1 Bis. Kaggle Text to Emotion Dataset

Md Moinul Islam (2307268)
Taufiq Ahmed (2307644)
Computer Science and Engineering
University of Oulu

GitHub: https://github.com/moinul7002/1bis.-Kaggle-Text-to-Emotion-Dataset

***Abstract*—In this project, we conducted an extensive analysis of a Twitter emotion dataset containing 40,000 records classified into 13 emotion categories. Our exploration began by visualizing the distribution of training samples across these categories, shedding light on the dataset's inherent imbalances. We then aimed to simplify the categorization by manually crafting three overarching categories: Positive, Negative, and Neutral. Additionally, we leveraged word2vec embeddings and k-means clustering to create an alternative categorization, providing insights into the relationship between word semantics and emotions. Furthermore, we assessed the quality of these categorizations through multiple metrics, including token overlap and the presence of common frequent words among records. We also incorporated the NRC lexicon to explore the connection between the dataset's labels and emotional word associations. Our analysis revealed the Pearson correlation between dataset labels and lexicon-based scores for each category. To enhance the understanding of category content and titles, we compared the similarity between category titles and the content of records using various word embeddings (word2vec, Glove and FastText). This analysis sought to uncover the degree of alignment between title-based and record-based similarities. In conclusion, our study provides a comprehensive assessment of categorization strategies and their alignment with semantic and stylometric features, contributing to the understanding of emotions in Twitter data.**

***Index Terms*—Emotion categorization, Word2Vec embeddings, k-Means clustering, most frequent words, token overlap, word semantics, Pearson correlation, NRC word emotion association lexicon, WordNet, WordCloud, semantic and stylometric features, cosine similarity**

I. Introduction

Artificial Intelligence (AI) has made significant contributions to natural language processing (NLP), a field of computer science that deals with the interaction between computers and human language. NLP techniques use computational and linguistic methods to enable computers to understand and generate human language in the form of text and speech. Since its inception in the 1950s, AI has helped to improve the accuracy and efficiency of NLP tasks such as machine translation, text summarization and sentiment analysis. AI has also been used to develop new NLP applications, such as chatbots and virtual assistants, which can interact with humans in a natural and engaging way. The contributions of AI to NLP have had a major impact on a wide range of industries and fields, including healthcare, education, customer service, and marketing. For example, NLP is now being used to develop new medical diagnostic tools, create personalized learning experiences for students, and improve the customer experience.

Overall, AI has played a vital role in the advancement of NLP, and it is likely to continue to do so in the future. As AI continues to develop, we can expect to see even more innovative and powerful NLP applications emerge. Natural language processing (NLP) is a field of computer science that deals with the interaction between computers and human language. NLP researchers are actively developing new and improved NLP technologies, such as translation systems, information retrieval (IR), question and answer (Q&A) systems, text summarization systems, and sentiment analysis (SA). One subfield of NLP, known as emotion detection (ED), focuses on extracting more detailed emotions from human language, such as happiness, sadness and anger. This contrasts with SA, which assigns a general polarity of positive, negative, or neutral. Emotion detection (ED) is important because it can help computers to better understand and respond to human emotions. This can be useful in a variety of applications, such as Customer service: emotion detection can be used to identify customer sentiment and ensure that customers are satisfied with their experience. In healthcare, emotion detection can be used to develop new tools for diagnosing and treating mental health disorders. In education, it can be used to create personalized learning experiences for students and identify students who may need additional support. Overall, emotion detection is a promising field of research with a wide range of potential applications.

As emotion detection technologies continue to develop, we can expect to see them used in even more innovative and impactful ways. One of the best resources for researchers to analyze emotion from text is Twitter data. The rapid growth of the World Wide Web has led to a significant increase in the use of social media platforms like Twitter, where people share their opinions through tweets. However, from a human perspective, this results in a vast amount of information that is impractical to manually extract, read, assess on a tweet-by-tweet basis, summarize, and organize into a coherent structure within a reasonable timeframe. Sentiment analysis is a technique used to identify and extract opinions and emotions from text. It is often used to analyze customer feedback to understand how customers feel about a company's products or services. Microblogging platforms such as Twitter are a valuable source of customer feedback, as they allow customers to share their thoughts and opinions in a real-time and public manner. However, the large volume of data generated by microblogging platforms can make it difficult to manually analyze.

Text is a thriving source of knowledge, but it can be a bit challenging and time-consuming to get insight because it is unstructured. Emotion classification from text has been more prominent in recent years due to the increased usage of emotions in marketing, political science, psychology, human-computer interaction, and artificial intelligence. In text classification, very much anything may be arranged, structured and categorized. New articles, for example, may be ordered by subject and tickets for assistance can be handled urgently, chat chats by language can be arranged, etc. This includes an independent model for text analysis and one of the most active fields in natural language processing (NLP), data mining and text mining [14]. The rise of social micro-blogging sites, news portals, online markets and other applications produces a very large volume of data every day in text format. This type of text categorization, therefore, plays a key role in the rapid use of huge amounts of text data.

Text emotion classification includes automatic assignment from a specified list of categories. Due to the growing number of virtual platform users and the quick production of online material, reading emotions in online information is crucial. This applies to customers, corporations, experts and other individuals. There are six basic emotions according to Ekman, which are happiness, sadness, fear, anger, surprise and disgust [5]. These emotions are mostly identifiable by facial expressions. Those feelings can also be extracted from texts.

This study aims to address this challenge by developing new and improved methods for using sentiment analysis to understand the user's opinion in microblogging data. The study will also develop a software tool that can be used to process and analyze large amounts of user's tweet data in a more efficient and effective way. This study also shows similarities between various types of sentiment classes as well.

This project outlines a thorough approach and a set of objectives with the purpose of developing a model for classifying emotions from Twitter texts. The tasks are developed by utilizing a publicly accessible dataset. The primary objectives and contributions of this project are as follows:

1) We explored the distribution of the Twitter dataset across various emotion categories.
2) We preprocessed the twitter texts by removing stop-words, unusual characters, symbols, numbers and repetitions.
3) We concatenated tweets within each emotion category and extracted top 10 most frequent keywords before and after the text preprocessing.
4) We generated WordCloud for visualization of the each emotion category.
5) We transformed the existing categories to only three simple categories by constructing manually and word2vec embeddings along with kMeans.
6) We compared the two constructions in terms of token overlap and most frequent words, considering common tokens in both data frames assigned to the same class.
7) We identified if a given term in all data frames has an entry in the WordNet lexical database and later calculated the number of unmatched tokens in each data frame and their corresponding percentage.
8) We calculated the entity value for each record and category based on the NRC Word-Emotion Association lexicon and the Twitter dataset labels. We constructed vectors V1 and V2 for each category and calculated Pearson correlation coefficients and P-values.
9) We calculated the similarity between records in two categories using word2vec embeddings, Glove and FastText and created a matrix of record-based similarity between each pair of categories.
10) Finally, we developed a robust emotion detection model for texts-based data, capable of classifying texts into different emotion categories.

The rest of this report is organized as follows: Section II discusses the recent works in the current domain. Section III represents an overview of the dataset, the methodology and the performance evaluation metrics. Finally, Section IV provides a brief discussion on the findings of this project.

