trained system.

CHAPTER 4: REAL WORLD IMPLEMENTATION & USAGE

4.1 Conversational Agent Bots:

Conversational agents are referred to by various terms, including chatbots, dialogue systems, intelligent virtual assistants, intelligent virtual agents, smart bots, interactive agents, digital assistants, and relational agents. A chatbot is a human-computer interaction model with an artificial intelligence program (Eleni et al. 2020). A chatbot is described as a "computer software meant to simulate interaction with human users, particularly over the Internet, by using spoken or written natural language as a primary means of communication.". Conversational agents, often known as chatbots, mimic human interaction and are utilized in various areas such as business, education, healthcare, entertainment, and many more. The primary goals of chatbots are entertainment, social contact, and novelty interaction, with a strong emphasis on productivity. In chatbots, there has been a tremendous requirement to simulate human-like characteristics and Behavior. Chatbots are usually depicted, built, and developed as a flow of communication between multiple components as Natural Language Understanding (NLU), Dialogue Management (DM), and Natural Language Generation (NLG). The NLU generates a semantic representation of a user text, such as intent or purpose, to extract the "meaning" of the text. The NLU's main role is parsing, which involves taking a collection of words, generating a linguistic formation for the words, and storing in the knowledge repository. The second component is dialogue management, which formalizes communication structure and performs contextual understanding by assessing word disambiguation and identifying phrases connected by references and supports domain knowledge supervision. It can match expressions (emotion) using lexicon and handling synonyms, negation, and intensity based on the presence of terms (emotional). The third component is NLG or response module. Commonly, Chatbot responses are restricted to providing a direct response to the participant's question. NLG gets a communication response from the dialogue manager and creates a textual depiction that corresponds.

Conversational agents are increasingly being used in industrial applications, from simple conversation interfaces to sophisticated human aides. Industries are infusing human-like qualities, features, and behaviors into conversational agents to increase productivity. A chatbot implemented by (Thomas 2016) responds to the users with AIML, and LSA uses a dataset of Frequently Asked Questions. (Xue et al. 2018) designed a neural-based conversational agent that uses Bi-LSTM with attention to reply to repetitive questions. (Sun and Zhang 2018) developed a personalized recommendation system.

4.2 Social Media:

The majority of the world's population uses popular social networking sites or applications such as Facebook, Instagram, Reddit, Twitter, YouTube, and WhatsApp. Furthermore, microblogging services such as Instagram, Reddit, and Twitter are immensely popular. Consequently, we may state that postings that include text, images, audio, and video, online media has developed as a strong medium for expressing, communicating, and sharing people's thoughts, opinions, perspectives, views, and including native to worldwide issues and topics. So, it becomes important to analyse the emotions behind people's opinions, thoughts, views, and perspectives. This study will be useful in finding the impact of OSM on different online trends(positive/negative/influential/biased), social be-haviour, and individual social conduct. Feelings, emotions, character traits, and their impact on societal trends can all be found using TBED. There are various online social media application areas such as business intelligence, be-haviour analysis, epidemics, event detection, crime detection, sentiment, opinion, and emotion analysis, recommendations.

They studied the ability of weather to affect the spread of COVID-19 pandemics. Their work was based on Twitter data and used Gradient Descent Support Vector Machine (GDSVM), random forest, naïve Bayes, and TF-IDF machine learning algorithms. Topic modeling is used to find out major topics of discussion in the pandemic. Authors in (Sun et al. 2015) designed a social regulation method that incorporates social network information to benefit recommendation systems. The authors considered the social relationship among the users for improving the accuracy of the recommendation system. They used a bi-clustering algorithm to calculate the trusted friends in the network.

Many difficulties must be overcome while identifying emotions via social media. First, the casual writing style of users in social media is a big challenge. Grammatical and spelling mistakes, as well as slang language, sarcasm, and irony, may be found in social media posts. Simultaneously, the use of informal language and short messages has been studied, but only to a limited extent. Second, the expression of human emotions and the texts that convey them are subjective and ambiguous. As a result, correctly inferring and interpreting the author's emotional states is challenging. Third, emotions are multifaceted constructs with unclear confines and varying expressions. As a result, human mapping is important. Finally, affective expression is difficult for automated systems to understand. Because of the wide range of topics discussed on social media, manually creating a complete dataset of labeled data that includes all conceivable emotional circumstances is difficult.

4.3 Review Systems:

BI (Business Intelligence) (Sandra et al. 2021) is a set of methodologies, frameworks, and approaches for transforming data into actionable information that helps businesses run more efficiently. It is, in fact, a set of apps and services that convert data into usable knowledge and experience. The use of the BI paradigm to social media data benefits businesses of various domains. Social media platform data can benefit various business domains. Business-related information gathered from product/customer reviews, tweets, likes, shares, comments of specific and related business data and disseminated through online social media platforms can assist businesses in producing high-quality products and services. Especially in product design and development, early feedback, detection, and conceptual concepts from customer study are crucial. A good user experience with the product, system, and service is gradually recognized as a significant product/service design advantage. In general, user experience/feedback (UX) refers to how a user feels about utilizing a product, system, service, or object (Bai et al. 2019). Customer holding and trustworthiness are critical components of business success. Because there is widespread agreement that customer satisfaction is strongly linked to slow customer destruction, many business strategies focus on analysing and improving customer satisfaction. According to the application area, emotion recognition from product and customer reviews has been researched more than service reviews.

Even while this field's research isn't inextricably connected to the crucial user feedback idea "emotion" (e.g., happiness, joy, fear, surprise, and disgust), user sentiment like a neutral, negative, or positive does refer to feelings, albeit not to the same extent. Therefore, it is needed to present a possible map from sentiments to feelings by further leveraging the language elements in the feedback/ evaluations, such as sorting them based on emotion categories. Emotion detection from review text has mainly been done in movie review systems, hotel review and recommendation systems, E-commerce review systems, agriculture product review systems.

The authors also suggested a technique for constructing a knowledge base of user comments from online consumer reviews. User feedback discovery, which extracts user feedback data from a single review, integrates with similar group data, and user feedback network formalization, which establishes causal linkages among user feedback data groups, are the three stages of their approach. By analysing a large number of online customer data, this study analyses the feasibility of automatically uncovering relevant user feedback data and their correlations for product creation and strategic business planning. Feature extraction using syntactic rules and determining the opinion's polarity according to their emotional power from pros/cons reviews highlight their work.

4.4 Product Improvement Analysis:

Product reviews play a crucial role in shaping consumer decisions, with emotional cues often being key indicators of satisfaction or dissatisfaction. Implementing emotional detection in analysing customer reviews can significantly enhance product review strategies and foster improvements. By harnessing sentiment analysis tools, companies can extract valuable insights from customer feedback, enabling them to refine their products and services. To optimize product review strategies, companies should focus on several key areas. Firstly, they should prioritize sentiment analysis to identify prevailing emotions within reviews, such as joy, frustration, or disappointment. This enables businesses to gauge overall customer sentiment accurately. Next, companies can categorize reviews based on emotional themes, allowing for targeted improvements in areas where customers express dissatisfaction or delight. For instance, if numerous reviews convey frustration with a product's usability, the company can prioritize enhancing user experience through design modifications or clearer instructions.

