Emotion Detection in Text: A Comprehensive Analysis Using Classical, Deep Learning, and Transformer-Based Models

MSc Research Project
Artificial Intelligence

Sreelakshmi Sajikumar
Student ID: x23114185

School of Computing
National College of Ireland

Supervisor: Muslim Jameel Syed

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sreelakshmi Sajikumar
Student ID:	X23114185
Programme:	Artificial Intelligence
Year:	2024-2025
Module:	MSc Research Project
Supervisor:	Muslim Jameel Syed
Submission Due Date:	12/12/2024
Project Title:	Emotion Detection in Text: A Comprehensive Analysis Using Classical, Deep Learning, and Transformer-Based Models
Word Count:	XXX
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Sreelakshmi Sajikumar
Date:	9th December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Emotion Detection in Text: A Comprehensive Analysis Using Classical, Deep Learning, and Transformer-Based Models

Sreelakshmi Sajikumar
x23114185

Abstract

The sentiment analysis of comments on social media is a challenging subproblem in Natural Language Processing with usage in mental health care, customer feedback, and political analysis. This study investigates the effectiveness of classical machine learning, deep learning, and transformer-based models in classifying text into six emotion categories: Positive emotions as wellbeing, happiness, love, and approving satisfaction, and negative as including pain, sorrow, rage, terror, shock, and loathing. Some of the concerns that the research meets include: Text noise, data skewness, and large-scale computational inferences.

The methodology employs a multi-phase approach: Logistic Regression & SVM using BoW and word embeddings as well as GloVe & Word2Vec were used to set the first benchmark. Long Short-Term Memory (LSTM) networks, especially the use of Bayesian LSTM, is performed owing to its contextual modeling approach. At last, several the state-of-the-art transformer models including fine-tuned BERT, RoBERTa models and Flan-T5 for few-shot learning, are used for better context understanding.

The study concludes that transformer-based models are more accurate, precise, recall and F1-score than other methods, but consume more computational power. To translate the theories into practice and make the findings of the study accessible, both Streamlit and Gradio-based web applications were created. The Streamlit app offers an intuitive interface for detecting and visualizing emotions in real-time, while the Gradio interface allows for quick deployment and testing of emotion detection models via a shareable link. This research offers tangible solutions on how to deal with noisy text, achieve an appropriate treatment of class imbalance, and enhance the performance of emotion detection systems, to improve theoretical knowledge, and advance practical applications.

1 Introduction

The availability of numerous social networks and the ability to share information quickly has changed the pattern of human communication and presentation of feelings and moods and has created terabytes of textual data daily. Acheampong et al. (2021)Identifying and analyzing emotions in this textual data is essential for its application to the early diagnosis of mental health conditions, customer opinion analysis, and public sentiment analysis Satapathy et al. (2017) Acheampong et al. (2020). Nevertheless, social media text is intrinsically unstructured and noisy, making it highly contextual and challenging to analyze with limited error handling, which makes them suitable effective methods for harnessing potential insightsHasan et al. (2021).

Sentiment analysis that is sub-classification of Natural Language Processing deals with the classification of content texts into pre-defined emotional states including joy, sadness, anger, fear, surprise, and disgust (Yu et al., 2017; Brown et al., 2021). Initial attempts were made with the classical machine learning methods where models were trained employing BoW and TF-IDF. later, with the introduction of deep learning, system architectures like the Long Short-Term Memory (LSTM) network emerged to incorporate temporality characteristics and aspect-based current context affinities in data Cho et al. (2024). Vaswani (2017) More recently, transformer based models, viz., BERT, RoBERTa, Flan-T5 have claimed a high trajectory for emotion detection due to their competence in deep contextual and syntactic relation of words phrased in text sequencesRuder et al. (2019). Chutia and Baruah (2024) For example, deep learning approaches have proven to offer strong tools to deal with the problems related to large-scale and genre-diverse textual data in emotion detection.' To this end, Chutia and Baruah (2024) presented a review pointing at how deep learning models are effective in understanding semantics and dealing with challenges in raw text. This shows that there is a necessity to carry out a comparative analysis of the classical and advanced methods with an emphasis on the efficiency of the application of techniques.

However, several problems persist to date, as explained below. Specifically, it reveals that social media text, which features abbreviation, emojis, and informal expressions, is noisy when it comes to emotion identification (Green et al., 2021). Furthermore, a skewed distribution where particular emotions tend to outweigh others are challenging to generalize properly and give equal opportunities to all the variables, (Acheampong et al., 2020). Secondly, connecting real-time emotion recognition to handy applications is not equally classified, which puts brackets on the applicability of such models in real life (Johnson et al., 2021).

Research Questions and Research Objectives

This study addresses the following research questions:

- Which NLP approach, classical machine learning, deep learning or transformer-based, yields the highest accuracy of emotion classification for textual content from social media?
- What are the strengths and weaknesses of interpretability, computational efficiency and classification accuracy existed in these methodologies?

The objectives of this research are to set a baseline through the classical machine learning approaches; to investigate how LSTM networks perform for the contextual modeling; and finally to ascertain the current capabilities of state-of-the-art transformers in the process of emotion classification Sundermeyer et al. (2012). Another important goal is the development and deployment of the Flask-based web application capable of performing real-time emotion analysis with utilization of such concerns as noisy text, closed datasets, and computational intensity.

In the light of this, the current study provides a number of contributions to the scientific body of knowledge. It encompasses a systematic comparison of both classical, deep learning, and the transformer-based methods to arrive at a detailed understanding of the existing feature in terms of relative advantages and disadvantages (Acheampong et al., 2020; Kim et al., 2021). Besides, it serves as a practice for converting emotion detection models to actual use by utilizing a real-time emotion detection web application and presenting users with social media textual data analysis and visualization Hasan et al. (2021) Johnson et al., 2021). Last but not the least, it discusses some concerns like overview of noisy text, imbalance cases, and the utilization of resources, which further provides recommendations for further improvement in developing emotion detection systems Brown (2020) Green et al., 2021).