II. Related Works

There has been a lot of work on text mining. The authors [4] define opining mining as a way of using computers to understand the feelings and opinions that people express in text. It is a complex task, but it has many useful applications, such as helping businesses understand what their customers think about their products and services or helping social scientists track public opinion on current events. The authors state that [10] Opinion mining has a wide range of practical uses across various fields such as accounting, law, research, entertainment, education, technology, politics, and marketing. In the past, the emergence of numerous social media platforms has provided internet users with a platform to freely express and disseminate their ideas and viewpoints. There are lots of microblogging platforms where users can express their opinions. Among them, Twitter is one of the most popular platforms in the world. Users utilize tweets to convey their viewpoints on a wide range of topics related to their everyday experiences. Twitter serves as an optimal platform for gathering the collective opinions of the public on specific matters [10]. By analyzing the sentiment of comments and tweets, we can gain a better understanding of public opinion on a variety of topics [3]. Sentiment analysis is a broad approach used to determine the positivity or negativity of text or phrases by extracting polarity and subjectivity based on the strength of words and their contextual orientation [16]. Two primary methods for automated sentiment extraction are the lexicon-based approach and the machine-learning-based approach [18]. Lexicon-based techniques utilize pre-established word lists in which each word is linked to a particular sentiment [6]. Sentiment analysis involves categorizing text sentiments into various classes, often using lexicons like SentiWordNet. We have gone through some of the work regarding sentiment analysis from Twitter data and there is some interesting work that we would like to mention. The authors introduce a method to expand sentiment lexicons with Word2vec and fastText embeddings, achieving 88.14Machine learning is a type of artificial intelligence that allows computers to learn without being explicitly programmed. Supervised classification is a machine learning task that involves training a model to predict the category of a data point based on a set of labeled training data. In sentiment detection, supervised classification is often used to train a model to predict whether a piece of text is positive or negative. This requires a dataset of labeled text, where each piece of text is labeled as positive or negative. The model is trained on this dataset, and then it can be used to predict the sentiment of new pieces of text [3]. The authors created a live web application to monitor emotions conveyed by individuals in Saudi Arabia by analyzing their tweets. They compiled a dataset of 4000 Arabic words, including emojis, by combining information from three sources: emotional survey responses from people in Saudi Arabia with various dialects, the Almaany Arabic Lexicon, and popular Twitter hashtags. They processed the data and categorized it into six emotions: happy, sad, scared, surprised, angry, and neutral, for instances where words did not convey any emotions [1]. The authors noticed a proliferation of emotion detection studies in English texts compared to Bengali, prompting them to develop a system to identify basic emotions like happiness and sadness in Bengali. They first determined the overall sentiment (positive or negative) in 301 Bengali sentences from a survey using a hash algorithm and compared hash values with a Word Lexicon containing 350 words. Matching values were stored in a database, enabling the detection of sentiments and subsequent classification into happy and sad emotions. Despite 77.16% accuracy, the system's performance was affected due to limited data and words that were not present in the Lexicon, requiring their removal [13]. The authors developed a model combining BiLSTM, self-attention, and CNN and emphasized the importance of effective word embeddings in enhancing text-based emotion detection (ED) system performance. They assessed three-word embeddings - Google Word Embedding, GloVe Embedding, and FastText Embedding - using the ISEAR dataset, the SemEval2018 Task1 dataset, and SemEval-2019 Task3 dataset. Results on the ISEAR dataset favored GloVe and FastText embeddings for most emotion classes, except for sadness. In contrast, FastText embedding outperformed on the SemEval-2018 Task1 and SemEval-2019 Task3 datasets. They recommended the use of FastText embedding for future research [12]. The authors labeled Twitter data with emotion classes using the NRC emotion lexicon and employed SVM for emotion classification. They then identified action patterns, applied action rules to convert negative or neutral emotions into positive states and reclassified them. The system achieved an accuracy of approximately 88The authors used the Naive Bayes (NB) method to develop a Twitter application for detecting user emotions. They collected 105 tweets using the Twitter API and preprocessed them by converting them to lowercase, removing stop words, mentions, URLs, and emoticons. Then, they used the NB classifier to classify the texts into six emotions: happiness, sadness, fear, anger, surprise, and disgust. They used 10-fold cross-validation to evaluate the accuracy of their classification, which was 83%. However, they recommended using a larger training dataset to improve performance in the future. They also suggested removing duplicate tweets to improve the response time of their system. Finally, they mentioned that other classification methods, such as support vector machines (SVMs) and K-nearest neighbors (KNN), could also be used to improve performance [19]. The authors introduced the Emotex model, which is designed to identify emotions in text by applying supervised learning and emotion dictionaries. Their methodology included two components: an offline and an online classification process. In the offline phase, they developed the Emotex system to build emotion classification models using emotion-labeled Twitter data and three classifiers: SVM, NB, and decision tree. Data preprocessing and feature vector construction were conducted to generate training datasets for the classification model. In the online phase, the model developed in the offline stage was employed to categorize real-time Twitter streams. The model achieved 90% accuracy, although it had limitations in terms of semantic features [7]. The authors addressed the persistent challenge of semantic extraction in text-based emotion detection by implementing a two-stage approach for text feature extraction. This process involved a semantic stage, where they extracted meaning and semantics from the text using a POS tagger, followed by a statistical stage where the chi-square method was used to eliminate weak semantic features. They applied this methodology to the ISEAR dataset with the SVM classifier and successfully identified seven emotions: joy, anger, fear, disgust, guilt, shame, and sadness. Their results demonstrated performance improvements over baseline methods. However, a limitation of their approach was its failure to consider feature relationships [15]. The author classified the emotions in the ISEAR database using four different classifiers: logistic regression, SVM, KNN, and XG-Boost. He found that logistic regression outperformed the other classifiers, with a precision of 86%, recall of 84%, and F-Score of 85%. They suggested that using a deep learning technique could further improve the performance of his system [2]. The authors proposed a deep learning model for sentiment and emotion detection in YouTube comments. They collected data and conducted preprocessing to eliminate stop words. Word embedding was obtained using both Skip-Gram and continuous bag of words (CBOW) in Word2Vec. The output was then used as input for their two-phase model, consisting of a long short-term memory (LSTM) architecture in the first phase and a convolutional neural network (CNN) in the second phase. The study demonstrated that their deep learning techniques, LSTM and CNN, significantly outperformed traditional machine learning methods like SVM and NB, achieving an accuracy of 59.2% for emotion classification and 65.97% and 54.24% for multiclass (3 and 5) sentiment label accuracy, respectively [17].

III. Methodology

The main goal of the project is to comprehensively analyze and understand a Twitter emotion dataset containing 40,000 records categorized into 13 emotion categories. This analysis involves several key objectives. Firstly, we aimed to explore the distribution of training samples across these emotion categories to gain insights into potential imbalances in the dataset. Subsequently, we strived to simplify the categorization by manually crafting three overarching emotional categories: Positive, Negative and Neutral. Additionally, we employed word2vec embeddings and k-means clustering to create an alternative categorization that leverages word semantics. The analysis extends to evaluating the quality of these categorizations using various metrics, including token overlap and the presence of common frequent words among records. Furthermore, we incorporated the NRC Word-Emotion Association lexicon to examine the association between dataset labels and emotional word semantics. We also assessed the alignment between category titles and record-based content similarity using different word embeddings, shedding light on the relationship between the emotional categories and their textual representations. Finally, we evaluated the classification results for different emotion categories. Figure 1 shows an abstract view of the methodology.

