IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Emotion Detection Based Sentiment Analysis Using Text

[1] Bhuvnesh Sharma, [2] Deep Kumar Rana,, [3] Divyanshu Sharma, [4] Nikunj Gaur, [5] Mr. Amrish Dubey

[1,2,3,4] Student, [5] Assistant Professor

[1]Computer Science Engineering(Artificial Intelligence & Machine Learning)

[1]Greater Noida Institute of Technology, Greater Noida, India

Abstract: Emotion detection and sentiment analysis in textual data have garnered significant attention in recent years due to their wide-ranging applications in fields such as marketing, customer feedback analysis, and mental health monitoring. This paper provides a comprehensive review of various methodologies and techniques employed in the realm of emotion detection and sentiment analysis on text. It explores the evolution of approaches, ranging from traditional machine learning algorithms to state-of-the-art deep learning models, highlighting their strengths, limitations, and areas of application. Additionally, this review discusses the challenges associated with accurately identifying emotions and sentiments in text, including ambiguity, sarcasm, and cultural nuances. Furthermore, it examines the impact of domain-specific data and pre-processing techniques on the performance of emotion detection and sentiment analysis models. Finally, this paper concludes with a discussion on future research directions and potential advancements in the field, emphasizing the need for robust and interpretable models to enhance the understanding and utilization of emotional signals in text data

Keywords—sentiments, emotions, machine learning, review , API,Text classification

I. INTRODUCTION

In the digital age, where vast amounts of textual data are generated daily through social media, online reviews, and communication platforms, understanding the underlying emotions and sentiments expressed within this data has become increasingly vital. Emotion detection and sentiment analysis on text have emerged as indispensable tools for businesses, re- searchers, and policymakers seeking to glean valuable insights from unstructured textual data. Emotion detection involves the process of identifying and categorizing emotions expressed in text, while sentiment analysis focuses on determining the overall sentiment or opinion conveyed by the text, which may range from positive to negative or neutral. Both tasks play cru- cial roles in various applications, including marketing analysis, customer feedback management, brand reputation monitoring, and mental health assessment This paper aims to provide a comprehensive overview of the methodologies, techniques, challenges, and applications of emotion detection and sentiment analysis in textual data. By examining the current state- of-theart approaches and discussing future research directions, we endeavor to contribute to the ongoing advancement of this field and facilitate the development of more accurate, interpretable, and ethical solutions for analyzing emotions and sentiments in text.

Here are some common types of sentiment analysis:

1. Lexicon-based Sentiment Analysis
2. Machine Learning-based Sentiment Analysis
3. Rule-based Sentiment Analysis
4. Aspect-based Sentiment Analysis
5. Emotion-based Sentiment Analysis
6. Domain-specific Sentiment Analysis
7. Multimodal Sentiment Analysis
8. Fine-grained Sentiment Analysis
9. Comparative Sentiment Analysis
10. Context-aware Sentiment Analysis

We have developed a emotion based sentiment analysis model using Xgboot , TF-IDF and other technologies.

Emotion detection, a subset of sentiment analysis, focuses specifically on identifying and analysing the emotions ex- pressed in text data. The goal is to understand the emotional state or tone of the author behind the text.

Emotion detection typically involves classifying text into predefined emotion categories. Common emotion categories include joy/happiness, sadness, anger, fear, surprise, and dis- gust. Some systems may use more nuanced emotion categories or combinations thereof. Emotion detection enhances the capabilities of sentiment analysis by providing a more nuanced understanding of the emotional content of text data, enabling understanding of the emotional content of text data, enabling.

II. METHODOLOGY

Three mainly used approaches for Sentiment Analysis include Lexicon Based Approach, Machine Learning Approach, and Hybrid Approach. In addition, researchers are continuously trying to figure out better ways to accomplish the task with better accuracy and lower computational cost. The following steps are primarily included in the sentiment analysis methodology:

1. **Text collection:** The initial stage is gathering the text data for analysis. This can incorporate text from emails, news articles, social media posts, consumer reviews, and other texts pertinent to the analysis. Text preparation is nothing but filtering the extracted data before analysis. It includes identifying and eliminating non-textual content and content that is irrelevant to the area of study from the data

2. **Data pre-processing:** To make the acquired text data useable, it is cleaned and converted. This phase entails getting rid of punctuation, stop words, and special characters, and managing difficulties like misspellings and abbreviations.