Furthermore, leveraging emotional detection facilitates personalized responses to reviews, demonstrating attentiveness to customer feedback and fostering stronger connections with consumers. By acknowledging and addressing the emotional aspects of reviews, companies can cultivate loyalty and trust among their customer base. Moreover, emotional detection enables companies to tailor their marketing strategies more effectively. By understanding the emotional drivers behind customer reviews, businesses can create targeted campaigns that resonate with their audience on a deeper level. For instance, if positive reviews frequently mention feelings of joy or excitement, companies can craft advertising messages that evoke similar emotions, thereby enhancing brand affinity and increasing conversion rates. Emotional detection allows companies to anticipate potential issues before they escalate, enabling proactive problem-solving and preventing customer churn. By identifying patterns of negative emotions in reviews, businesses can swiftly address underlying concerns and implement corrective actions, showcasing a commitment to customer satisfaction and loyalty. This proactive approach not only resolves immediate issues but also showcases the company's dedication to continuous improvement, fostering long-term trust and loyalty among customers.

In conclusion, integrating emotional detection into product review strategies empowers businesses to glean actionable insights from customer feedback, driving continuous improvement and enhancing overall customer satisfaction. Additionally, emotional detection can guide product development initiatives by uncovering unmet emotional needs or desires among consumers. By identifying gaps in the market where customers express longing or dissatisfaction, companies can innovate products and services that fulfill these emotional demands, gaining a competitive edge and fostering brand loyalty.

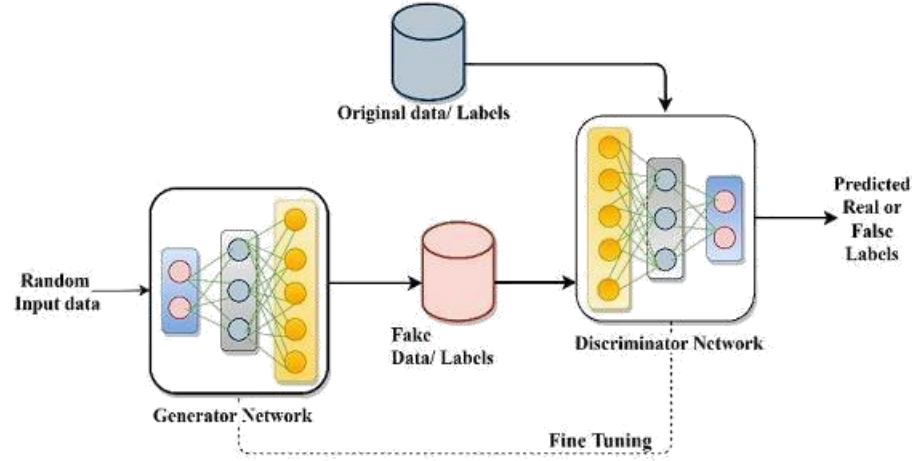
CHAPTER 5: Future Directions in Text-Based Emotion Detection

Techniques or methodologies in TBED have witnessed huge evolutions over the past several years. Combining keyword-based approaches with n-hidden layer-based deep learning techniques, for example. Artificial Intelligence advances in recent years have bolstered text-based emotion detection algorithms. Some of the outstanding difficulties in text-based emotion detection have been addressed by AI-based technologies such as data augmentation (generative adversarial networks), transfer learning, explainable AI, autoencoders, adversarial machine learning, and different learning styles. Figure and table showcase several challenges and their solutions using these methods with references.

5.1 Data Augmentation:

In Natural language processing (NLP), many tasks like text classification to question answering require vast training data to improve the performance of the model. However, acquiring and annotating additional data can be an expensive and time-consuming process in order to generate a large amount of training data. In this scenario, data augmentation techniques can be applied. Data augmentation is a machine learning technique that artificially generates additional synthetic training data through label preserving transformations (Bayer et al. 2021). Though data augmentation techniques have achieved great success in computer vision applications, they are powerful in NLP applications. Data augmentation is achieved by thesaurus, word embeddings, back translation, replacing synonyms, text generation, etc. In (Bayer et al. 2021), the authors proposed a novel text generation method to increase the performance of short and long text classifiers using data augmentation techniques. The authors used GPT-2 355 million parameter model for text generation in the implementation. The authors in (Shi et al. 2020) developed an efficient data augmentation algorithm for labeled sentences called AUG-BERT for text classification. A deep neural network-based text classifier based on data augmentation and semi-supervised learning built using free-text obtained from a survey on sleep-related issues (Shim et al. 2020). One of the techniques trending for data augmentation is GAN, explained in detail as follows –

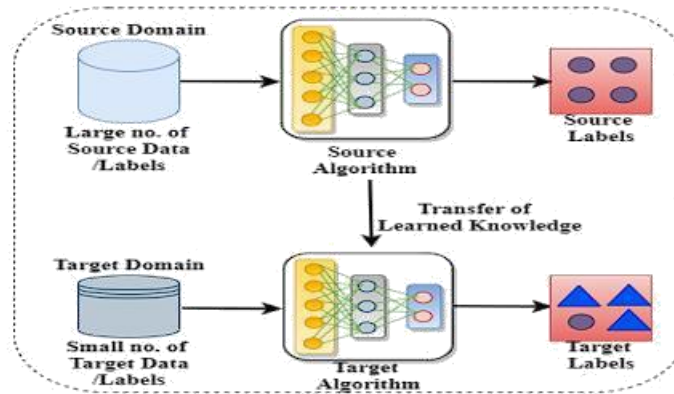
Generative Adversarial Networks (GAN) – The lack of balanced data over all labels in the data is one of the most critical issues in text-based emotion detection. Class imbalance causes over-representation of majority classes, poor model performance, and classification models may disregard or treat small classes as noise. GAN is the most recent technique for dealing with imbalance. Figure 15 shows the architecture of the generative adversarial network. A generative adversarial network (GAN) is a new framework that sets two neural networks against one other (thus the term "adversarial") to create new, synthetic data instances that look like actual data. The discriminator evaluates whether each instance of data it examines corresponds to the actual training dataset or not, while the generator generates new data instances.



5.2 Transfer Learning:

Data collection and labeling are time-consuming and frequently demand expert knowledge. Transfer learning is an excellent technique to handle difficulties like data scarcity and a lack of human labeling. Transfer learning allows you to reduce the amount of labeled data you need and take advantage of a previously trained model to increase performance. The knowledge of the source domain can be transferred to separate but related target domains using transfer learning. A set of data containing adequate data samples, a large number of labels, and relatively high quality is referred to as a source domain. A target domain's data, on the other hand, may contain a small number of samples, limited amounts of data, or no labels at all, as well as being noisy.

Transfer learning strategies try to increase the learning of the target task using information from the source domain when given a source and a target (Feng and Chaspuri 2019). Figure 16 gives a closer look at transfer learning. Transfer learning for emotion recognition via cross-lingual embeddings was proposed by the authors in (Zishan et al. 2020). Their developed a deep transfer learning framework using CNN and Bi-LSTM efficiently transfers the captured relevant information through the shared space of English and Hindi. (Omara et al. 2019, Devamanya et al. 2020, Yaraswini et al. 2021) proposed using a transfer learning approach for the scarcity of data.



5.3 Explainable Artificial Intelligence (XAI):

Deep learning and machine learning models have drawback of not being human interpretable. These models suffer the major problem of lack of transparency. The interpretability of AI systems is the topic that is currently generating interest in the scientific community. The non-linear and nested structure of deep neural networks, which enhances prediction model accuracy, makes it difficult to understand how information flows from a human-understandable input such as an image or a piece of text to a predicted output. Deep neural networks are regarded as “black boxes” in this sense. Explainable AI proposes that models built-in machine learning or deep learning should explain their decision-making process and actions. Very little work has been done in this area. Figure 17 presents the architecture of XAI. The Authors in (Zucco et al. 2018)] highlighted the need for explainable sentiment analysis for medical applications in order to extract human-reliable knowledge related to opinions and emotions from the user generated text information.