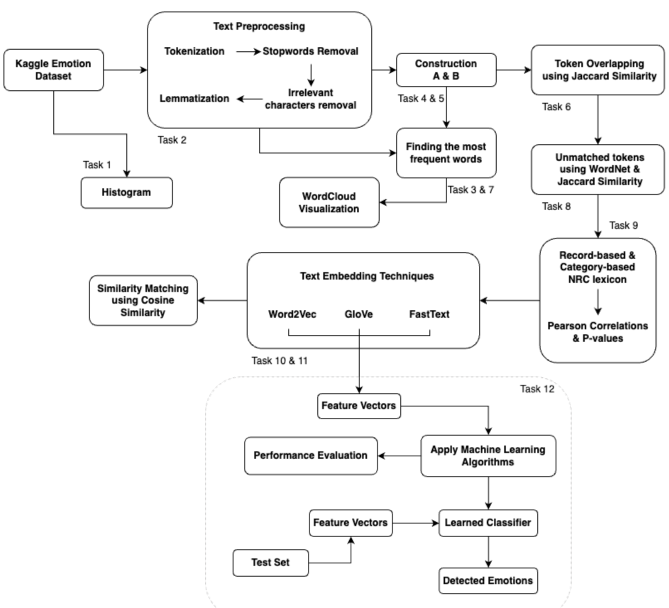

Fig. 1. Overall Methodology for Project 1 Bis

A. Dataset

The Kaggle dataset[1] contains 40,000 twitter messages, each labeled with one of 13 emotions: neutral, worry, happiness, sadness, love, surprise, fun sentiment, relief, hate, empty, enthusiasm, boredom, and anger. This dataset is valuable for natural language processing research, especially for the challenging task of emotion detection from text. Some of the main challenges in this field include the lack of labeled datasets and the complexity of multi-class emotion classification.

[1]https://www.kaggle.com/datasets/pashupatigupta/emotion-detection-from-text/data?select=tweet_emotions.csv

B. Data Statistics

The dataset contains 40,000 Twitter texts divided into 13 different emotion categories. After the preprocessing, repetitive texts were removed and the texts were reduced to 39,837. The statistics of each class with and without preprocessing are represented in Table I.

Class	Total Instances	Tokens (before)	Tokens (after)
Neutral	8531	122124	57039
Worry	8444	146548	57039
Happiness	5205	87929	41510
Sadness	5158	87936	39639
Love	3839	64416	29922
Surprise	2184	37682	16769
Fun	1776	32236	15044
Relief	1525	25904	11995
Hate	1322	23459	10653
Empty	810	11855	5485
Enthusiasm	759	12731	5821
Boredom	179	2966	1395
Anger	110	1928	854

TABLE I
Amount of instances and number of tokens before and after preprocessing in each class

C. Initial Exploration

The dataset doesn't have similar number of instances which may lead to class imbalance problems. This class imbalance can significantly impact the performance of machine learning models, particularly in tasks like sentiment analysis or emotion detection.

To output the histogram of the different emotion categories in the Twitter dataset, we started by loading the dataset from the provided CSV file. We then needed to count the number of records or tweets belonging to each of the 13 emotion categories: neutral, worry, happiness, sadness, love, surprise, fun sentiment, relief, hate, empty, enthusiasm, boredom, and anger. This count for each category represents the frequency of the samples. Utilizing data visualization libraries such as Matplotlib or Seaborn in Python, we created a histogram which can be found in Figure 2, where each category is represented on the x-axis and the corresponding frequency (number of samples) is represented on the y-axis. This histogram provides a visual representation of the distribution of training samples across the various emotion categories. By examining the histogram, we gained insights into the dataset's class distribution, which is crucial for understanding potential imbalances and biases in the data, and for guiding subsequent data processing and modeling decisions.

D. Data Preprocessing

Text preprocessing is an important stage in natural language processing. In order to fill in missing values, smooth noisy

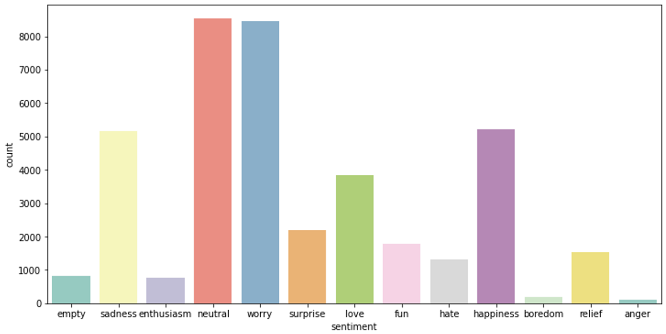

Fig. 2. Histogram of each class

data, to detect or eliminate the outliers and to resolve discrepancies, data pre-processing is necessary. All texts are prepared by the preprocessor in our model. Our emotional classifier model was trained using the core body of the text. A list of terms and their frequencies have been represented in our text document. We have utilized a stop words list which includes words that don't categorize text. These words are eliminated by matching the stop word list in our preprocessing section. It was helpful to enhance the efficiency of the model, since redundancy reduces the functionality of our model and raises the model's computational complexity. Text preprocessing includes removing words and symbols from feature space, and lowering noise and dimension. This helps classifier models learn important features and improve performance. In the domain of Natural Language Processing (NLP), the quality of data profoundly influences the efficacy of resultant models. Preprocessing is paramount, especially for datasets like the Kaggle emotion dataset, which is derived from the bustling world of social media comments. Such comments often encapsulate the eclectic nature of online interactions, replete with emojis, links, and even embedded HTML.

Given this backdrop, cleaning and structuring the data transcends mere standard practice. It involves meticulously removing platform-specific elements like emojis, links, and HTML tags. By doing so, we ensure that our models are primed to identify and learn from relevant patterns, unobstructed by the noise characteristic of social media texts.

1) Tokenization was used to distinguish words in the document set separated by white spaces or new lines and present them as tokens for processing.
2) Removal of special symbols, emoticons or digits (e.g., [0-9, a-z, @,!,#,$,%,&,.,ff,—,:,],+) from data collection were done since they do not help categorize text.
3) Replacement of contractions and conversion of all characters to lowercase were done to find the semantic meaning.
4) Removing stop words helps classifier models capture key traits as most feature extraction methods favor word frequency. A conventional stop-word list were used to remove them from the dataset.
5) Noise, in the context of text data, refers to any piece of information that doesn't contribute meaningfully to the content's essence or may lead the model astray during training.
6) Text data, especially when scraped from web pages, often contains residual HTML tags. These tags, which are instrumental for web browsers, are mere noise for NLP tasks and hence are stripped away.
7) URLs, while crucial for web navigation, don't typically offer semantic value for text analytics. Their diverse and inconsistent structures can mislead models. Thus, removing them ensures that models aren't perplexed by these irrelevant strings.
8) Emojis, though expressive, are graphical symbols that can introduce variability into the dataset. Their removal simplifies the text, paving the way for a more consistent dataset.

E. Text Normalization

Initially, once the texts were denoised, the next step involves structuring it and tokenization is the primary step in that direction. NLTK (Natural Language Toolkit) offers specialized tools for texts and its tokenizer efficiently breaks down text into individual words or tokens. This granularity is essential for models to process and understand text. Tokenization allows models to capture the essence of the language at a granular level, helping in better representation and understanding. By removing stopwords, the dataset becomes more concise, allowing models to focus on words that carry richer semantic value. This can lead to improved model efficiency and performance. Given NLTK's specialization, it offers a curated list of stopwords tailored for texts. This ensures that the removal process is both comprehensive and nuanced. Lemmatization is the process of reducing inflected words to their word stem or root form. It ensures that words derived from the same root are treated as equivalents, making data more consistent for modeling. With its rich morphological variations, it can benefit immensely from lemmatization. The WordNetLemmatizer, by reducing words to their root forms, ensures that the dataset remains dense with information while being devoid of unnecessary variations. A distribution of token length after the preprocessing task is represented in Figure 3

F. Construction A: Manual Crafting

In our project work, we introduced a crucial step of categorization and classification by crafting a new column in the dataset that represents emotions. This manual crafting process aimed to simplify and reorganize the complex landscape of 13 distinct sentiment categories into a more manageable and interpretable format. To achieve this, we established a clear mapping strategy denoted as "construction A." This involved associating each of the 13 original sentiment categories with one of three overarching emotional states: Positive, Neutral, or Negative. This mapping was based on a deep understanding of the underlying sentiment conveyed within the tweets and considered the nuanced expressions of emotions. For instance, tweets originally categorized as