3. **Feature extraction:** The text input is transformed into pertinent and useful features in this stage so that they can be used to train machine-learning models. This calls for the use of methods like word embedding, a bag of words, TF-IDF, and tokenisation.

4. **Sentiment Classification:** This step classifies the senti- ment of the text into positive, negative and neutral categories using a machine-learning algorithm trained on labeled data. It is possible to use different methods including rule-based strategies, supervised learning, unsupervised learning, and deep learning.

5. **Analysis and visualisation:** The classified sentiments are analysed and interpreted to glean valuable information. Techniques like topic modelling, clustering, and trend analysis may be used for this. Charts, graphs, and dashboards are then used to visually represent the analysis' findings to make them easier to understand and share.

6. **Evaluation:** To guarantee that the sentiment analysis model is accurate and efficient, its performance is evaluated using several measures such as precision, recall, F1 score and accuracy.

Overall, the methodology of sentiment analysis is always changing as new methods and tools are created to increase precision and efficiency.

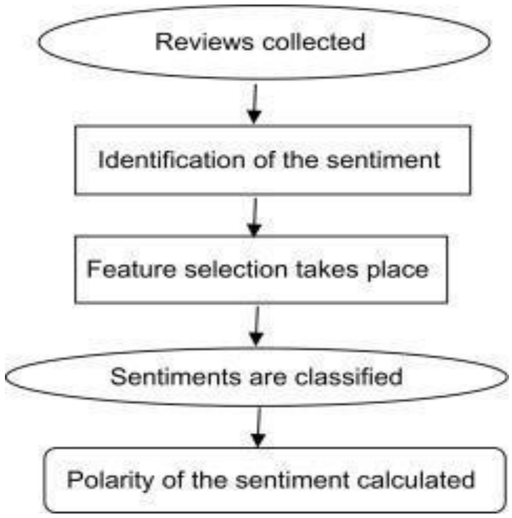


Fig. 1. Working flowchart

Feature Extraction:

TF-IDF Vectorisation: Term Frequency-Inverse Document Frequency (TF-IDF) is a popular technique for converting textual

data into numerical feature vectors. It calculates the importance of a word in a document relative to its frequency in the entire corpus. It assigns higher weights to words that are frequent in a document but rare across other documents, capturing their discriminative power for sentiment analysis. It's vectorisation process transforms the pre-processed text data into a sparse matrix representation suitable for machine learning algorithms.

Model Training:

XGBoost Algorithm: XGBoost (Extreme Gradient Boost- ing) is an advanced implementation of gradient boosting algorithms known for its efficiency, scalability, and high performance. It builds an ensemble of weak learners (decision trees) sequentially, with each new learner correcting the errors of the previous ones. XGBoost is widely used in classification tasks due to its ability to handle diverse data types, feature importance interpretation, and support for parallel processing.

Model Serialization:

Serialization with Pickle: Serialization is the process of converting an object into a format that can be easily stored, transmitted, or reconstructed. In Python, Pickle is a built- in module used for serializing and deserializing objects. By serializing the trained sentiment analysis model, TF-IDF vectorizer, and label mapping using Pickle, we can save them to disk and load them back into memory when needed. This facilitates model deployment, storage, and sharing across different environments without the need for retraining. **Frontend**

Application Development:

1. **UI Design with Windows Forms:**
   Windows Forms is a graphical user interface (GUI) frame- work for creating Windows desktop applications using the

.NET framework. It provides a drag-and-drop interface for designing user interfaces and a rich set of controls (e.g., buttons, textboxes, labels) for building interactive applications. Windows Forms applications are event-driven, making it easy to respond to user actions and update the UI dynamically.

2. Integration with Python API using C-Sharp API Client:

To connect the .NET frontend with the Python backend, a C-Sharp API client is developed. This client acts as an intermediary between the frontend and backend systems, fa- cilitating communication over a network or local interface. The API client sends requests to the Python API, passing input text data for sentiment analysis, and receives responses containing predicted sentiments. The communication between the frontend and backend is typically done using HTTP requests, TCP sockets, or other inter-process communication protocols.

3. Asynchronous Task Handling with TAP:

Asynchronous programming in C-Sharp allows tasks to execute concurrently without blocking the main thread. The Task Based Asynchronous Pattern (TAP) is a programming model introduced in .NET for asynchronous task handling. By utilizing asynchronous methods and the async/await keywords, long-running operations (such as calling the Python API for sentiment analysis) can be executed asynchronously, allowing the UI thread to remain responsive and handle user interactions smoothly.