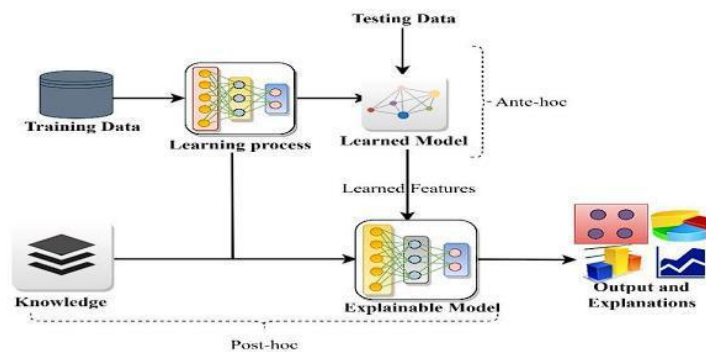


Fig. 17 Explainable Artificial Intelligence (XAI)

5.4 Auto-Encoders:

These are a kind of neural network where feature vectors are transferred into an arbitrary higher or lower dimensional space during the encoding phase, allowing the original feature vector to be rebuilt with minimal reconstruction error in a subsequent decoding phase. Figure 18 shows the representation of autoencoders. Autoencoders can deal with missing data, improve system performance in noisy environments, and be trained unsupervised. Denoising autoencoders, a form of autoencoder, have been employed in domain adaptation, which involves training a model on one Labeled corpus and then applying it to a separate un labeled corpus with a different context. In (Hesam et al. 2017) the authors used stacked denoising autoencoders (SDAs) to perform sentiment recognition on a multitude of domains and languages (Kumari et al. 2021) have used autoencoders to construct a deep learning model for detecting distinct stages of violence (indirect, direct, and no aggression) from posts on social media in a multilingual setting.

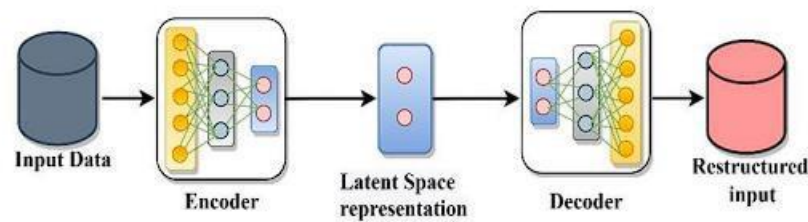


Fig. 18 Autoencoders

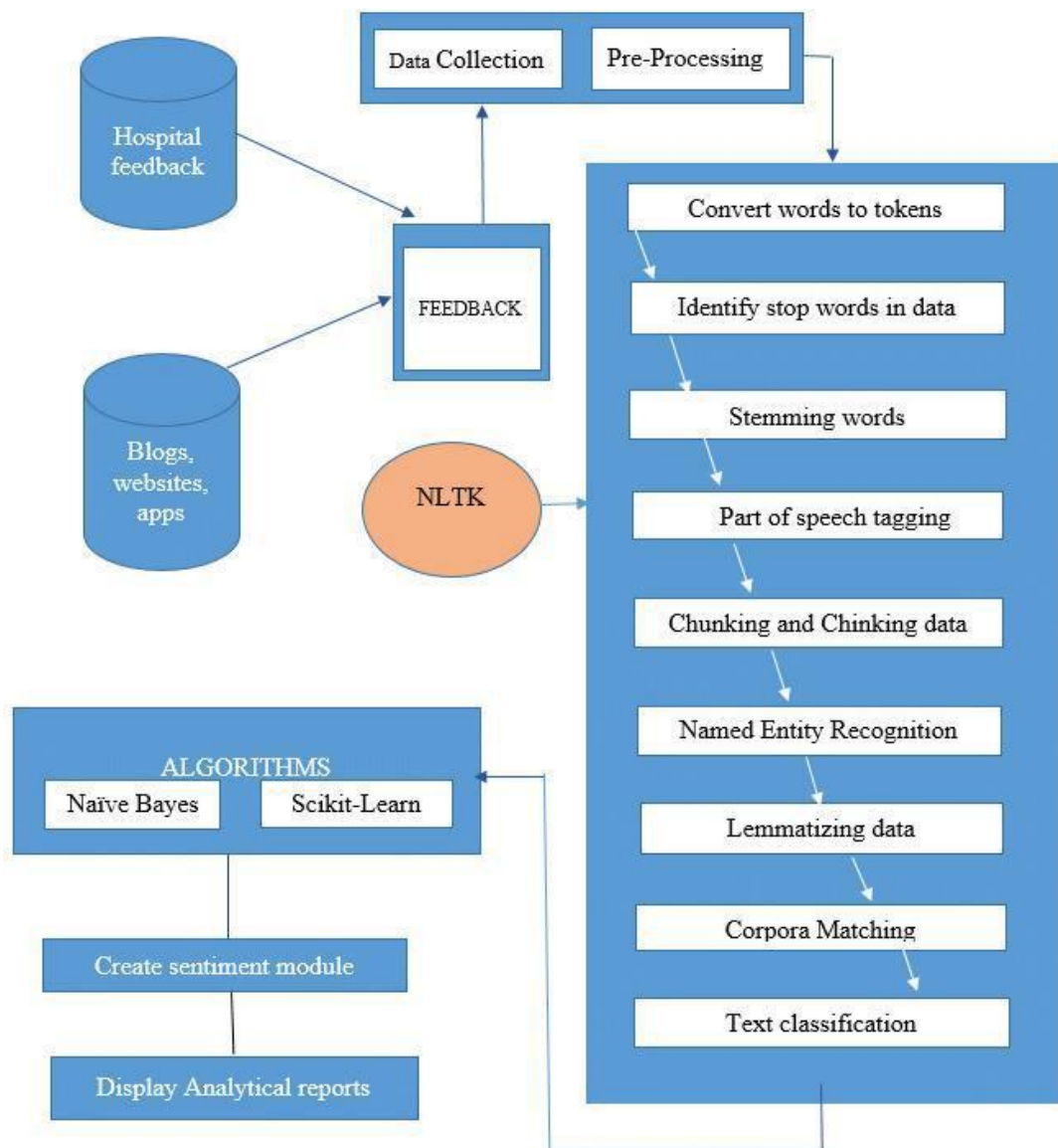
5.5 Adversarial Machine Learning (AML):

In numerous Natural Language Processing (NLP) tasks, many machine learning and deep learning models have had remarkable success. On the other hand, machine learning and deep learning models, are increasingly vulnerable to adversarial examples, according to a growing number of research publications. Adversarial machine learning (AML) is an emerging research area that brings together the best machine learning practices, robust statistics, and computer security (Deldjoo et al. 2021). The act of deploying attacks towards machine learning-based systems and making systems robust to attacks is known as Adversarial Machine Learning (AML). In adversarial machine learning, inputs are modified by creating small perturbations to cause a malfunction in the target model, which outputs incorrect results. While in (Deldjoo et al. 2021), discussed attack/defense strategies for adversarial recommender systems, generative adversarial networks in detail.