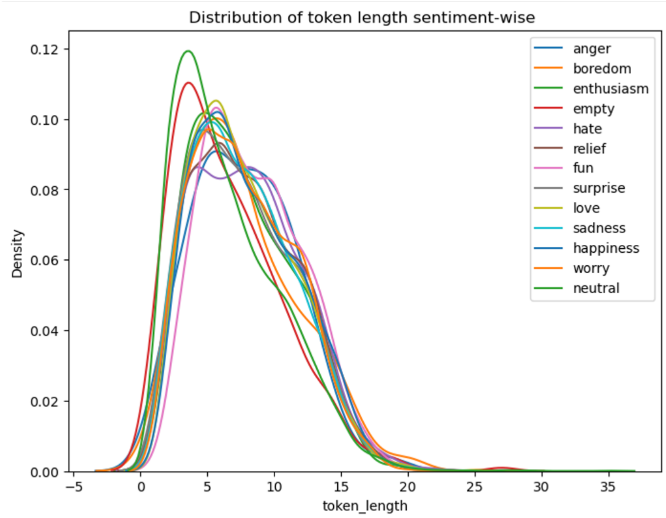

Fig. 3. Distribution of token length after text preprocessing

"happiness," "love," "surprise," and "fun" were collectively mapped to the "Positive" category. In contrast, categories such as "sadness," "worry," "hate," "anger," and "boredom" were regrouped under the "Negative" label. Categories like "empty," "enthusiasm," "neutral," and "relief" were mapped to the "Neutral" category, reflecting emotions that do not fall clearly into the positive or negative spectrum. This mapping process not only simplified the dataset but also aligned it with the fundamental emotional dimensions commonly employed in sentiment analysis, making it more conducive to further analysis and modeling. It provided a structured and concise representation of emotions, enabling more effective research into sentiment trends and patterns within the Twitter dataset. Mapping of the construction A is represented in Table II.

G. Construction B

In our research methodology, we embarked on an innovative approach to constructing a new classification scheme, denoted as "construction B," using Word2Vec embeddings and k-Means clustering techniques. This method sought to reorganize the 13 original sentiment categories into a more data-driven, semantically cohesive structure, enhancing our understanding of the emotional landscape in the Twitter dataset. We initiated this process by training a Word2Vec model on the category titles, converting the category names into distributed word embeddings in a continuous vector space. Word2Vec, a widely utilized natural language processing technique, allowed us to capture the semantic relationships between category titles, thus extracting the underlying structure of emotions as represented in the dataset. Following the embedding generation, we employed k-means clustering, with the aim of grouping the category titles into three distinct clusters, effectively representing the Positive, Neutral, and Negative emotional dimensions. This data-driven approach resulted in a new set of labels for the categories, aligning them with their most semantically related clusters. The result of this approach was a redefined categorization of emotions, one that relied on the intrinsic semantic associations present in the category titles. By employing Word2Vec embeddings and clustering, we introduced a novel perspective that offered a more data-centric and contextually grounded representation of emotions. This mapping strategy not only simplified the dataset but also enhanced the granularity of the sentiment analysis, potentially capturing subtleties in emotional expression that may not have been apparent in the original categories. This innovative technique provided a valuable and data-driven alternative to emotion classification, offering new insights into the sentiment landscape of the Twitter dataset. Mapping of the construction B is represented in Table II.

Class	Construction A	Construction B
Empty	Neutral	Neutral
Sadness	Negative	Negative
Enthusiasm	Positive	Neutral
Neutral	Neutral	Neutral
Worry	Negative	Negative
Surprise	Positive	Neutral
Love	Positive	Negative
Fun	Positive	Positive
Hate	Negative	Negative
Happiness	Positive	Positive
Boredom	Negative	Negative
Relief	Positive	Positive
Anger	Negative	Neutral

TABLE II
AMOUNT OF INSTANCES AND NUMBER OF TOKENS BEFORE AND AFTER PREPROCESSING IN EACH CLASS

The total number of instances with the newly constructed data frames A & B are represented in Table III.

Construction	Neutral	Negative	Positive
A	62524	118386	121061
B	85968	147454	68549

TABLE III
AMOUNT OF INSTANCES IN EACH CLASS

H. Most Frequent Tokens

We conducted a detailed analysis of the Twitter dataset categorized into 13 emotion classes. We aimed to uncover the most frequent keywords in each category before and after text preprocessing. To achieve this, we utilized a Python script and data visualization techniques. The script first grouped tweets by sentiment category and concatenated them into single text documents. We then preprocessed the text data, removing stop words and other unwanted elements such as symbols and numbers. After this data preparation stage, we calculated the top 10 most frequent keywords for each category. The representations of the 10 most frequent words of one of the classes (Neutral) using histogram can be found in Figure 4 (before preprocessing) and Figure 5 (after preprocessing).

Our findings provided valuable insights into the most prevalent terms associated with different emotions. For example, we discovered that in the "happiness" category, keywords like "happy" and "joy" were frequently mentioned. Preprocessing

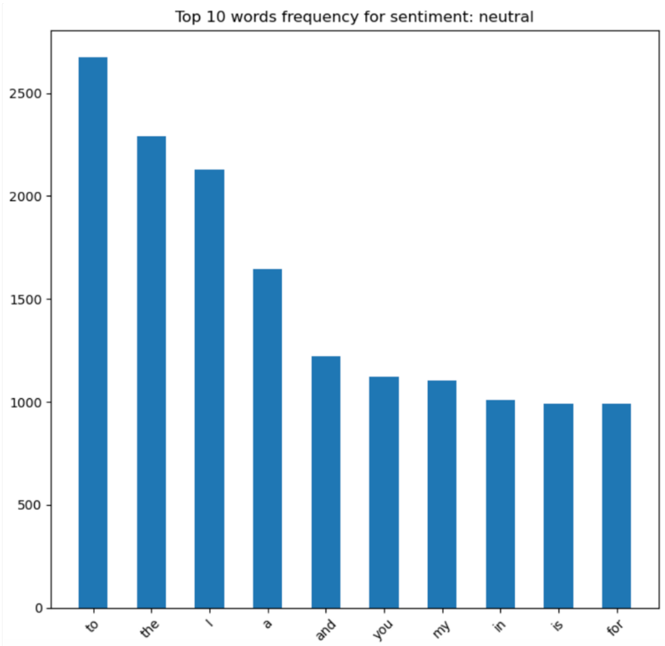

Fig. 4. Top 10 words frequency for sentiment ”Neutral” before text preprocessing

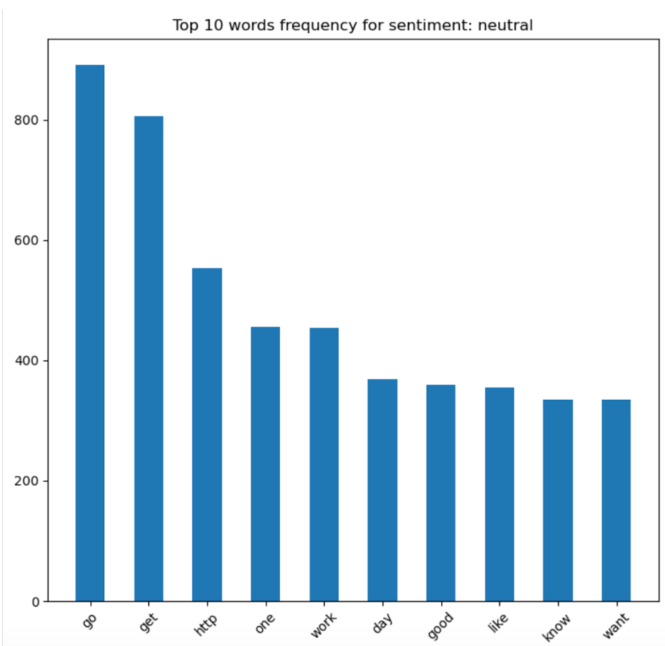

Fig. 5. Top 10 words frequency for sentiment ”Neutral” after text preprocessing

was essential to remove noise and ensure meaningful results. To visualize the keyword distribution, we created bar charts displaying the frequency of the top 10 words in each category and word clouds to present the overall content graphically. This methodology allowed us to gain a better understanding of the language used to express various emotions on Twitter, which is essential for further sentiment analysis and emotion detection tasks.