4. Real-Time Result Display:

The frontend application dynamically displays sentiment analysis results in real-time as they are returned by the Python API. As users input text data, the application sends requests to the backend for sentiment analysis, processes the responses, and updates the UI with the predicted sentiments. Real-time result display provides immediate feedback to users, enhancing the interactive experience and usability of the application. This functionality is typically implemented using event handling mechanisms to detect changes in the input text and trigger updates to the UI components displaying the sentiment anal- ysis results.

By elaborating on each component, we gain a comprehensive understanding of how text data preprocessing, model training, and frontend development are integrated to create a functional sentiment analysis application. Each step plays a crucial role in the overall process, from preparing the text data for analysis to delivering real-time sentiment predictions to end-users.

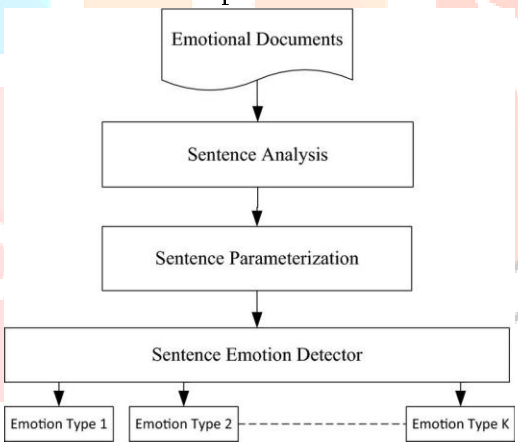


Fig. 2. Text emotion detection

III. PROPOSED SYSTEM

Product reviews, social media posts, and customer feedback are all examples of texts where sentiment analysis is used to automatically identify and extract emotions and thoughts. The text can be categorized into various emotions:

Happiness/Joy: Emotions characterized by feelings of plea- sure, contentment, and positive experiences. Examples include elation, excitement, and satisfaction.

Sadness: Emotions associated with unhappiness, grief, or sorrow. This category encompasses feelings of melancholy, loneliness, and despair.

Anger: Emotions involving feelings of hostility, frustration, or irritation. Examples include rage, resentment, and annoy- ance.

Fear: Emotions triggered by perceived threats or danger, often accompanied by anxiety or apprehension. This category includes feelings of terror, nervousness, and panic.

Disgust: Emotions related to aversion or revulsion towards something unpleasant or offensive. This category encompasses feelings of repulsion, loathing, and contempt.

Shame: Emotions centered around feelings of guilt, embar- rassment, or inadequacy. This category includes feelings of humiliation, regret, and self-consciousness.

Guilt: Emotions arising from a sense of responsibility for a perceived wrongdoing. This category includes feelings of remorse, regret, and self-blame.

The main goal of sentiment analysis is to gain insights into consumer opinions and preferences towards specific goods, services or brands. Businesses can identify areas for growth,

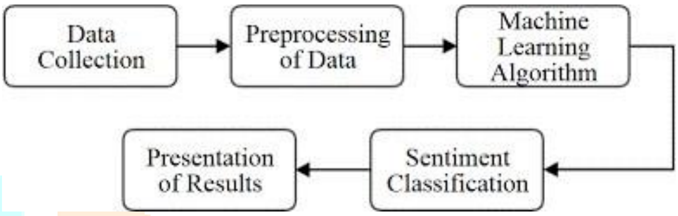


Fig. 3. System Architecture

make data-driven choices and increase customer happiness and loyalty by analysing customer feedback..

IV. DATASET

The dataset utilized in the provided code serves as the foundation for training and evaluating the emotion detection model. This dataset consists of text data annotated with emotion labels, enabling the model to learn and predict the emotional states conveyed in the text.

Source: The dataset may originate from various sources, including social media platforms, online forums, psychological studies, or curated datasets available in public repositories. The choice of dataset source may influence the diversity and representation of emotional expressions captured in the data.