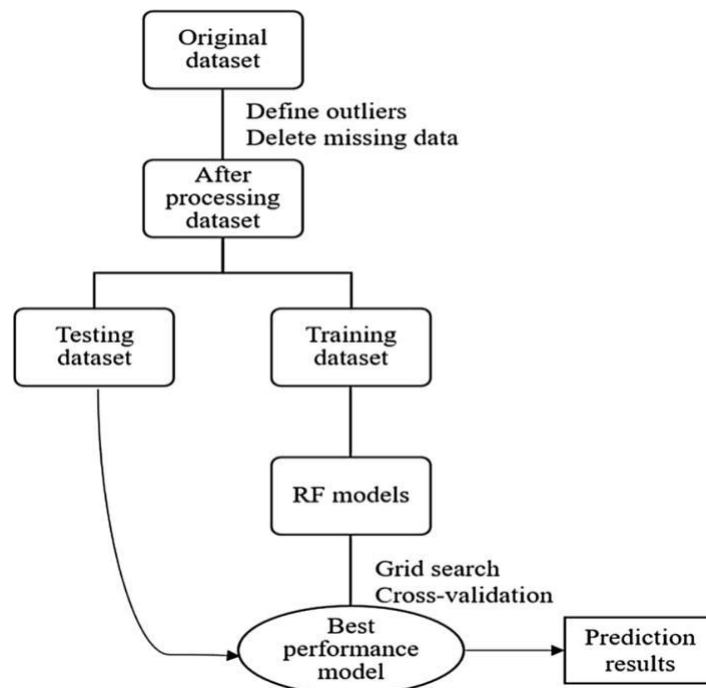
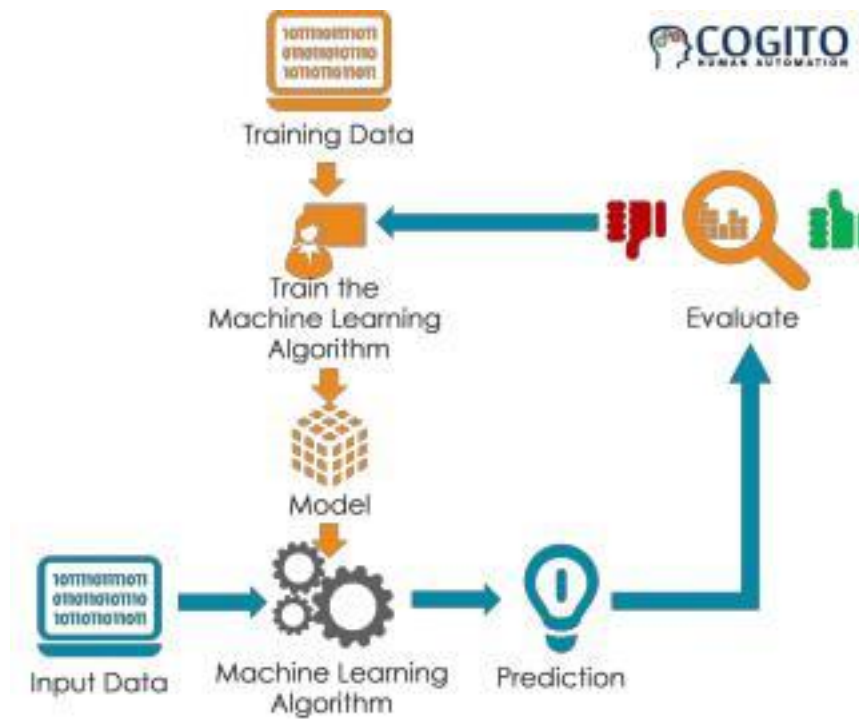
CHAPTER 6: Flowcharts For Emotion Detection from Word NLP

6.1 Flowchart for Emotion Detection from Word NLP:

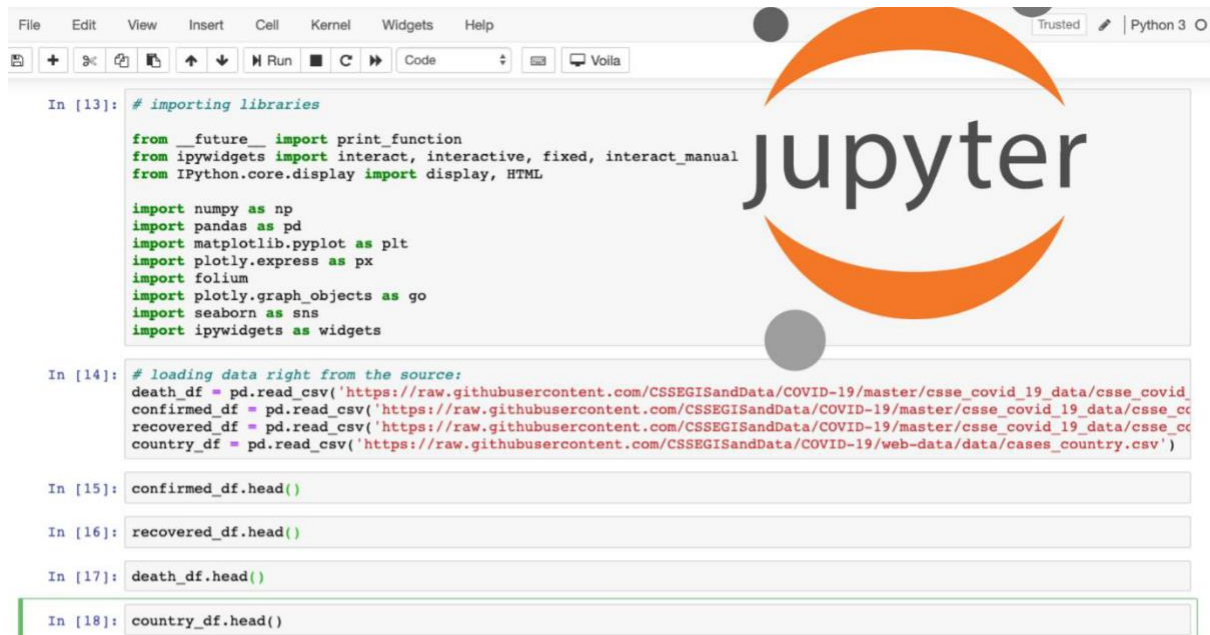
- 1) **Input:** Text as Data / Scrapped Reviews
- 2) **Tokenization:** Break Text into Words
- 3) **Word Embedding:** Convert words into vectors.
- 4) **Emotion Lexicon:** Match word vectors with emotion lexicon.
- 5) **Aggregate Emotions:** Summarize emotions for each text.
- 6) **Threshold Check:** Determine if emotion scores surpass a threshold.
- 7) **Output:** Classify text into emotions (e.g., happy, sad, angry).
- 8) **End.**



6.2) DATASET TRAINING PROCESS DIAGRAM:



6.3) Dataset Training IDE Diagram:



The image shows a Jupyter IDE interface with a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and viewing. The main area displays a Jupyter notebook with the following code:

```
In [13]: # importing libraries

from __future__ import print_function
from ipywidgets import interact, interactive, fixed, interact_manual
from IPython.core.display import display, HTML

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import folium
import plotly.graph_objects as go
import seaborn as sns
import ipywidgets as widgets

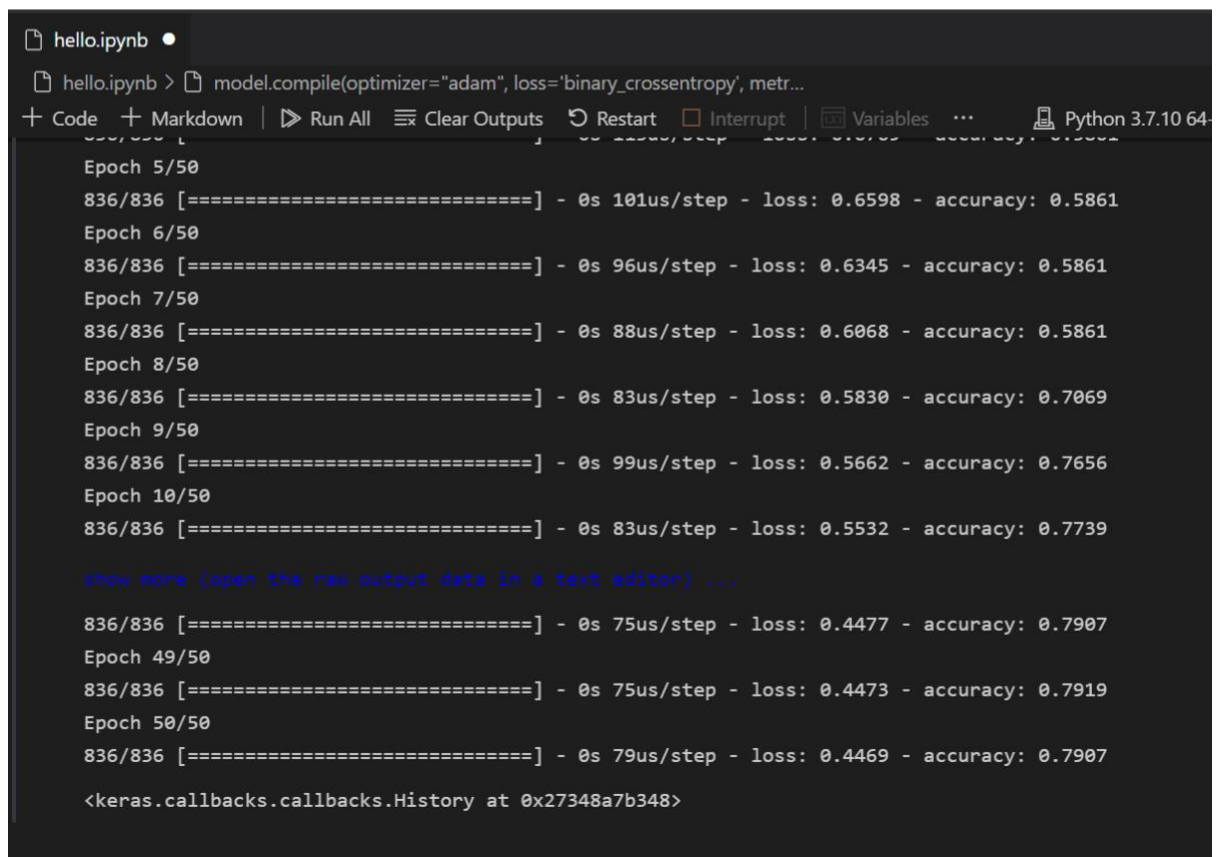
In [14]: # loading data right from the source:
death_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_data/death_df.csv')
confirmed_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_data/confirmed_df.csv')
recovered_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_data/recovered_df.csv')
country_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/web-data/data/cases_country.csv')

In [15]: confirmed_df.head()

In [16]: recovered_df.head()

In [17]: death_df.head()

In [18]: country_df.head()
```



The image shows a Jupyter notebook output for a training process. The output is displayed in a dark-themed interface with a file explorer on the left showing 'hello.ipynb'. The main area shows the following output:

```
hello.ipynb > model.compile(optimizer="adam", loss="binary_crossentropy", metr...
+ Code + Markdown | ▶ Run All | ✖ Clear Outputs | 🔄 Restart | ⏏ Interrupt | 📄 Variables | 📄 Python 3.7.10 64-...

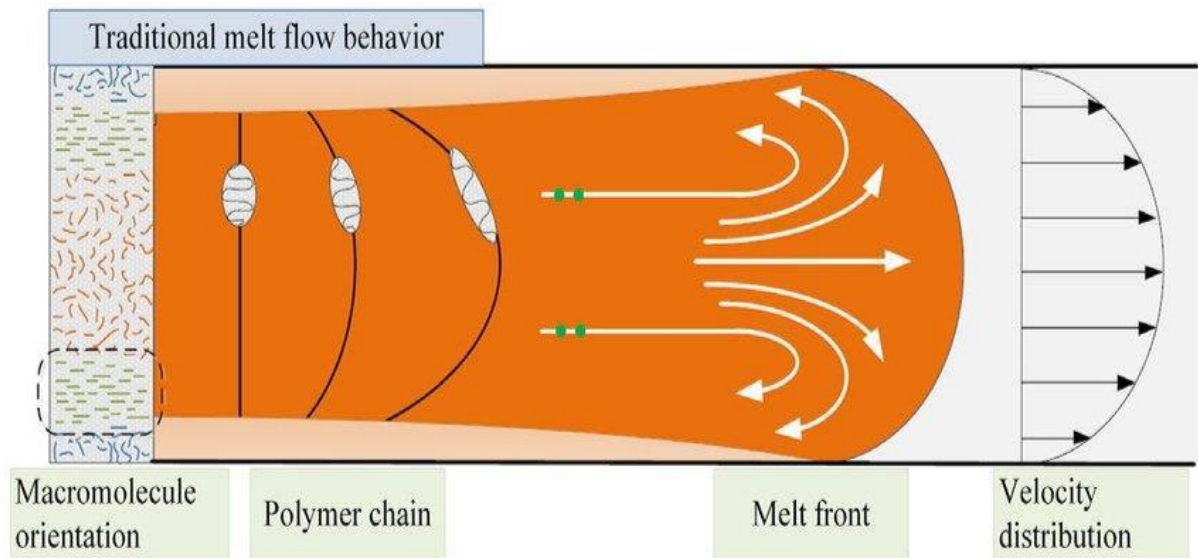
Epoch 5/50
836/836 [=====] - 0s 101us/step - loss: 0.6598 - accuracy: 0.5861
Epoch 6/50
836/836 [=====] - 0s 96us/step - loss: 0.6345 - accuracy: 0.5861
Epoch 7/50
836/836 [=====] - 0s 88us/step - loss: 0.6068 - accuracy: 0.5861
Epoch 8/50
836/836 [=====] - 0s 83us/step - loss: 0.5830 - accuracy: 0.7069
Epoch 9/50
836/836 [=====] - 0s 99us/step - loss: 0.5662 - accuracy: 0.7656
Epoch 10/50
836/836 [=====] - 0s 83us/step - loss: 0.5532 - accuracy: 0.7739

show more (open the raw output data in a text editor) ...

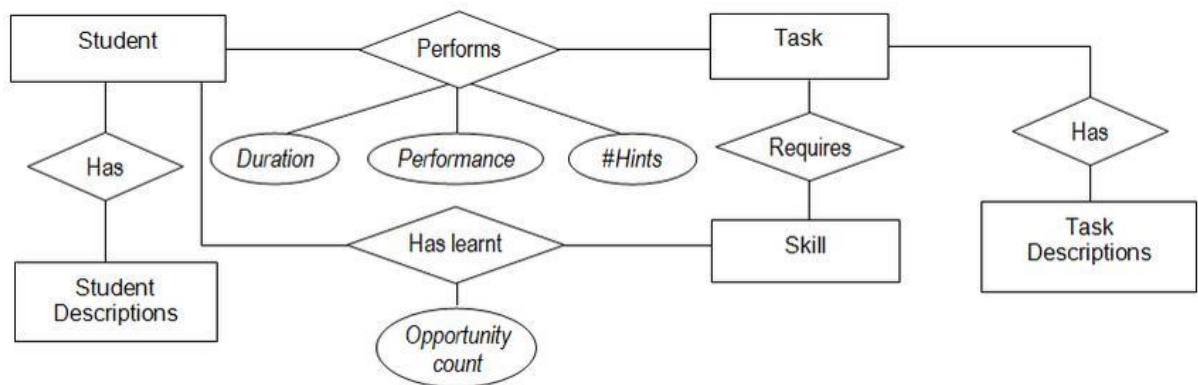
836/836 [=====] - 0s 75us/step - loss: 0.4477 - accuracy: 0.7907
Epoch 49/50
836/836 [=====] - 0s 75us/step - loss: 0.4473 - accuracy: 0.7919
Epoch 50/50
836/836 [=====] - 0s 79us/step - loss: 0.4469 - accuracy: 0.7907

<keras.callbacks.callbacks.History at 0x27348a7b348>
```