In our project, we leveraged WordCloud as a powerful visual tool to complement our analysis of the Twitter dataset categorized into 13 distinct emotion classes. Word clouds provided a striking and intuitive way to showcase the most prominent terms within each sentiment category, offering a visually appealing representation of the textual data. These word clouds were generated for each emotion category before and after preprocessing the text data. Each word cloud was a vivid display of words, with the size of each word being directly proportional to its frequency within the category's tweets. As a result, the most frequently occurring terms appeared larger and more prominently in the cloud, making it easy to identify the dominant keywords associated with a specific emotion. The word clouds served as a powerful means of summarizing the content and sentiment of each category. They allowed us to quickly grasp the essence of the language used to express emotions on Twitter, offering a clear visual distinction between categories. Additionally, word clouds were instrumental in providing a high-level overview of the dataset, making it easier to identify the key terms that characterize each emotion. Overall, word clouds added a dynamic and visually engaging dimension to our research, enhancing our ability to communicate the findings and insights related to the emotional expressions in the Twitter dataset. The representations of one of the classes (Neutral) using WordCloud can be found in Figure 6 (before preprocessing) and Figure 7 (after preprocessing).

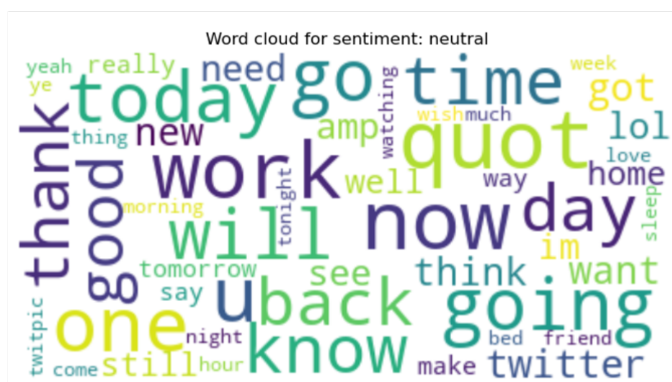

Fig. 6. Top 10 words frequency for sentiment ”Neutral” before text preprocessing

I. Tokens Overlapping

In our project, we undertook a systematic and data-driven comparison between the two distinct methods of emotion categorization, denoted as ”construction A” and ”construction

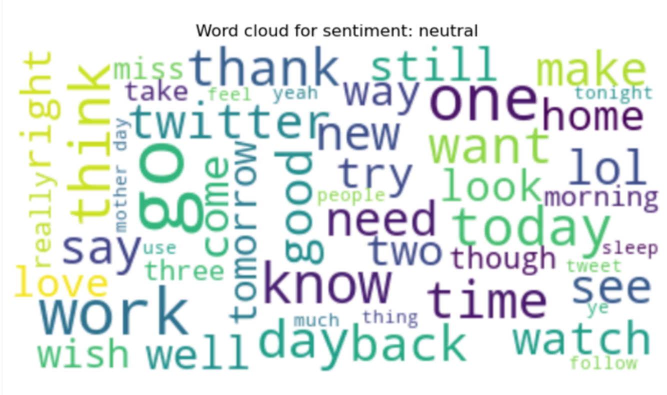

Fig. 7. Top 10 words frequency for sentiment "Neutral" after text preprocessing

B," and sought to evaluate the degree of token overlap between the resulting sentiment classes. This step was crucial in understanding the commonalities and differences between the two classification approaches. To achieve this, we aggregated tweets within each sentiment category for both constructions, yielding two consolidated datasets representing **Construction A** and **Construction B**. These datasets comprised tweets categorized into Positive, Negative, and Neutral emotions based on the respective classification approaches. We then developed a custom Python script to calculate the number of common tokens, or overlapping words, among these constructed categories in IV. This analysis revealed how many words were shared between the sentiment classes created using "construction A" and "construction B." The results demonstrated the level of consistency and correspondence between the two classification methods. This evaluation served as a rational and quantitative metric to assess the degree to which the alternative approaches aligned with each other in categorizing emotions. It shed light on the extent to which words, and thus emotions, were similarly grouped and understood by both methods. By quantifying the token overlap, we gained valuable insights into the concordance and divergence in categorization between the two constructions, facilitating a more informed assessment of their respective strengths and limitations in capturing the nuanced expressions of sentiment within the Twitter dataset. This analysis provided a data-centric foundation for comparing and evaluating the two emotion categorization strategies.

Category	Common Tokens Count
Neutral	16376
Negative	15075
Positive	20873

TABLE IV
COMMON TOKENS COUNT FOR EACH CLASS IN CONSTRUCTION A & B

J. Common Frequent Words for Construction A & B

In our project, we delved into a comprehensive analysis of common frequent words within the sentiment categories, specifically comparing "construction A" and "construction B," which represent two different approaches to emotion categorization. This critical step aimed to reveal the overlap and shared language between these alternative classification methods.

We adopted a meticulous approach to achieve this objective, beginning with the creation of word clouds and bar charts displaying the top 20 most frequent words for each sentiment category. These visualizations provided a vivid representation of the language most strongly associated with each emotional state in both "construction A" and "construction B." To perform a quantitative analysis of word overlap, we developed a custom function to extract the top N most frequent words for each category. This allowed us to create dictionaries of top words for both "construction A" and "construction B." Furthermore, we calculated the number of common frequent words for each class in both constructions, employing Jaccard Similarity to assess the intersection of top words. This process provided a tangible measure of the degree of shared vocabulary within each emotional category. The results illuminated the extent to which the two construction methods captured similar words and expressions for a given emotion, shedding light on the consistency and divergence in categorizing emotions based on language characteristics. This comprehensive analysis was instrumental in evaluating the strengths and limitations of both approaches, contributing to a more profound understanding of their effectiveness in capturing sentiment nuances within the Twitter dataset.

Considering the 20 most frequent words in each data frame, we found the number of common frequent words in both data frames of each category within the 20 most frequent words using Jaccard Similarity. Jaccard similarity [8] is a measure of similarity between two sets. It is commonly used to compare the similarity or overlap between two sets, typically used in the context of set data or binary data. It is calculated by taking the number of shared labels between two sets and dividing it by the total number of unique labels in those sets. This measure is employed to assess how well a set of predicted labels for a sample matches the actual set of labels. The Jaccard similarity between two sets, A and B, is defined as the size of their intersection divided by the size of their union. Mathematically, it is expressed as:

$$J(A,B) = |A \bigcap B| / |A \bigcup B| \tag{1}$$

where $|A \bigcap B|$ represents the size of the intersection of sets A and B and $|A \bigcup B|$ represents the size of the union of sets A and B. The Jaccard similarity coefficient ranges from 0 to 1, where 0 indicates no similarity (completely dissimilar sets), and 1 represents perfect similarity (identical sets). The number of common frequent words and their corresponding Jaccard similarities in both data frames (Construction A & B) within the 20 most frequent words are represented in Table V.

The 20 most frequent words of one of the classes (Negative) for both construction A & B are represented in Figure 8 and 9 respectively.