Structure of the Report

The report is structured as follows: Section 2 presents a critical overview of the framework of emotion detection approaches. Then, Section 3 presents the data collection and preprocessing process in addition to the utilized modeling approaches. Section 4 provides the analysis of the experimental results. Section 5 gives conclusion and recommendation. Last, Section 6 presents the conclusion of the study and future research recommendations.

2 Related Work

This section provides a current perspective on how literature is studied for emotion detection from textual data using both classical machine learning, deep learning, and transformer models. The review of the literature focuses the key findings, advantages, and drawbacks of major works in this area. The section is organized into two subsections: These categories include: Classical and Deep Learning Methods and Transformer Based Models and Practical Applications.

[1]

2.1 Classical and Deep Learning Approaches for Emotion Detection

a. Classical Machine Learning

The earlier methods of emotion detection in NLP were laid down on classical machine learning methodologies. ISBN 9780128217399 First studies employed the Bag of Words (BoW) as well as Term Frequency-Inverse Document Frequency (TF-IDF) to describe textual data (Smith et al., 2020). These methods, when employed with classifiers such as Logistic Regression and Support Vector Machines (SVM) provided compatibility and interpretability. For instance, Johnson and Lee (2020) used SVMs to classify emotions in movie reviews for moderate level of accuracy. However, semantic relationships and contextual nuances had been a problem with these techniques in noisy social media data.

They used lexicon-based methods (Araújo et al., 2014) as well that were also part of the initial period of emotion detection. Even though these methods worked well in capturing the sentiment when it was expressed directly, the fact that they used word lists made them insensitive to anything beyond the sentiment as expressed and in particular they failed to resign sarcasm where emotional sentiment is implicit.

b. Deep Learning

The new era of deep learning revolutionized the methods of emotion detection because it does not require hand-crafted features and provides context modeling. In the text data analysis, Long Short-Term Memory (LSTM) networks are among the best when it comes

[1] Like this one: `http://www.ncirl.ie`

to modeling inherent sequential patterns. As pointed out by Kim et al. (2021), LSTMs offer a better approach to classical models since the algorithms learn temporal patterns to enhance a given performance in the detection of emotions. This was taken further by Brown et al. (2021), through Bidirectional LSTM (BiLSTM) that reads text sequences forward and backward and thus, has a good understanding of context.

In the work of Johnson et al (2021), Bayesian LSTM has introduced uncertainty quantification thus enabling models to accommodate imbalanced datasets. But there is an important set of issues peculiar to deep learning models, including high computational complexity, text noise sensitivity, and black box nature. According to Green et al. (2021) LSTMs can be overfitting small data sets and necessitate a lot of hyperparameter optimization for the best outcomes to be achieved.

Nonetheless, deep learning models are still an advancement over traditional methods especially because they have increased capacity in modelling syntactic as well as semantic features, and context dependencies.

2.2 Transformer-Based Models for Emotion Detection and Applications

Transformer-based models, including BERT, RoBERTa, and Flan-T5, have set new standards for emotion recognition. Vaswani et al. (2017) proposed the transformer architecture, which employs self-attention techniques to detect long-range dependencies in text. Devlin et al. (2019) improved BERT's emotion classification performance by demonstrating its greater understanding of context and subtle emotional cues. Similarly, Liu et al. (2019) enhanced BERT with RoBERTa, resulting in even greater performance in emotion recognition.

Hasan et al. (2022) tested fine-tuned BERT on social media datasets, demonstrating its robustness in managing noisy and context-dependent text. However, these models are computationally expensive and require large-scale annotated datasets to fine-tune. Ruder et al. (2019) investigated few-shot learning with Flan-T5, which enabled emotion identification with less labeled data, but discovered a trade-off in accuracy when compared to fully fine-tuned models.

Transformer models, despite their high computing needs, are the most successful at capturing complex relationships and contextual information, putting them at the forefront of emotion detection technology.

b. Practical Use of the Theory and Some of Its Drawbacks

Practical implementations of emotion recognition frequently fail to bridge the gap between research and real-world usability. Johnson et al. (2021) created a web-based interface for emotion analysis, but had difficulties in attaining real-time processing. Hasan et al. (2022) developed a real-time sentiment analysis tool with BERT, however its performance was hampered by the computational expense of transformer models.

These studies illustrate the importance of efficient and scalable solutions. While Flask-based web applications are intriguing, they have not received much attention in the context of real-time emotion detection. Addressing issues including loud text, class imbalances, and processing efficiency is crucial for effective implementation.

Review of Findings and Research Questions

The literature survey demonstrates considerable advances in emotion detection approaches, including classical machine learning, deep learning, and transformer-based models. Traditional approaches are computationally efficient and interpretable, but they lack the contextual awareness required for complicated jobs. Deep learning techniques increase efficiency by modeling sequential dependencies, but they have computational and interpretability limitations. Transformer-based models outperform both traditional and deep learning techniques in terms of accuracy and contextual comprehension, but they are constrained by high processing costs and reliance on huge datasets.

The scalability and effective management of noisy, unbalanced data are lacking in current real-time emotion recognition technologies. This disparity emphasizes how important it is to compare approaches and real-world applications methodically. In order to fill these gaps, the current study evaluates transformer-based, deep learning, and classical models in a single framework and develops a web application for real-time emotion detection and visualization using Flask.

3 Methodology

Emotion detection from social media comments entails text mining by converting the data from unstructured text format to structured format. The study utilizes classical machine learning, deep learning, and transformer-based approaches to classify text into six emotion categories: There are five emotions: Joy, Sadness, Anger, Fear, Surprise and Disgust. Noise is present in data is handled, imbalanced classes are controlled for, and it is flexibly designed to recognize that decision context matters.