6.4) Dataflow Diagram:



6.5) Entity Relationship Diagram:



CHAPTER 7: EXISTING SYSTEM & PROPOSED SYSTEM

7.1 Existing System:

Emotional Detection Analysis involves understanding the feelings expressed in text. First, We collect text as data from sources like social media or reviews. Then, we clean and prepare the text by removing unnecessary elements. Next, we extract features like word frequency. We choose a model, like machine learning or deep learning, to analyze emotions on text. We train the model using labeled data, teaching it to recognize positive, negative, or neutral emotions. After testing its accuracy, we deploy the model to analyze new text. Regular monitoring ensures it stays effective. Overall, Emotional Based Text helps us understand opinions in text quickly and accurately. One existing system for emotion detection from text using Natural Language Processing (NLP) with prediction probability is a sophisticated machine learning framework trained on vast datasets of annotated emotional text. This system employs a combination of NLP techniques and advanced algorithms to analyze textual content and infer underlying emotional states. Initially, the system undergoes a rigorous training phase, where it learns to recognize patterns and associations between specific linguistic features and corresponding emotions. Through this process, the model develops a deep understanding of how different words, phrases, and syntactic structures correlate with various emotional states.

Disadvantages:

The existing emotion detection system using NLP with prediction may have several disadvantages. Firstly, it could suffer from limited accuracy, especially when dealing with complex emotions or subtle nuances in language. Additionally, the system might struggle with out-of-vocabulary words or slang, leading to misinterpretations.

Moreover, the model might be biased towards certain emotions or demographic groups due to the nature of the training data, resulting in unfair predictions. Another challenge is the inability to handle context effectively, as emotions can be influenced by surrounding text or previous conversations.

7.2 Proposed System:

In our Proposed System, we would add the result as a graphical representation after the prediction process takes place and with emotions encapsulated as Emote and it would be easy to understand with bar graph representing the level of emotions. We use the advanced level of Machine learning algorithms and NLP (Natural Language Processing) with reduces in space and time complexity. In Existing only certain level of algorithms were supported and it is rare because only certain dataset supports the algorithms, But With Proper training and recent technologies involved in making algorithm based workflow it would produce result even in graphical forms as well as prediction systems enables the emotions behind every text in a graphical form from user.

Advantages:

Emotional recognition from texts using NLP with prediction probability offers a comprehensive approach to understanding the emotional content in textual data. By providing probability estimates for each emotion class, the system not only enhances insight but also improves accuracy by capturing varying degrees of emotional intensity and uncertainty in text. This nuanced understanding enables users to gain deeper insights into sentiment expressed, facilitating granular analysis and decision support. Moreover, the inclusion of prediction probabilities allows for customization and adaptability, empowering users to tailor the system's thresholds and parameters to suit specific use cases and requirements. Overall, this approach enhances the effectiveness, accuracy, and usability of emotional recognition systems across diverse applications and domains, making them invaluable tools for understanding and responding to human emotions expressed through text.

CHAPTER 8: MODULES IMPLEMENTATION

Modules:

1) Requests – The **requests** module in Python is a powerful HTTP library that allows you to send HTTP requests easily. It simplifies the process of making HTTP requests and handling responses, making it an essential tool for web scraping, accessing web APIs, and interacting with web services.

2) JobLib – The **JobLib** module in Python is primarily used for efficiently saving and loading Python objects, particularly large numerical arrays, to disk. It is especially popular in the context of machine learning and data analysis workflows where saving and loading trained models or intermediate results is common.

3) Altair – The **Altair** module in Python is a declarative statistical visualization library that allows you to create interactive and expressive visualizations with minimal code. It is built on top of the Vega and Vega-Lite visualization grammars, making it powerful yet easy to use for data visualization tasks.

4) TextBlob – The **TextBlob** module in Python is a simplified and intuitive NLP (Natural Language Processing) library built on top of NLTK (Natural Language Toolkit) and Pattern libraries. It offers a straightforward API for common NLP tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

5) Streamlit – The **Streamlit** module in Python is a powerful framework for building interactive web applications with ease. It allows developers and data scientists to create web-based user interfaces for their Python scripts or machine learning models without needing to have expertise in web development.

6) Numpy – The **Numpy** module in Python is a fundamental package for scientific computing, providing support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. It serves as the foundation for many other scientific computing libraries in Python.

7) Pandas – The **Pandas** module in Python is a powerful library for data manipulation and analysis. It provides easy-to-use data structures and functions for working with structured and time-series data, making it a fundamental tool for data scientists, analysts, and developers.

8) NLP - The term "NLP Module" is a bit broad and can refer to different libraries or frameworks in Python specifically designed for Natural Language Processing (NLP) tasks. Some of the popular NLP modules in Python include NLTK (Natural Language Toolkit), spaCy, TextBlob, Transformers (via Hugging Face), and Gensim.

9) Scikit-Learn – The **Scikit-Learn** module, often abbreviated as **sklearn** is one of the most widely used machine learning libraries in Python. It offers a simple and efficient toolset for data mining and data analysis tasks, including classification, regression, clustering, dimensionality reduction, and model selection.

10) VADER (Valence Aware Dictionary and Sentiment Reasoner) - VADER is a lexicon and rule-based sentiment analysis tool specifically designed for social media text. It provides pre-trained models to analyse sentiment in text data, making it suitable for classifying positive and negative feedback.

CHAPTER 9: SYSTEM REQUIREMENTS

SYSTEM REQUIREMENTS:

Development Requirements:

- 1) **Dataset Training** – Jupyter Notebook
- 2) **Project Integrated Development Environment** – Anaconda – Pycharm IDE
- 3) **Raw Script** – MS Excel
- 4) **Language & Module** – Python & Stream-Lit

Hardware Requirements:

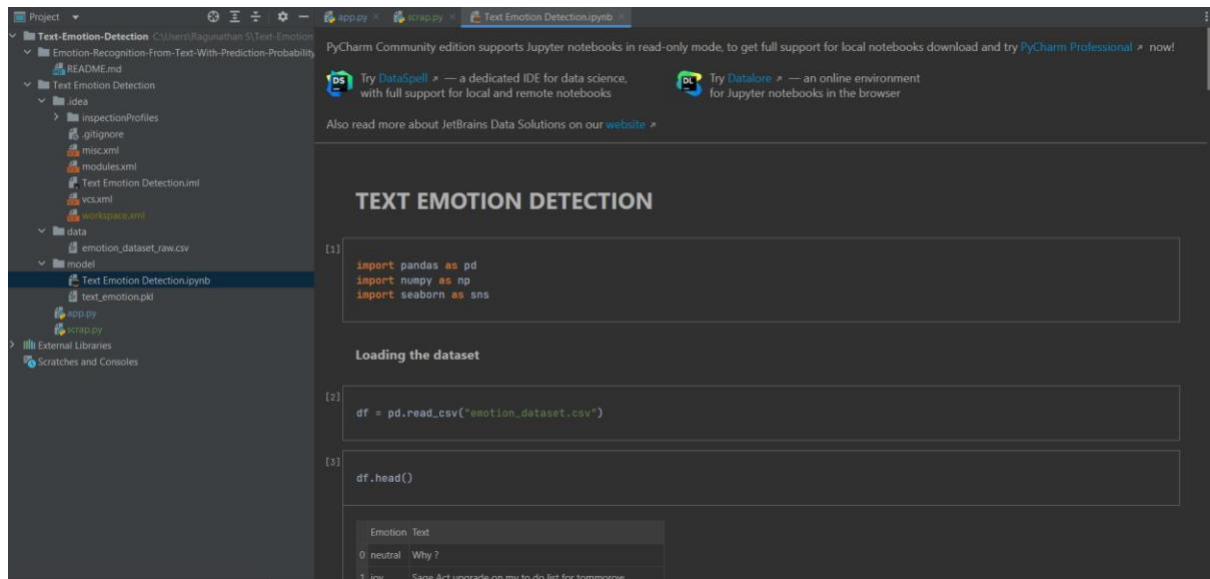
- 1) **Web Browser** – Chrome(Latest Version), Mozilla Firefox, Safari.
- 2) **Processor** – i3 Gen
- 3) **Ram** – 2GB