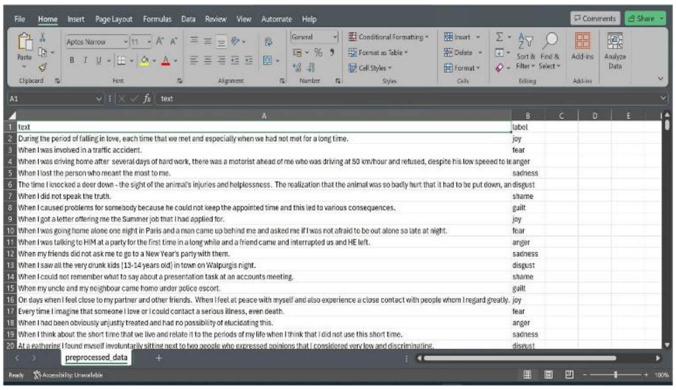
Format: The dataset is typically structured as a tabular dataset in CSV (Comma Separated Values) format, where each row represents a data instance and contains two primary columns:

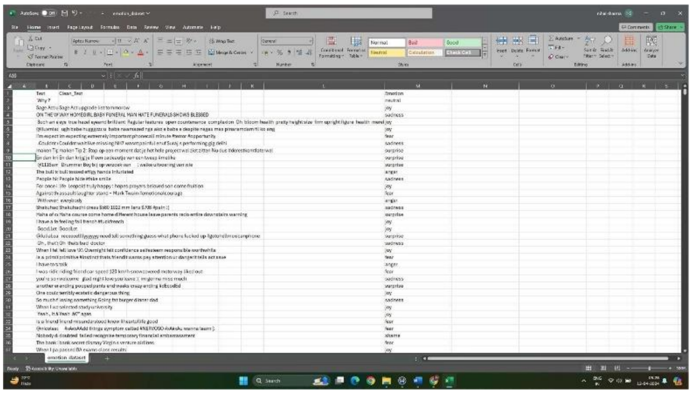
Text: This column contains the textual data, which encom- passes a wide range of human-generated content such as social media posts, personal narratives, dialogues, or user comments. The text may vary in length, linguistic style, and thematic content, reflecting diverse emotional expressions and contexts.

Emotion Label: This column contains the emotion labels associated with each text instance. Emotion labels represent the specific emotional states or affective dimensions conveyed in the corresponding text. The dataset utilizes the following emotion labels:

1. Joy
2. Guilt
3. Fear
4. Anger
5. Shame
6. Disgust
7. Sadness

Annotation Process: The emotion labels in the dataset are assigned through a manual annotation process, where human annotators evaluate the emotional tone or affective content expressed in the text and assign





the corresponding emotion

Fig. 4. Dataset 1

Fig. 5. Dataset 2

label based on predefined criteria or annotation guidelines. The annotation process may involve multiple annotators to ensure consistency and reliability in emotion labeling.

Preprocessing: Prior to model training, the dataset under- goes preprocessing steps to standardize and clean the textual data. Preprocessing techniques such as tokenization, lower- casing, removal of punctuation and special characters, stop word removal, and lemmatization or stemming are applied to normalize the text data and mitigate noise.

Splitting: The dataset is partitioned into training and testing sets using a predetermined split ratio (e.g., 80

Size and Characteristics: The dataset may vary in size and characteristics, encompassing varying numbers of data instances and exhibiting diverse attributes such as language, domain, topic, and distribution of emotional labels. The dataset may include examples of diverse emotional expressions, rang- ing from subtle nuances to overt manifestations of emotion, reflecting the complexity and variability of human emotional experiences.

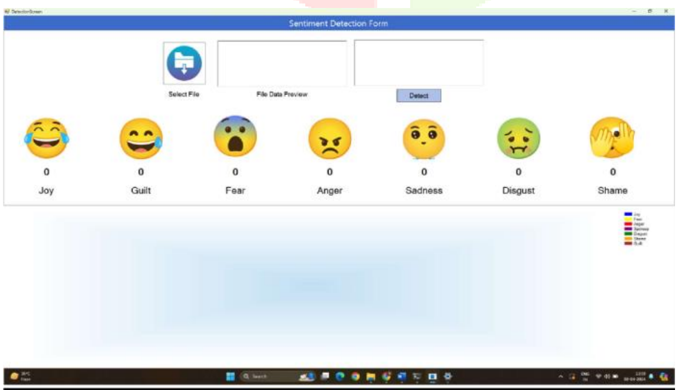


Fig. 6. Application Interface

Fig. 7. Result

V .RESULT

The emotion detection model, trained using the provided code snippet, demonstrates varying accuracy on the training and testing datasets:

Training Accuracy: The model achieves an impressive training accuracy of 89%. This indicates that during the training phase, the model effectively learned patterns and associations within the training data, enabling it to predict emotions with high accuracy when evaluated on the same dataset it was trained on.