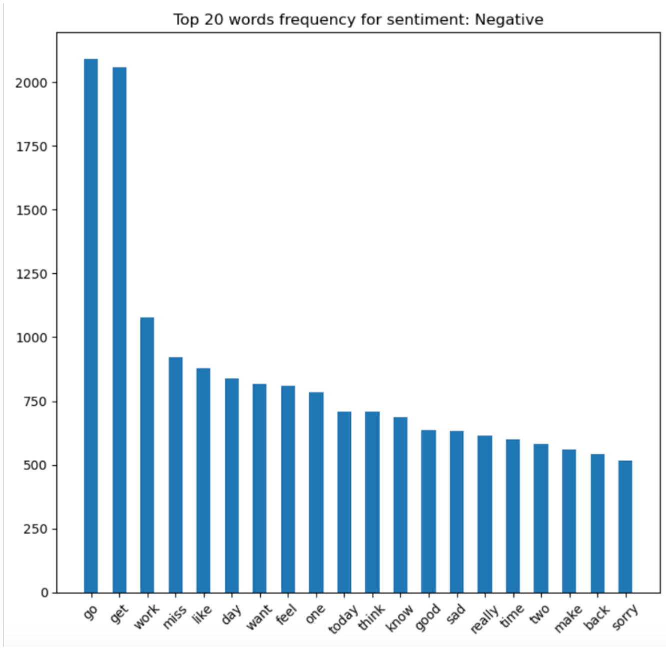

Fig. 8. Top 20 words frequency of sentiment ”Negative” for Construction A

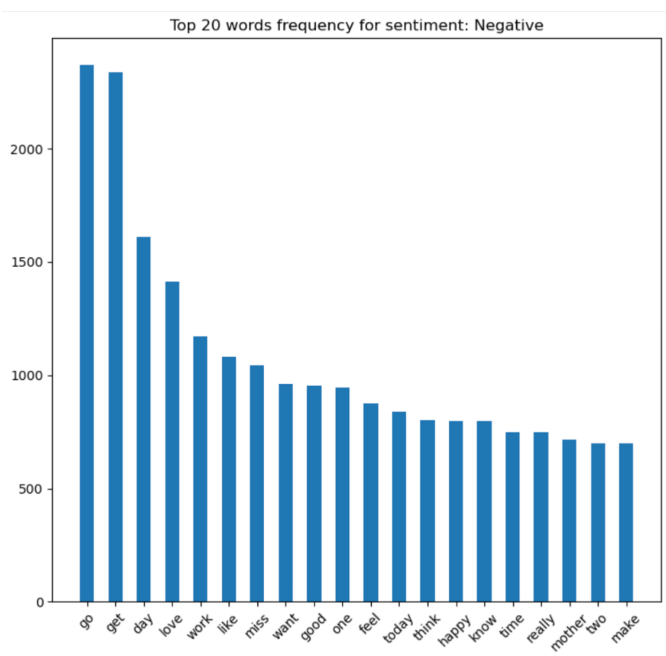

Fig. 9. Top 20 words frequency of sentiment ”Negative” for Construction B

Category	Common Frequent Tokens (among top 20)	Jaccard Similarity Score
Neutral	18	0.8182
Negative	17	0.7391
Positive	17	0.7391

TABLE V
COMMON FREQUENT TOKENS COUNT (AMONG 20) AND THEIR CORRESPONDING JACCARD SIMILARITY SCORE FOR EACH CLASS IN CONSTRUCTION A & B

K. *Unmatched Tokens using WorNet*

In this task, we conducted a comprehensive analysis of the linguistic content within the sentiment categories, specifically evaluating the presence of tokens in the WordNet lexical database[2]. This critical step allowed us to assess the linguistic validity and semantic coherence of the words used in tweets across both ”construction A” and ”construction B,” providing insights into the quality of the language employed to express emotions. We began by developing a custom Python script that checked the existence of each token in the WordNet database, a valuable linguistic resource. This process was essential in determining whether the words used in the dataset were recognized and validated by a well-established lexical reference, ensuring that the words were within the boundaries of the standard English language. The analysis included a calculation of the number of unmatched tokens and their corresponding percentage for each sentiment category. These metrics revealed the extent to which the words in the tweets did not have a recognized entry in WordNet, offering a quantitative measure of linguistic validity. The results were presented in two concise summary tables, one for each of the emotion categorization approaches. These tables depicted the number of unmatched tokens and the percentage of unmatched words for each sentiment category, thereby providing a detailed assessment of the linguistic quality and consistency within the constructed categories. This analysis contributed to a deeper understanding of the linguistic robustness of the emotions expressed in the Twitter dataset, shedding light on potential discrepancies and challenges associated with the choice of words in sentiment classification. Summary of the results for both Construction A & B is represented in Table VI and VII respectively.

Category	Unmatched Tokens	Percentage
Negative	22071	18.64
Neutral	14942	23.90
Positive	26130	21.58

TABLE VI
UNMATCHED TOKENS IN CONSTRUCTION A

L. *Stylometric Features using NRC Lexicon*

The NRC Word-Emotion Association lexicon[3] is a valuable resource in the field of natural language processing and sentiment analysis. This lexicon contains a comprehensive list

[2]https://wordnet.princeton.edu/download

[3]https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

Category	Unmatched Tokens	Percentage
Negative	20067	23.34
Neutral	11194	20.92
Positive	31882	19.61

TABLE VII
UNMATCHED TOKENS IN CONSTRUCTION B

of English words and their associated emotions or sentiments. Each word is tagged with one or more emotional categories, such as trust, fear, negative, sadness, anger, surprise, positive, disgust, joy, and anticipation. These associations are based on human judgments and can be used to determine the emotional tone or sentiment expressed in a piece of text. Researchers and analysts often use the NRC lexicon to assess and categorize the emotional content of textual data. It allows for the quantification of emotional expression within text, which can be beneficial in applications like sentiment analysis, social media monitoring, and content classification. By matching words in a text to their associated emotions in the lexicon, one can gauge the emotional context of the text and gain insights into the author's sentiments or the overall mood of the content.

The original and preprocessed data frames (Construction A & B) contain texts and corresponding sentiment labels. The primary objective is to assess the relationship between the sentiment labels and the presence of specific emotional categories from the NRC Word-Emotion Association lexicon. The task leverages the NRC Word-Emotion Association lexicon to explore the relationship between sentiment labels and specific emotional categories. The task's main steps include tokenizing the text, identifying emotional categories associated with individual tokens, and counting the occurrences of each emotional category within each record. This information is organized into two DataFrames: one containing emotional category counts and another containing labeling information, which indicates whether a specific sentiment label corresponds to an emotional category. These DataFrames are saved as CSV files for further analysis. Overall, the script facilitates a deeper understanding of the emotional content within the text data and how it aligns with sentiment labels, offering insights into sentiment analysis and emotional analysis applications. We also calculated the Pearson correlation coefficient and p-value for each emotional category in the NRC lexicon concerning both the labeling information and the counts of emotional categories within the texts. This analysis yields insights into the relationship between the emotional content of text data and the assigned sentiment labels, contributing to sentiment analysis and emotional analysis tasks. Table VIII, IX, X represent the Pearson Correlation and P-Value for the original twitter dataset, Construction A and Construction B respectively.

M. Category Similarity for Construction A & B

In this task, we embarked on a comprehensive analysis of category similarity, focusing on both "construction A" and "construction B" approaches to emotion categorization. This step aimed to quantify the resemblance between the sentiment categories by measuring the cosine similarity of their associated tweets at the record level.