Software Requirements:

- 1) **Anaconda** => Pycharm IDE
- 2) **Browser (Latest)**

- ☐ Chrome
- ☐ Mozilla Firefox
- ☐ Internet Explorer
- ☐ Safari

OUTPUTS & SCREENSHOTS



PyCharm Community edition supports Jupyter notebooks in read-only mode, to get full support for local notebooks download and try [PyCharm Professional](#) now!

Try [DataSpell](#) — a dedicated IDE for data science, with full support for local and remote notebooks

Try [DataLore](#) — an online environment for Jupyter notebooks in the browser

Also read more about JetBrains Data Solutions on our [website](#)

TEXT EMOTION DETECTION

```
[1] import pandas as pd
import numpy as np
import seaborn as sns
```

Loading the dataset

```
[2] df = pd.read_csv("emotion_dataset.csv")
```

```
[3] df.head()
```

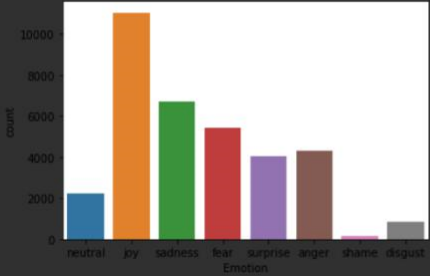
Emotion	Text
0	neutral Why ?
1	joy Sage Act upgrade on my to do list for tomorrow

```
[4] df['Emotion'].value_counts()
```

```
joy          11045
sadness      6722
fear         5410
anger        4297
surprise     4062
neutral      2254
disgust       856
shame        146
Name: Emotion, dtype: int64
```

```
[5] sns.countplot(x='Emotion', data=df)
```

<AxesSubplot:xlabel='Emotion', ylabel='count'>



Emotion	count
neutral	2254
joy	11045
sadness	6722
fear	5410
surprise	4062
anger	4297
shame	146
disgust	856

Detect Emotions In Text

Feedback Classifier

Input Feedbacks

I am Happy to do t

Analyze

Number of positive feedbacks: 1

Text Emotion Detection

Detect Emotions In Text

Type Here

Submit

Product Review Viewer

Enter the URL of the product page:

<https://mamaearth.in/product/vitamin-c-daily-glow-sunscreen-with-vitamin-c-turmeric-for-sun-protect>

Product Reviews

	itype	datePublished	reviewBody
0	review	2024-03-11 11:00:28	This is the first item I ordered in mama earth when I put sunscreen morning upto
1	review	2024-03-06 06:54:57	this is the best sunscreen I have ever used
2	review	2024-03-06 06:43:59	i really like this product
3	review	2024-02-29 18:15:05	Sunscreen is a every day need and i heard alot about Mamaearth's vitamin c sur
4	review	2024-02-29 17:53:55	Mamaearth vitamin C daily glow sunscreen is really good and effective...it's very g
5	review	2024-02-25 18:11:47	This sunscreen is by far the best one that I've used. It really suits me. I just love th
6	review	2024-02-24 12:55:40	This mamaearth vitamin c daily glow sunscreen is best as for daily use. I like to us
7	review	2024-02-20 07:49:55	I like this product very much as it has no harmful effects

Detect Emotions In Text

Type Here

when i am happy the all around us will be happy

Submit

Original Text

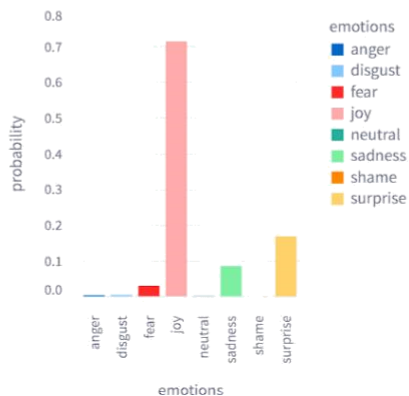
when i am happy the all around us will be happy

Prediction

joy: 😄

Confidence:0.7135842410479502

Prediction Probability



Text Emotion Detection

Detect Emotions In Text

Type Here

i am kolli sad

Submit

Original Text

i am kolli sad

Prediction

sadness: 😞

Confidence: 0.8923421838316793

Infographic Prediction Probability



CHAPTER 10: CONCLUSION & FUTURE ENHANCEMENT

Conclusion:

1) Advanced Classification: Natural Language Processing (NLP) techniques, when applied to emotion detection, leverage sophisticated machine learning algorithms to accurately classify and interpret the emotional content within textual data, enabling a deeper understanding of human emotions expressed through written communication.

2) Enhanced Reliability: The incorporation of prediction probabilities provides valuable insights into the confidence levels of the model's predictions, enhancing the reliability and interpretability of emotion detection systems. This ensures more robust and dependable analysis of emotions in text.

3) Facilitated Decision-making: By offering insights into the certainty of predictions, prediction probabilities assist in better decision-making across various applications, such as emotion detection in customer feedback, emotion tracking on social media, and personalized user experiences. This enables organizations and individuals to make more informed decisions based on the emotional context of written messages.

4) Nuanced Understanding: Prediction probabilities enable a nuanced understanding of emotional expression by capturing varying degrees of emotional intensity and uncertainty in text. This allows for a more comprehensive analysis of emotions, taking into account subtle variations within emotion categories.

5) Improved Responsiveness: NLP models equipped with prediction probabilities empower organizations and individuals to better understand and respond to the feelings represented in written messages. This capability enhances communication strategies, fosters empathetic interactions, and ultimately leads to improved user experiences.

Future Enhancement:

1) Contextual Understanding: Future advancements aim to enhance models' ability to grasp the subtleties of emotions in text by improving contextual understanding. This will enable more accurate interpretations of emotional content, leading to more effective communication and analysis.

2) Integration of Multiple Modalities: Integrating multiple modalities such as text, images, and audio can provide deeper insights into emotional expression. By considering various forms of communication, models can offer a more comprehensive understanding of emotions, enriching the analysis process.

3) Domain Adaptation and Real-time Analysis: Advancements in domain adaptation ensure models are adaptable to different contexts, while real-time analysis capabilities enable timely responses, particularly in critical scenarios like mental health support or crisis management. These developments empower applications to respond effectively to dynamic situations, improving overall outcomes.