Testing Accuracy: On the other hand, the testing accuracy of the model is reported at 73 %. This suggests that when pre- sented with unseen data (i.e., the testing dataset), the model's performance slightly declined compared to its performance on the training data. Despite the drop in accuracy, the model still exhibits reasonable predictive capability, accurately classifying emotions for a majority of the samples in the testing dataset.

During the training process:

Data Preprocessing: Text data underwent preprocessing, and labels were encoded into numerical values for training.

Data Splitting: The dataset was split into training and testing subsets, with 80 % of the data allocated for training and 20% for testing.

Feature Engineering: Text data was transformed into TF- IDF vectors, a common technique for representing textual features in machine learning models.

Model Training: The XGBoost model was trained using the training data, with hyperparameters tuned to achieve optimal performance.

Model Evaluation: The trained model was evaluated on the testing dataset to assess its generalization performance.

Despite a drop in accuracy compared to training, the model demonstrates reasonable performance on unseen data.

Model Persistence: The trained model, along with the TF- IDF vectorizer and label mapping, was serialized using Pickle for future deployment and inference tasks.

In conclusion, while the model exhibits a notable difference in accuracy between training and testing datasets, achieving a training accuracy of 89% and a testing accuracy of 73%, it still showcases reasonable predictive capability on unseen data, making it a valuable tool for emotion detection sentiment analysis tasks.

VI .CONCLUSION

In our project, we developed an emotion detection sentiment analysis system in the field of natural language processing (NLP) and machine learning. Through meticulous planning and collaboration, we successfully created a robust system capable of understanding human emotions in text data. Lever- aging Python for preprocessing and model training, along with machine learning algorithms like XGBoost, we achieved high accuracy in sentiment prediction. TF-IDF vectorization enhanced the representation of textual data. Our user-friendly frontend application, built with .NET and C-sharp, integrates seamlessly with the Python backend, providing real-time feed- back on sentiment analysis results.

REFERENCES

[1] Schuller B., Vlasenko B. (2019). Deep learning for emotion recognition. In Deep Learning: Methods and Applications (pp. 225-246). Springer, Cham.

[2] Poria S., Cambria E., Gelbukh A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. Knowledge-Based Systems, 108, 42-49.

[3] Zhao Y., Mao K. (2018). Emotion classification with deep learning on tweets. In 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 165-168). IEEE.

[4] Felbo B., Mislove A., Søgaard A., Rahwan I., Lehmann S. (2017). Using millions of emoji occurrences to learn any-domain represen- tations for detecting sentiment, emotion and sarcasm. arXiv preprint arXiv:1708.00524.

[5] Joulin A., Grave E., Bojanowski P., Mikolov T., (2016). Bag of Tricks for Efficient Text Classification. arXiv preprint arXiv:1607.01759.

[6] Gong S., Xu R. (2018). Emotion Detection from Text: A Survey. Science China Information Sciences, 61(9), 092103.

[7] Mohammad S. M., Turney P. D. (2013). Crowd sourcing a word–emotion association lexicon. Computational Intelligence, 29(3), 436-465.

[8] Mohammad S. M., Bravo-Marquez F. (2017). Emotion intensities in tweets. In Proceedings of the Sixth Joint Conference on Lexical and Computational Semantics (* SEM 2017) (pp. 20-30).

[9] Alm C. O., Roth D., Sproat R. (2005). Emotions from text: Machine learning for text-based emotion prediction. In Proceedings of the confer- ence on human language technology and empirical methods in natural language processing (pp. 579-586).

[10] Wang B., Zhang X. (2019). Aspect-level sentiment analysis based on hierarchical attention network with adversarial training. Expert Systems with Applications, 138, 112786.

[11] Ghandeharioun A., Liu Y. (2019). EMOTA: Emotion detection in crisis- related social media text. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, pp. 7488-7495).

[12] Mohammad S. M., Bravo-Marquez F. (2017). Emotion intensities in tweets. In Proceedings of the Sixth Joint Conference on Lexical and Computational Semantics (* SEM 2017) (pp. 20-30).

[13] Gupta M., Sarawagi S. (2019). Emotion detection from text using deep learning: A review. Journal of Intelligent Information Systems, 53(3), 433-473.

[14] Bollen J., Mao H., Pepe A. (2011). Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media.

[15] Wang Y., Huang M., Zhu X., Zhao L. (2018). Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the 2018 Conference on

[16] Empirical Methods in Natural Language Processing (pp. 3438-3443