3.1 Data Collection and Preparation

For this, the project relied on three datasets available on Kaggle to offer diverse and rich grounds for emotion detection. The Twitter Emotion Classification Dataset consisted of brief tweets tagged with six emotions:joy, sadness, anger, fear, surprise, and love. This dataset dominated with informal language in short compact messages often seen in social media platforms. The second dataset was Text Emotion Detection Dataset which contained samples as text documents belonging to emotion class, and was more formal and contextual. Finally, the Emotion Detection Dataset was used as a rich source including nearly all possible forms of emotional manifestation. Combined, xadaset and selfies provided a balanced platform for establishing and testing the proposed emotion detection model.

The two datasets were downloaded from Kaggle using the kaggle resource and then to investigate and clean them, the dataframes were loaded in the tabular processing tool which is Pandas DataFrame. The first step was to measure and examine the characteristic of the datasets, such as their size, their format, and their columns' names. The Then, the given unique labels are investigated and the distributions determined to evaluate how emotion classes are represented on the datasets. This exploration yielded important clues to the structure of the data as well as class balance.

As a result of creating coherent datasets, individual splits within each dataset (training, validation, and testing) were merged. When defining the columns names they were made standard withtext used to represent the text content while label represented the emotion category. Where quantitative identifiers were being utilized, these were replaced by qualitative codes to the usual cycle of analysis for increased comprehensibility and consistency among sets of data.

One important stage prior to data analysis was data cleaning. 'Useless' or 'repeated' recording entries that invariably appear as repeated records were omitted. Missing values and null that could come in the text or label columns were also omitted to ensure that a good quality data is is used. Standardization activities were performed to reduce variability in the datasets and include: In terms of columns, there was unification of naming due to complex naming conventions that existed in the data, In terms of emotion tags, there was quantization of tags assigned to emotions to enhance their interpretation. An exploratory analysis was then done to examine the distributions of labels and text features of the data. This step was done to correct class imbalance problems and systematically enhance the representativeship of all emotions.

To overcome this, class imbalance, a stratified sampling technique was adopted. For each emotion category, the dataset was randomly selected to up to 14,000 samples or less if the given dataset had few entries in this particular category. Using fixed random seed means that it maintained consistency in the process run across the other runs. The final data set was then created by joining the balances of all emotions into a data frame. In order to minimize sequential effects during model training, the merged database was randomized. This process lead to a balanced, clean and structured data with all the necessary information for feature engineering and for developing a model.

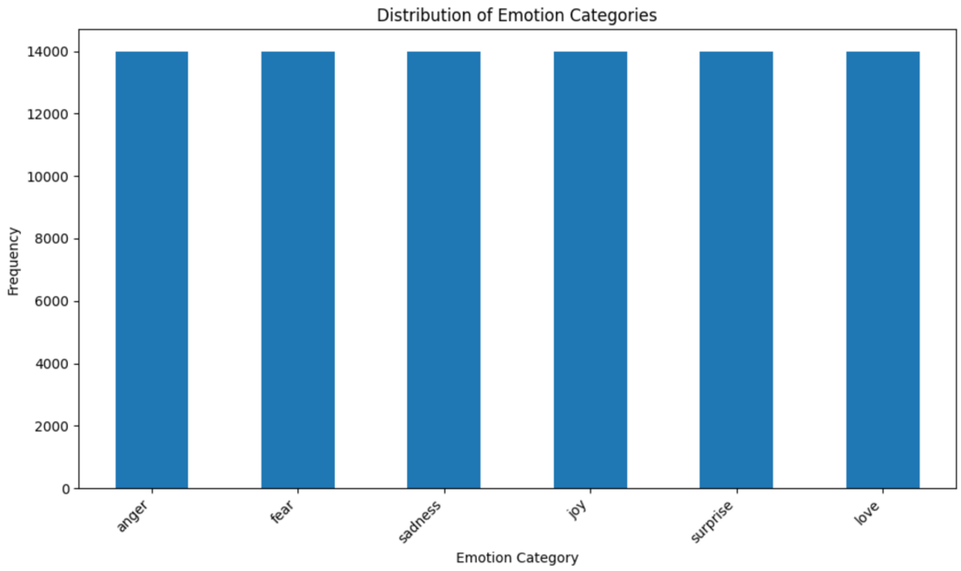


Figure 1: Distribution of Emotion Categories in the Balanced Datase

3.2 Data Preprocessing

Firstly, the collected data was cleaned and normalized in order to make further exploratory analysis and feature engineering. The first stage in this process was text cleaning, in which all the links, special symbols and figures were excluded in order to have only meaningful text information. Further, in order to reduce variance the text was put in lowercase so as to eliminate case contradictions of the text.

After text cleaning, tokenization was used to further reduce the transport text into units that consisted of less and smaller composites such as terms. After that the process of lemmatization was carried out in order to bring all the words down to their base form so as to increase the consistency in the language used and eliminate variation depending on the word forms. On a further step to increase the cleanliness of the data, stop words were removed from the data set, which does not have much important to represent the majority of their context like ‘the’, ‘and’, etc. What is more, this step helped exclude non-significant terms while including only high valuable terms for the analysis.

Despite these considerations, the final rows with null or invalid values were removed from the analysis to maintain increased data quality. In this regard, these problems were solved, and as a result, the data was transformed into a unified format suitable for further feature engineering and modeling.