REFERENCE

1. Huang, MH., Rust, R.T. (2021). A strategic framework for artificial intelligence in marketing. *J. of the Acad. Mark. Sci.* 49, 30–50 <https://doi.org/10.1007/s11747-020-00749-9>
2. Francisca Adoma Acheampong, Chen Wenyu, Henry Nunoo-Mensah (2020), Text-based emotion detection: Advances, challenges, and opportunities.
3. Wiley Online Library. Chung W, Zeng D, (2018), Dissecting Emotion and User Influence in Social Media Communities: An Interaction Modeling Approach, *Information, and amp; Management*, <https://doi.org/10.1016/j.im.2018.09.008>
4. Sameh Al-Natour, Ozgur Turetken (2020), A comparative assessment of sentiment analysis and star ratings for consumer reviews. *Elsevier International Journal of Information Management* 54, <https://doi.org/10.1016/j.ijinfomgt.2020.102132>
5. S. Godfrey Winster, M. Naveen Kumar (2020), Automatic classification of emotions in news articles through Ensemble decision tree classification techniques. *Journal of Ambient Intelligence and Humanized Computing*, <https://doi.org/10.1007/s12652-020-02373-5>
6. Dimple Chehal, Parul Gupta, Payal Gulati, (2020), Implementation and comparison of topic modeling techniques based on user reviews in e-commerce recommendations, *Journal of Ambient Intelligence and Humanized Computing*, Part of Springer Nature, <https://doi.org/10.1007/s12652-020-01956-6> Graterol.
7. W, Díaz-Amado J, Cardinale Y, Dongo I, Lopes-Silva E, Santos-Libarino C(2021) Emotion Detection for Social Robots Based on NLP Transformers and an Emotion Ontology. *Sensors*, 21,1322. <https://doi.org/10.3390/s21041322>
8. Plutchik R. (1980) A general Psycho Evolutionary theory of emotion. Amsterdam, Netherlands: Elsevier; (pp. 3–33). Russell J. A (1980) A circumplex model of affect. *J Pers Soc Psychol.*;39(6):1161. Scherer KR (2005) Appraisal theory.
9. In: Dalglish T, Power MJ (eds) *Handbook of cognition and emotion*. Wiley, New York, pp 637–663 Ortony A, Clore GL, Collins A (1990) *The Cognitive Structure of Emotions*. Cambridge, MA: Cambridge University Press.
10. Russell JA, Mehrabian A (1977) Evidence for a three-factor theory of emotions. *J Res Pers.*;11(3):273-294. Cambria E., Livingstone A., Hussain A. (2012) *The Hourglass of Emotions*.
11. In: Esposito A., Esposito A.M., Vinciarelli A., Hoffmann R., Müller V.C. (eds) *Cognitive Behavioural Systems. Lecture Notes in Computer Science*, vol 7403. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-34584-5_11
12. Tomás Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, (2013) Efficient Estimation of Word Representations in Vector Space.
13. ICLR (Workshop Poster) Jeffrey Pennington and Richard Socher and Christopher D. Manning (2014) Glove: Global vectors for word representation, In EMNLP.

APPENDIX

Source Code:

Trained Dataset:

```
import pandas as pd
import numpy as np
import seaborn as sns
```

Load The Dataset:

```
df = pd.read_csv("emotion_dataset.csv") // emotion_dataset.csv file //

df.head()
```

Count The Emotion Content In Dataset:

```
df['Emotion'].value_counts()

sns.countplot(x='Emotion',data=df)
```

Dataset Preprocessing:

```
import neattext.functions as nfx

# Remove the user handles
df['Clean_Text'] = df['Text'].apply(nfx.remove_userhandles)
```

Splitting data into input variables and target variable:

```
x = df['Clean_Text']
y = df['Emotion']
```

Splitting data into train and test set:

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=42)
```

Training the Model:

```
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression

pipe_lr = Pipeline(steps=[('cv',CountVectorizer()),('lr',LogisticRegression())])
pipe_lr.fit(x_train,y_train)
pipe_lr.score(x_test,y_test)

pipe_svm = Pipeline(steps=[('cv',CountVectorizer()),('svc', SVC(kernel = 'rbf', C = 10))])
pipe_svm.fit(x_train,y_train)
pipe_svm.score(x_test,y_test)

0.62195822954589

pipe_rf = Pipeline(steps=[('cv',CountVectorizer()),('rf',
RandomForestClassifier(n_estimators=10))])
pipe_rf.fit(x_train,y_train)
pipe_rf.score(x_test,y_test)

0.5632305039279556
```

Saving The Model:

```
import joblib
pipeline_file = open("text_emotion.pkl","wb")
joblib.dump(pipe_lr,pipeline_file)
pipeline_file.close()
```

SCRAP.PY

```
import requests

from bs4 import BeautifulSoup

import json

import streamlit as st

import pandas as pd


def get_reviews(url):

    r = requests.get(url)

    soup = BeautifulSoup(r.content, 'lxml')

    data = json.loads(soup.find('script', type='application/ld+json').text)

    reviews = data.get('review', [])

    return reviews


def main():

    st.title('Product Review Viewer')

    # User input for URL

    url = st.text_input('Enter the URL of the product page:')

    if url:

        try:

            reviews = get_reviews(url)

            if reviews:

                st.subheader('Product Reviews')

                df = pd.DataFrame(reviews)

                st.write(df)

            else:
```

```
        st.error("No reviews found for this product.")
except Exception as e:
    st.error(f"An error occurred: {e}")
```

```
if __name__ == "__main__":
    main()
```

APP.PY

```
import streamlit as st
import re
import pandas as pd
import numpy as np
import altair as alt
import joblib
from textblob import TextBlob
def analyze_sentiment(text):
    blob = TextBlob(text)
    polarity = blob.sentiment.polarity
    if polarity > 0:
        return 'positive'
    elif polarity < 0:
        return 'negative'
    else:
        return 'neutral'

pipe_lr = joblib.load(open("model/text_emotion.pkl", "rb"))
```

```
emotions_emoji_dict = {"anger": "😡", "disgust": "😬", "fear": "😱", "happy": "😄",
"joy": "😂", "neutral": "😐", "sad": "😞",
"sadness": "😓", "shame": "😳", "surprise": "😲"}
```

```
def predict_emotions(docx):
```

```
    results = pipe_lr.predict([docx])
```

```
    return results[0]
```

```
def get_prediction_proba(docx):
```

```
    results = pipe_lr.predict_proba([docx])
```

```
    return results
```

```
def main():
```

```
    st.title("Text Emotion Detection")
```

```
    st.subheader("Detect Emotions In Text")
```

```
    st.title("Feedback Classifier")
```

```
    feedbacks = st.text_area("Input Feedbacks", "")
```

```
    if st.button("Analyze"):
```

```
        if feedbacks:
```

```
            feedback_list = feedbacks.split('\n')
```

```
            positive_count = 0
```

```
            negative_count = 0
```

```
            for feedback in feedback_list:
```

```
                sentiment = analyze_sentiment(feedback)
```

```
                if sentiment == 'positive':
```

```

        positive_count += 1
    elif sentiment == 'negative':
        negative_count += 1
    st.write(f"Number of positive feedbacks: {positive_count}")
    st.write(f"Number of negative feedbacks: {negative_count}")

with st.form(key='my_form'):
    raw_text = st.text_area("Type Here")

    submit_text = st.form_submit_button(label='Submit')

if submit_text:
    col1, col2 = st.columns(2)

    prediction = predict_emotions(raw_text)
    probability = get_prediction_proba(raw_text)

    with col1:
        st.success("Original Text")
        st.write(raw_text)

        st.success("Prediction")
        emoji_icon = emotions_emoji_dict[prediction]
        st.write("{}:{}".format(prediction, emoji_icon))
        st.write("Confidence: {}".format(np.max(probability)))

    with col2:
        st.success("Infographic Prediction Probability")
        #st.write(probability)

```

```

proba_df = pd.DataFrame(probability, columns=pipe_lr.classes_)
#st.write(proba_df.T)

proba_df_clean = proba_df.T.reset_index()
proba_df_clean.columns = ["emotions", "probability"]


fig = alt.Chart(proba_df_clean).mark_bar().encode(x='emotions', y='probability',
color='emotions')

st.altair_chart(fig, use_container_width=True)


if __name__ == '__main__':
    main()

```