Category	Pearson Correlation	P-Value
trust	-	-
fear	-	-
negative	-	-
sadness	0.094445	6.376317e-80
anger	0.017885	3.474314e-04
surprise	0.005388	2.812191e-01
positive	-	-
disgust	-	-
joy	-	-
anticipation	-	-

TABLE VIII
MAPPING BETWEEN NRC LEXICON AND TWITTER DATASET

Category	Pearson Correlation	P-Value
trust	-	-
fear	-	-
negative	0.217856	0.0
sadness	-	-
anger	-	-
surprise	-	-
positive	0.234431	0.0
disgust	-	-
joy	-	-
anticipation	-	-

TABLE IX
MAPPING BETWEEN NRC LEXICON AND CONSTRUCTION A

Category	Pearson Correlation	P-Value
trust	-	-
fear	-	-
negative	0.194550	0.0
sadness	-	-
anger	-	-
surprise	-	-
positive	0.131184	5.151647e-153
disgust	-	-
joy	-	-
anticipation	-	-

TABLE X
MAPPING BETWEEN NRC LEXICON AND CONSTRUCTION B

To facilitate this analysis, we employed pre-trained Word2Vec embeddings, a state-of-the-art word representation model. The procedure began with preprocessing the dataset, which involved tokenization and the removal of stopwords, ensuring that the text data was suitably structured for embedding calculations. We loaded a pre-trained Word2Vec model, specifically the GoogleNews vectors, and proceeded to compute record-based embeddings for the tweets within each sentiment category. The record embeddings represented the collective semantic content of the tweets, enabling us to capture the essence of the emotions conveyed in the language used. We then derived category-level embeddings by averaging the record embeddings within each sentiment category, effectively representing the essence of each emotional state. These embeddings served as a compact representation of the semantics associated with each category. The final step involved the calculation of cosine similarity between the category embeddings, producing a similarity matrix that quantified the relatedness between sentiment categories. This matrix provided a quantitative measure of how closely related the emotions were, allowing us to assess the similarity structure for both "construction A" and "construction B." The results were presented in the form of similarity matrices, shedding light on the degree of similarity and distinction between the emotion categories based on record-based semantic content. This analysis contributed to a deeper understanding of how well each categorization method captured the inherent sentiment patterns within the Twitter dataset, offering valuable insights into the efficacy of the two construction approaches. The record-based category-wise similarity scores for both Construction A and B are illustrated in Table XII and **??** respectively.

	Neutral	Negative	Positive
Neutral	1.0	0.9907	0.9842
Negative	0.9907	1.0	0.9786
Positive	0.9842	0.9786	1.0

TABLE XI
CATEGORY-WISE SIMILARITY SCORE USING WORD2VEC FOR CONSTRUCTION A

	Neutral	Negative	Positive
Neutral	1.0	0.9930	0.9857
Negative	0.9930	1.0	0.9860
Positive	0.9856	0.9860	1.0

TABLE XII
CATEGORY-WISE SIMILARITY SCORE USING WORD2VEC FOR CONSTRUCTION B

We have also applied the same method by using different types of embedding techniques such as GloVe and FastText. However, this time we have got different types of similarity scores as compared to word2Vec. FastText [9] is an open-source, free, lightweight library that allows users to learn text representations and perform text classification tasks efficiently. It was developed by Facebook's AI Research (FAIR). FastText is known for its ability to generate word embeddings (dense vector representations) for words and subwords, and it can be used for various natural language processing (NLP) tasks, including text-based emotion classification. The first step is to train word embeddings on the text data. FastText can learn word representations based on the context in which words appear. This involves learning vector representations for words and subwords (character-level n-grams). Based on the dataset, we used the trained FastText model to convert the text data into word embeddings. For each text sample, we can average the word embeddings of the words in the text to obtain a fixed-length vector representation.

	Neutral	Negative	Positive
Neutral	1.0	0.9893	0.9895
Negative	0.9893	1.0	0.9861
Positive	0.9895	0.9861	1.0

TABLE XIII
CATEGORY-WISE SIMILARITY SCORE USING FASTTEXT FOR CONSTRUCTION A

GloVe [11] (Global Vectors for Word Representation) is another popular word embedding method, similar to FastText. It can be used in emotion text classification tasks to capture semantic relationships between words and improve the performance of text classification models. We started by obtaining pre-trained GloVe word embeddings. GloVe provides pre-trained embeddings trained on large text corpora. We downloaded these pre-trained embeddings from the GloVe website or used Python libraries like SpaCy or Gensim to load them into the model. We created an embedding layer in the machine learning and neural network architectures to map words to their corresponding GloVe embeddings. This layer was used to convert text data into vector representations. Using pre-trained GloVe embeddings helped capture the semantic relationships between words and improve the model's ability to understand the emotional context within the text. The choice of architecture and hyperparameters depends on the specific characteristics of the dataset and the nature of the emotion classification task. There are pre-trained vectors available for both GloVe and FastText. We trained both embedding techniques with window size 5 and the embedding dimension was the length of the total feature vectors. We discovered that deep learning models perform well on pre-trained vectors using the dataset.

	Neutral	Negative	Positive
Neutral	1.0	0.9932	0.9915
Negative	0.9932	1.0	0.9923
Positive	0.9915	0.9923	1.0

TABLE XIV
CATEGORY-WISE SIMILARITY SCORE USING FASTTEXT FOR CONSTRUCTION B

	Neutral	Negative	Positive
Neutral	1.0	0.9936	0.9907
Negative	0.9936	1.0	0.9876
Positive	0.9907	0.9875	1.0

TABLE XV
CATEGORY-WISE SIMILARITY SCORE USING GLOVE FOR CONSTRUCTION A

	Neutral	Negative	Positive
Neutral	1.0	0.9956	0.9930
Negative	0.9956	1.0	0.9930
Positive	0.9930	0.9930	1.0

TABLE XVI
CATEGORY-WISE SIMILARITY SCORE USING GLOVE FOR CONSTRUCTION B

N. Machine Learning Techniques

After extracting features from the texts, these features are used to train our machine-learning model. Logistic Regression, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN) and Decision Tree algorithms are used for training purpose. For classification, logistic regression is employed where the objective of classification is to forecast the target class. When it comes to logistic multinomial regression, we have applied logistic regression methods to predict the target class (total 06), e.g. disgust, anger or joy. The purpose of using the Support Vector Machine algorithm in our research is to locate an N-dimensional hyper-plane, where, N is the number of features extracted from the documents. It determines a limit among the classes, which maximizes the margin, i.e. the distance among the data in these classes. For each of the potential values of that input variable in a decision tree, an inner node corresponds to one of the input variables and children are provided with edges for each. Each leaf is the target variable value given the input variable values represented by the root to the leaf path.

O. Performance Evaluation

To evaluate our proposed model, we used several evaluation metrics. The number of false positives, true positives, false negatives, and true negatives is reported in the confusion matrix. In our proposed model, if the number of texts are labeled and classified as happiness, it's true positive, TP and if not labeled and classified correctly, then it's true negative, TN. If the number of texts are labeled as happiness, but not classified correctly, then it's false negative, FN and if the no. of documents are not labeled correctly, but classified as happiness, it's false positive, FP. These metrics are used to calculate other evaluation measures such as, precision, recall, f1-score and accuracy.

The performance of traditional machine learning models using FastText on Construction A as shown in Table XVII is as follows, Support Vector Machine achieved a precision of 57.30%, indicating decent accuracy in making positive predictions. It also demonstrated an F1 score of 57.44% and an accuracy of 56.89% compared to other models. In summary, using FastText embeddings, the models' performance varied, with SVM and Logistic Regression achieving relatively better results in terms of precision, recall, F1 score, and accuracy, while Decision Tree and KNN lagged behind in performance metrics. Random Forest exhibited intermediate results.

Methods	Precision	Recall	F1-score	Accuracy (%)
Logistic Regression	0.5730	0.5523	0.5591	55.23
Decision Tree	0.4204	0.4107	0.4197	41.90
Random Forest	0.5381	0.5499	0.5259	54.99
SVM	0.5853	0.5689	0.5744	56.89
KNN	0.4727	0.4848	0.4739	48.47

TABLE XVII
PERFORMACE ANALYSIS FOR TRADITIONAL ML METHODS ON CONSTRUCTION A USING FASTTEXT

The performance of traditional machine learning models using FastText on Construction B as shown in Table XVIII is as follows, Support Vector Machine (SVM) achieved the highest precision of 56.81% compared to other models, indicating decent accuracy in making positive predictions. SVM also demonstrated an F1 score of 51.85% and Random Forest achieved the highest accuracy of 56.89% compared to other models. In summary, using FastText embeddings, the models' performance varied, with SVM, Random Forest and Logistic Regression achieving relatively better results in terms of precision, recall, F1 score, and accuracy, while Decision Tree and KNN lagged behind in performance metrics.