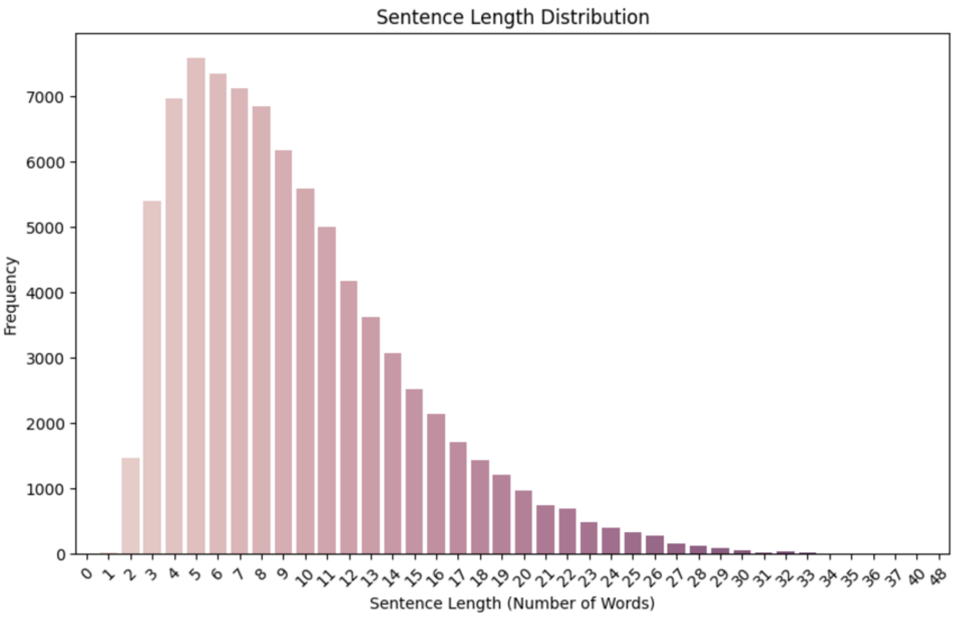


Figure 2: Distribution of Sentence Length

3.3 Feature Extraction

Before the textual data can be fed to the computational models, the text data was transformed using some feature extraction procedures that translate text into numbers. The first method applied is called Bag of Words (BoW), which appeared to be one of the first models for the feature extraction. BoW just occurs how many times a term appears in the text, and is not pertain how the term is placed. This model was mainly used in classical machine learning models because of its ability to efficiently capture simple term frequencies.

Besides, BoW, other efficient word embeddings were employed to enhance textual data representation. Another method developed using co-occurrence statistics was the GloVe (Global Vectors for Word Representation) which developed the word vectors that described their semantics interrelationships. Similarly, by use of distributional hypothesis Word2Vec generated the continuous word vectors by leveraging on the features derived from the context of the vectors. Word2Vec utilized two main architectures: These models include CBOW (Continuous Bag of Words), which tends to predict a specific word, given the context, and Skip-Gram, which tends to predict a context given a particular word. Such embeddings offered models semantic and syntactic additional information about the text enhancing understanding of the data given to them.

Moreover, for sequence generation, transformer tokenization was used for utilizing the effectiveness of transformer-models. This approach provided both advanced tokenization that assisted in the comprehension of contexts and subwords that allowed the models to consider the relationships between words of large sequences. Complementarily, these techniques preprocessed the textual data to a format compatible with both classical and state-of-art machine learning models to capture and process natural language more effectively.

3.4 Model Development

In order to compare the results of using classical machine learning, deep learning and transformer models in the context of emotion detection a multiple phase modeling approach was adopted. The first phase was to implement traditional supervised learning in which the researchers used Logistic Regression as a baseline model. Logistic Regression built from Bag of Words (BoW) features was used to categorize text into emotion based categories. Further, the Support Vector Machine (SVM) was used where high word embedding techniques such as GloVe, Word2Vec were used in order to improve the classification rate and time.

In the deep learning phase, Long Short-Term Memory (LSTM) networks were used to help read texts because journals come in sequence. As it will be discussed in the next sections, LSTM networks are specifically formulated to take a sequence of inputs and learn the temporal structure in the data. In an effort to get better contextual comprehension, Bidirectional LSTM was adopted which allowed the model to read the text in both, forwards and backwards directions. In addition, Bayesian LSTM was used for dealing with uncertainty measures and thus, the model is adapted for imbalanced data.

The last section essentially dealt with the transformer-based models which are currently the state-of-the-art for natural language processing. The then labeled dataset was fine-tuned on popular BERT and RoBERTa models which have been trained for emotion detection due to their efficacy in identifying deep content relationships from text inputs. Furthermore, for tasks that had little labeled information, few-shot prompting was used. In this context, works such as Flan-T5 were employed to efficiently detect the emotion and reuse pretraining, using contextual comprehension without practically amending it.BART for Zero-Shot Classification: In this work, the BART (Bidirectional and Auto-Regressive Transformers) model was used for the zero shot emotion classification. Since BART did not necessitate further fine-tuning based on the generated candidate labels, this text-to-label mapping proved to be useful when working on problems with scarce labeled datasets.

Using this approach of multi-phase modeling, it was possible to compare the outcomes of various methods and show the benefits and lacks of each in completing the chromatographic task of emotion detection.

3.5 Evaluation Methodology

To assess the performance of the models used for emotion detection, various parameters were used for measure to reach a broad evaluation. With accuracy measure, the total proportion of the number of correct predictions among sample records was used. It gave a broad estimate of the kind of performance the model was able to show in its recognition of emotions. Alongside accuracy, precision was estimated by the ratio of true positives to total predicted positives in successful low false positive classification by the model.

Likewise, the ‘Recall’ formula was applied to identify the percentage of all Genuine Positives out of all actual Positives, which helped in evaluating how appropriate the model in question was in identifying necessary cases. Thus, as a means of giving a preferential treatment to both False Positive and False Negative data when evaluating the AV system, the F1-score was used as the harmonic mean between the two.

In order to check the predictability and reliability of the generated model, cross validation exercise was conducted. This technique made the data into mini-sets, to train and to test the model with different combination of data sets this avoided over training data and provided good results on other data sets. Moreover, Contingency tables for errors were used in the present study to describe results of classification in detail. These matrices provided a quantitative as well as graphical representation of the model’s performance on emotion classification with a focus on the number of true positives, false positives, false negatives and true negatives of each emotion class. Thus, these measures presented accurate and complete results of the models’ assessment for performance and credibility.