Methods	Precision	Recall	F1-score	Accuracy (%)
Logistic Regression	0.5605	0.4947	0.5080	49.47
Decision Tree	0.4387	0.4377	0.4382	43.76
Random Forest	0.5195	0.5435	0.4844	54.34
SVM	0.5681	0.5062	0.5185	50.62
KNN	0.4634	0.4761	0.4687	47.60

TABLE XVIII
PERFORMACE ANALYSIS FOR TRADITIONAL ML METHODS ON CONSTRUCTION B USING FASTTEXT

Significantly, comparing both results, we came to a conclusion that Construction A generally outperformed construction B in terms of precision, recall, F1 score, and accuracy for Support Vector Machine. This indicates that the manually crafted construction based modeling yielded more accurate emotion categorization.

IV. CONCLUSION

Our thorough investigation of the Twitter emotion dataset has yielded valuable insights across various dimensions. The dataset's absence of irrelevant data underscores its quality and appropriateness for further analysis, providing a solid foundation for our research. Preprocessing played a pivotal role in unveiling hidden patterns and sentiments within the data. By eliminating noise, implementing stopword removal, and addressing unusual characters, we enhanced the dataset's quality, paving the way for subsequent analyses. The common tokens between "construction A" and "construction B" underscore the linguistic consistency and shared expressions of emotions. Remarkably, the "Positive" category exhibits the highest overlap, suggesting a standardized way of conveying positive emotions. Our examination of WordNet revealed unmatched tokens, with the "Negative" category having the highest percentage of unmatched words. This underscores the importance of considering language quality in emotion analysis. We assessed the similarity between "construction A" and "construction B" using various word embeddings (Word2Vec, FastText, and GloVe). These analyses provided diverse perspectives on emotion categorization, enriching our understanding of their relationships. Our exploration of stylometric features and the NRC Lexicon expanded our analytical toolkit, enabling us to approach emotional content from multiple angles. This comprehensive approach bolsters the robustness of our sentiment analysis. We harnessed a range of machine learning algorithms (Logistic Regression, Decision Tree, Random Forest, SVM, and KNN) to classify emotions in both "construction A" and "construction B." The performance metrics demonstrate the capabilities of these algorithms, with support vector machine achieving the highest accuracy. Finally, the integration of state-of-the-art machine learning algorithms can significantly improve the precision and granularity of emotion classification.

One significant limitation in our analysis was the inability to implement BERT for assessing similarity between textual categories and records. BERT, a state-of-the-art transformer-based language model, has demonstrated remarkable capabilities in capturing semantic relationships and contextual nuances in text. Leveraging BERT could have offered a more advanced and context-aware approach to measuring similarity, potentially providing more accurate and nuanced results. Unfortunately, due to technical constraints, the complexities involved in fine-tuning and applying BERT to our specific dataset and the need for substantial computational resources, we were unable to harness the full potential of this cutting-edge technology. This limitation highlights the challenges and resource requirements often associated with implementing advanced deep learning models, underscoring the importance of considering practical constraints when selecting and applying state-of-the-art approaches in natural language processing tasks. Despite this limitation, the analysis using alternative techniques and approaches still provided valuable insights into the relationships between emotional categories and sentiment labels in our dataset.

In summary, our study has provided a multifaceted view of emotion detection in the Twitter dataset. We've uncovered language usage patterns, addressed language quality concerns and applied advanced techniques to enhance emotion categorization. These findings and contributions can be valuable for a wide range of applications, such as social media monitoring, customer feedback analysis, and market research. By better understanding how emotions are expressed and categorized in the digital world, we can develop more effective tools and strategies to engage with users and track public opinion.

References

[1] Tahani Almanie et al. "Saudi mood: a real-time informative tool for visualizing emotions in Saudi Arabia using twitter". In: *2018 21st Saudi Computer Society National Computer Conference (NCC)*. IEEE. 2018, pp. 1–6.

[2] Fahad Mazaed Alotaibi. "Classifying text-based emotions using logistic regression". In: (2019).

[3] Michelle Annett and Grzegorz Kondrak. "A comparison of sentiment analysis techniques: Polarizing movie blogs". In: *Advances in Artificial Intelligence: 21st Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2008 Windsor, Canada, May 28-30, 2008 Proceedings 21*. Springer. 2008, pp. 25–35.

[4] Thomas Carpenter and Thomas Way. "Tracking sentiment analysis through Twitter". In: *Proceedings of the international conference on information and knowledge engineering (IKE)*. The Steering Committee of The World Congress in Computer Science, Computer ... 2012, p. 1.

[5] Paul Ekman et al. "Basic emotions". In: *Handbook of cognition and emotion* 98.45-60 (1999), p. 16.

[6] Pollyanna Gonçalves et al. "Comparing and combining sentiment analysis methods". In: *Proceedings of the first ACM conference on Online social networks*. 2013, pp. 27–38.

[7] Maryam Hasan, Elke Rundensteiner, and Emmanuel Agu. "Automatic emotion detection in text streams by analyzing twitter data". In: *International Journal of Data Science and Analytics* 7 (2019), pp. 35–51.

[8] GI Ivchenko and SA Honov. "On the jaccard similarity test". In: *Journal of Mathematical Sciences* 88 (1998), pp. 789–794.

[9] Armand Joulin et al. "FastText.zip: Compressing text classification models". In: *arXiv preprint arXiv:1612.03651* (2016).

[10] Alexander Pak, Patrick Paroubek, et al. "Twitter as a corpus for sentiment analysis and opinion mining." In: *LREc*. Vol. 10. 2010. 2010, pp. 1320–1326.

[11] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[12] Marco Polignano et al. "A comparison of word-embeddings in emotion detection from text using bilstm, cnn and self-attention". In: *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*. 2019, pp. 63–68.

[13] Tapasy Rabeya et al. "A survey on emotion detection: A lexicon based backtracking approach for detecting emotion from Bengali text". In: *2017 20th international conference of computer and information technology (ICCIT)*. IEEE. 2017, pp. 1–7.

[14] Armin Seyeditabari, Narges Tabari, and Wlodek Zadrozny. "Emotion detection in text: a review". In: *arXiv preprint arXiv:1806.00674* (2018).

[15] Lovejit Singh, Sarbjeet Singh, and Naveen Aggarwal. "Two-stage text feature selection method for human emotion recognition". In: *Proceedings of 2nd International Conference on Communication, Computing and Networking: ICCCN 2018, NITTTR Chandigarh, India*. Springer. 2018, pp. 531–538.

[16] Maite Taboada et al. "Lexicon-based methods for sentiment analysis". In: *Computational linguistics* 37.2 (2011), pp. 267–307.

[17] Nafis Irtiza Tripto and Mohammed Eunus Ali. "Detecting multilabel sentiment and emotions from bangla youtube comments". In: *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*. IEEE. 2018, pp. 1–6.

[18] Binita Verma and Ramjeevan Singh Thakur. "Sentiment analysis using lexicon and machine learning-based approaches: A survey". In: *Proceedings of International Conference on Recent Advancement on Computer and Communication: ICRAC 2017*. Springer. 2018, pp. 441–447.

[19] Liza Wikarsa and Sherly Novianti Thahir. "A text mining application of emotion classifications of Twitter's users using Naive Bayes method". In: *2015 1st International Conference on Wireless and Telematics (ICWT)*. IEEE. 2015, pp. 1–6.