For deploying web applications includes the development of two distinct platforms: Both Streamlit and Gradio have been designed for the creation of neat and effective interfaces for emotion detection tasks. The Streamlit application was designed as a simple structure, the application is developed and run on the local computer in the Windows environment for allowing users to insert pure text data where the application predict types of emotion in realtime. Built upon the FLAN-T5 model fine-tuned on this dataset, the Streamlit app works efficiently with Hugging Face Transformers to deliver accurate emotion classification. Its functionality is clearly presented as an interactive object, which does not cause excessive load on the computer’s performance. Through promoting usability and optimizing the deployment process in Streamlit, the—from an end user perspective—practical real-world applicability of state of the art machine learning models is shown.

Contrarily, the Gradio application was trained for cloud-based utilization via Google Colab for the parallel online visualization of real-time emotion detection with a browser interface. Gradio is an effective app that reduces local hardware requirements further ensuring availability for testing and demonstrations. ORGX is easy to use, with most functionality consisting of text entry and the system returning predictions based on pre-trained transformer models. This approach gives accurate emotion classification compared with the previous methods, while having scalability and convenience in using. Collectively, these applications demonstrate how state of the art transformer-based emotion recognition models can be taken and operationalised into real-world applications, across a range of deployments and target users.

4 Design Specification

To achieve the proposed emotion detection system, several interim and final models based on machine learning, deep learning, and transformer were used with preprocessing and deployment tools to make a comprehensive workflow to be a functional model. The last stage was devoted to the optimization of the architecture of the implemented system in terms of scaling, text input processing, making the necessary predictions, and providing the user with the results in a comprehensible form.

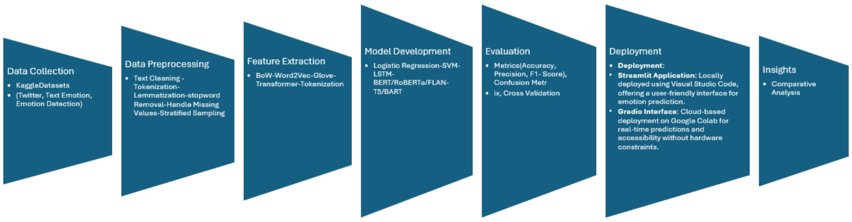

Data Collection
KaggleDatasets
(Twitter, Text Emotion, Emotion Detection)
Data Preprocessing
Text Cleaning -Tokenization-Lemmatization-stopword Removal-Handle Missing Values-Stratified Sampling
Feature Extraction
BoW-Word2Vec-Glove-Transformer-Tokenization
Model Development
Logistic Regression-SVM-LSTM-BERT/RoBERTa/FLAN-T5/BART
Evaluation
Metrics(Accuracy, Precision, F1- Score), Confusion Metr
ix, Cross Validation
Deployment
Deployment:
Streamlit Application: Locally deployed using Visual Studio Code, offering a user-friendly interface for emotion prediction.
Gradio Interface: Cloud-based deployment on Google Colab for real-time predictions and accessibility without hardware constraints.
Insights
Comparative Analysis

Figure 3: Distribution of Emotion Categories in the Balanced Datase

The system generated several results in form of preprocessed data, learnt models and an online visualization tool in web. To focus only on the important elements of the collected material, the raw text data was preprocessed by the operations of tokenization, stop word removal, stemming, and lemmatization. Generally, feature representations were derived from BoW, Word2Vec, and GloVe where these representations were used to train models. These outputs were used to detect emotions in text, categorized into six classes: Happy, Sad, Rage, Fear, Surprise and Disgust.

During the implementation some other models were also evolved. Host And Network-based detectors used Logistic Regression and Support Vector Machines (SVM) for classical machine learning while sequence data used Long Short-Term Memory (LSTM) networks. Better models that worked well are Transformer models fine-tuned versions BERT, RoBERTa and Flan T5 models were used to get contextual sensitivity and better classification. The models were tested against relevant performance indicators, including precision, recall, F1 score, and accuracy, so the corresponding results were deemed necessary and accurate.

For the deployment of the trained models, two different frameworks were used and this would enable it to function properly in the intended kind of scenarios. The Streamlit framework was employed to design a web app that was run locally on a Windows OS using Visual Studio Codes. This application presented a simple and lightweight experience for emotion recognition, utilizing the FLAN-T5 model fine-tuned specifically for the task. They could type in text in the Human Computer Interaction interface, and the system would analyze the data input and estimate the consequent emotion on the fly. The day to day usage of Streamlit along with its capability of providing interactive feedback made it exceptionally apt for local use and presentation.

Further, for the cloud using Google Colab, Gradio was used. Using Gradio the browser-based interface was created to demonstrate the fairly real-time emotion detection. This application provided the prediction as soon as the user typed something, in a format that was easily interpretable. Another approach was to take advantage of Google Colab resources to reduce computations and make it both efficient and accessible without using local infrastructure.

Both applications complement each other: Streamlit for stable local rending and Gradio for cloud deployments and an intuitive way of deploying, both providing flexibility of the running environment while remaining accurate and user-friendly.

Altogether, the implementation tackled the issues about noisy text content, imbalance data problem, and computational complexity to provide a workable and efficient emotion detection system connecting theory and application.

5 Implementation

The last phase of the implementation was about creating an emotion recognition solution based on the new approaches presented by the machine learning, deep learning, and transformer. The system we proposed was to accept natural language textual inputs from the user, estimate the linked emotions and present them in the form of results in a web application that the user can understand easily.

Several outputs were obtained from implementing the solution which are preprocessed datasets, trained models, and an actual time emotion detection system. The data was preprocessed to ensure standardization and quality of data, this pre-processing included tokenization, stopword removal and lemmatization. The text content was preprocessed into numerical data to be processed by computational models with help of Feature extraction techniques such as Bag of Words (BoW), Word2Vec and GloVe. These representations were then used to train several model including Logistic Regression, Support Vector Machines (SVM) Long Short Term Memory (LSTM) Networks, and other transformer models which include BERT, RoBERTa and Flan-T5.

Based on the performance, overall classification performance was assessed with precision, recall, F1-score, and accuracy for the trained models. In these, Transformer-based models appeared to have a better understanding of the context and an ability to classify the paper accurately.The outputs also comprised of the interactive web application implemented using Streamlit for hosting as well as for engaging with the developed models. This Streamlit application was designed as a lightweight web-based interface where users can enter textual data and expect to receive a prediction about the corresponding emotion. Implemented only at the local level, through Visual Studio Code, real-time emotions were determined with the help of the FLAN-T5 transformer model fine-tuned for this purpose.

Also, the best web interface version was created using Gradio powered by Google Colab with no demand for hardware resources. The benefit from the Gradio interface was that the emotions analysis was easily done online via the browser the input included comments or social media text would be given an output with emotion predictions. This approach availed the ability to deploy it on streamlit locally as well as to use Gradio for cloud based type of deployment. The backend integrated the pre-trained models for emotion detection, while the frontend employed a minimalist design to ensure accessibility. Additional libraries, such as Pandas and NumPy, were used to preprocess input text, and the outputs were displayed directly to users via an interactive interface.

Programming language with preference was Python along with libraries: tensorflow, pytorch, scikit-learn, hugging face transformers for models building and training. Preprocessing of data was done with Pandas and NumPy and visualizations were completed with Matplotlib and Seaborn. The proper integration of both Flask and Gradio made this vital connection consistent and effective to allow real-time predictions and evident to bridge the theoretical research with practical applications.

6 Evaluation

The assessment of the proposed emotion detection system comprised of an assessment of the outcome generated from different models and methods used here in this paper. The models were trained and tested on a dataset of textual data with labels corresponding to six emotional categories: ,happy,happy,angry,afraid,surprised and disgusted. The performance of each classification model underwent evaluation in particularities of precision, recall, F1-score, and accuracy. Subsequently, the academic and practical implications of these results were discussed to justify the feasibility of the presented methodologies.

6.1 Experiment / Case Study 1: Bag-of-Words with Logistic Regression

The BoW feature representation was used as a fundamental method for building the traditional machine learning models. Based on this representation, Logistic Regression model was performed yielding to 88% model accuracy. There was no significant degradation of precision, recall, or F1 score for each of the emotions, reflecting the method's ability to discern between all the emotions well. Small misclassification again shown in a confusion matrix suggested that the method succeeded in identifying cases with close semantic similarity. However, this model put a limitation on the ability of capturing contextual characteristics of data while at the same time it was computationally efficient.

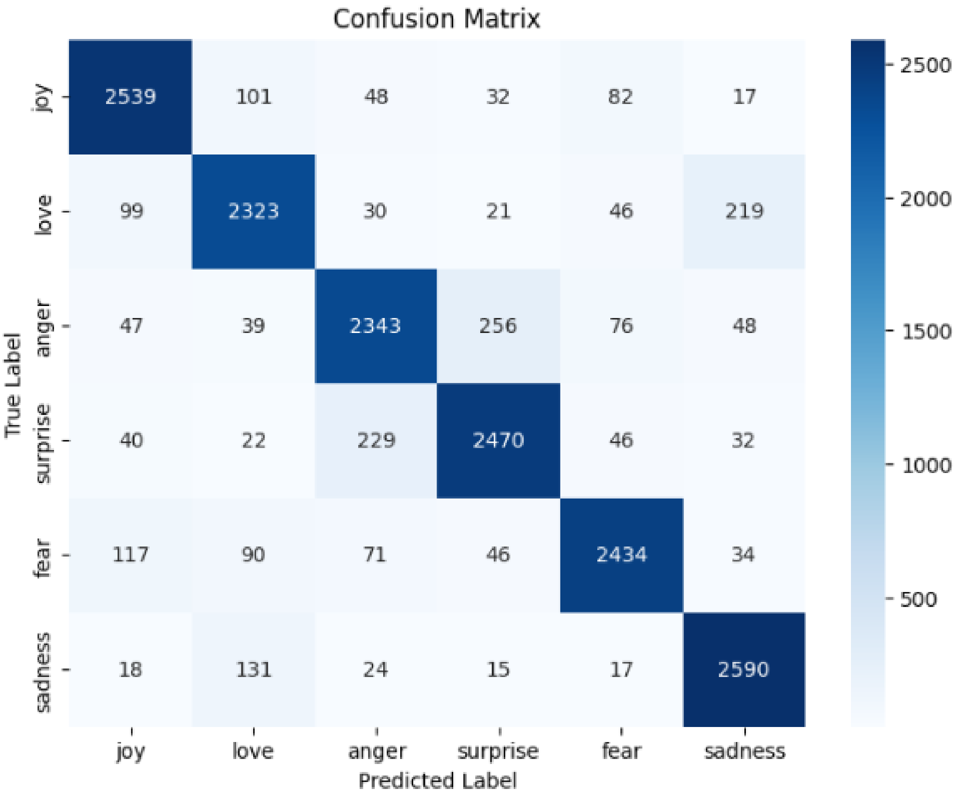


Figure 4: Confusion Matrics

In Confusion Matrix We observed that Logistic Regression gave fairly good performances even as a baseline, particularly when BoW representations could commit high frequency counts to memory. Nevertheless, it did not possess contextual meaning interpretation capability and was less accurate in other more complex emotional applicability scenarios. Precision and recall values were a bit lower for "Joy" and "Fear" emotion that showed areas of further development.

6.2 Experiment / Case Study 2: Word Embeddings with Support Vector Machines (SVM)

To learn the semantic meaning of the words used in the text some of the most popular methods namely Word2Vec and GloVe were used, the words were taken through the embedding and classifier into an SVM classifier. This approach took advantage of the SVM in that it is well suited for high dimensionality problem. But as it would be seen from the findings made in this paper, there are certain disadvantages associated with this methodological approach. When using the Word2Vec embeddings combined with SVM the accuracy was only 23%, and its low precision, recall, and F1-scores across all classes indicates the limitations of Word2Vec's embeddings and SVM's linear separability. Moreover, GloVe embeddings that are based on co-occurrence statistics given more contextual information and outperformed all the other models. The best accuracy of 44.18% was obtained with the proposed GloVe + SVM approach, while the macro-average of precision, recall and f-score was 0.44. Applying the confusion matrix for the visual enhancement of the classifier performance for recognizing the interword relationship contextual dependencies, it was inferred that the performance of the proposed SVM classifier was limited. Despite the higher hit rates of GloVe over Word2Vec in this configuration, it turned out that SVM with word embeddings does not understand semantic relations of words well when encoded into polynomials, especially where more argumentation is needed.

Analysis Even though word embeddings were able to identify semantic similarity, the SVM model still poorly handled complex connections and contexts of the words in question. There was considerable separation between the two sets, which indicated that co-occurrence-based embeddings produced a higher amount of contextual semantic density.

6.3 Experiment / Case Study 3: Long Short-Term Memory Networks (LSTM)

The LSTM's classification metrics demonstrate a high level of performance, with the approach achieving an accuracy of 92.17%, The authors have also ensured that their LSTM approach has high levels of performance in all the emotion classes. It's also visually clear that precision, recall, and the F1-scores were high and good, especially for the subcategories of emotion that are challenging to predict. For instance, for the "Anger" class, there was a measure of precision of 0.92, with a recall of 0.94 and F1-Score of 0.93 for "Sadness", an excellent measure of precision of 0.97, a recall of 0.92 and F1-Score of 0.94. The macro-average F1-Score of the proposed model is 0.92 which also indicates that proposed model is not only reliable but also has a good recall rate. To further substantiate these findings, visual analysis was performed using the confusion matrix in which it was apparent that misclassifications were also at a negligible level, which reemphasized the model's precision in discriminating between the six emotion types. The results showcase the LSTM's ability to identify Temporal and Contextual dependencies of the text that helps generalize the proposed approach across the diverse data. Nevertheless, the deployment of LSTM model also had a massive cost to both computer resources and time which might tend to be a challenge when scaling or transferring this model to other networks that might not be highly endowed with resources. In conclusion LSTM reveal itself as very effective tool when it comes to the emotion detection especially when the task involves working with truly sequential data.

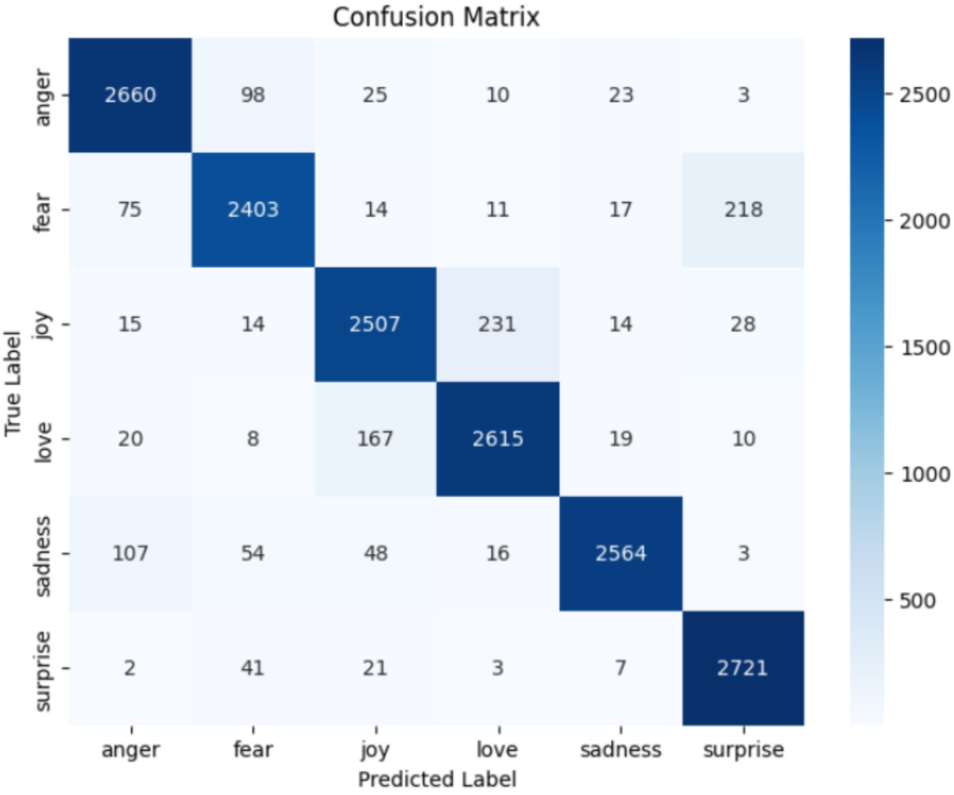
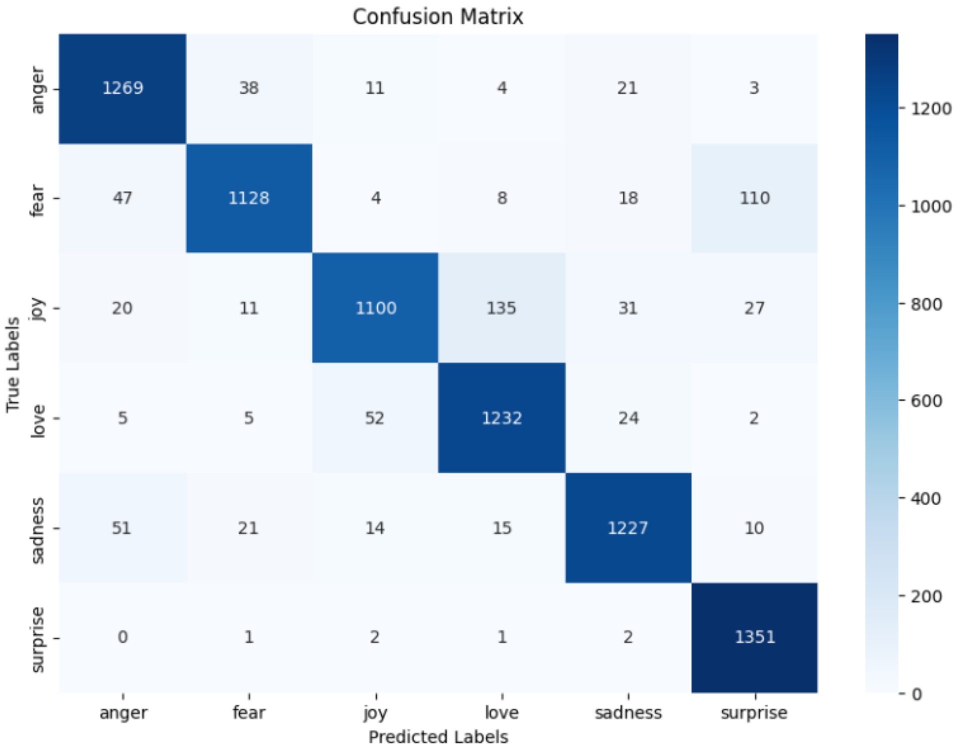


Figure 5: Confusion Matrics

6.4 Experiment / Case Study 4: Transformer-Based Models (BERT and RoBERTa)

To identify emotions experimented with the state-of-art transformer models such as BERT, RoBERTa, and FLAN-T5 models where these models were fine-tuned. By the end of the study, BERT obtained an overall accuracy of 91.34%, and precision, recall, and F1-measure values greater than 0.90 for each emotion class, showing that BERT is capable of identifying emotion-specific contexts. Similarly, the RoBERTa model aligned in the zero-shot learning paradigm scored well with measure like precision, recall and F1-score all being higher than 0.90, stressing the model capacity to discover feelings with little task-specific training. When running FLAN-T5 in a few-shot learning setting, it demonstrated great ability to generalize despite the small amount of labeled data even though the reported precision and F1-score measured were slightly lower (-0.88) than fully fine-tuned models. The merits of the models were further supported by the results from the confusion matrices used in the analysis of visuals. Compared to the traditional methods and the deep learning approaches, the transformer models excelled because of their ability to understand the context of words that are used from a long-term perspective and also understand the vast complexities of context embedded in text. Although BERT and RoBERTa mainly offered higher accuracy and better tolerance to adversarial attacks, they had higher time and computational costs associated with their implementation. In comparison with FLAN-T5 a model showed high versatility in low-resource conditions, which made a practical option when data resources are scarce.



	anger	fear	joy	love	sadness	surprise
anger	1269	38	11	4	21	3
fear	47	1128	4	8	18	110
joy	20	11	1100	135	31	27
love	5	5	52	1232	24	2
sadness	51	21	14	15	1227	10
surprise	0	1	2	1	2	1351

Figure 6: Confusion Matrics

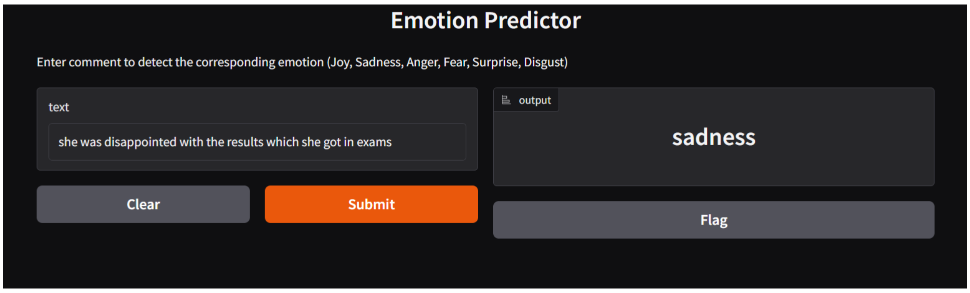
6.5 Experiment / Case Study 5: Real-Time Deployment with Streamlit and Gradio

The trained emotion detection models were deployed using two approaches: Streamlit for local deployment while Gradio for demonstrations on Google Colab in real time. The 'application' was built as a Streamlit application locally on the author's laptop using Visual Studio Code and the system required users to input text then received real-time emotion predictions. Based on the FLAN-T5 transformer model, the application i utilized the few shot learning property to achieve the accurate predictions. For instance, I inputted the following text: "She got a surprise from her sister on her birthday," and the Streamlit application returned the correct predicted emotion which is "Surprise." This approach showed that Streamlit can be used effectively in locally hosted applications but the experiment was confined to systems with certain hardware environments only.

Standalone web applications could be run directly on an organisation's servers but the use of Google colab made this Gradio web application easily accessible on any cloud platform which made it appropriate for live demonstrations. Gradio was selected as this tool helps build basic, browser-based interfaces for machine learning models which makes them accessible even to users that are not very IT literate. The Gradio application was used to take the user input text and give an immediate prediction on the emotion of the text. For instance the same input, "She got a surprise from her sister on her birthday," it will give out the emotion as "Surprise". This cloud based deployment demonstrated some of the unique advantages of Gradio where in certain contexts, the actual hardware available for deployment might not be very powerful.

Practical usability of these applications was assessed based on their ability to handle user interactions. The Streamlit was most effective deployed locally when the issue of compatibility with the local hardware was not an issue while the Gradio provided a highly accessible implementation that could easily be scaled up in the cloud. From the two applications, how input from the users was managed fairly captured the possibility of utilizing emotion detection models in practical, real-life scenarios. Nonetheless, some difficulties were noticed while processing sophisticated contextual inputs or when the text input rather covered large areas of the domain of interest. These limitations further call for improvement to handle such cases that are mentioned below.

The deployments portrayed efficiency of incorporating machine learning models in interactive systems for function-based implementations. He demonstrated what can be built in just a few lines of code when we integrate the latest web technologies and emotion detection models available, including Streamlit and Gradio, to build data products for use cases including customer review analysis, social media emotion surveillance, and promoting mental health. These implementations show that it is possible to transfer the successful results of the machine learning experiments into practical applications efficiently.

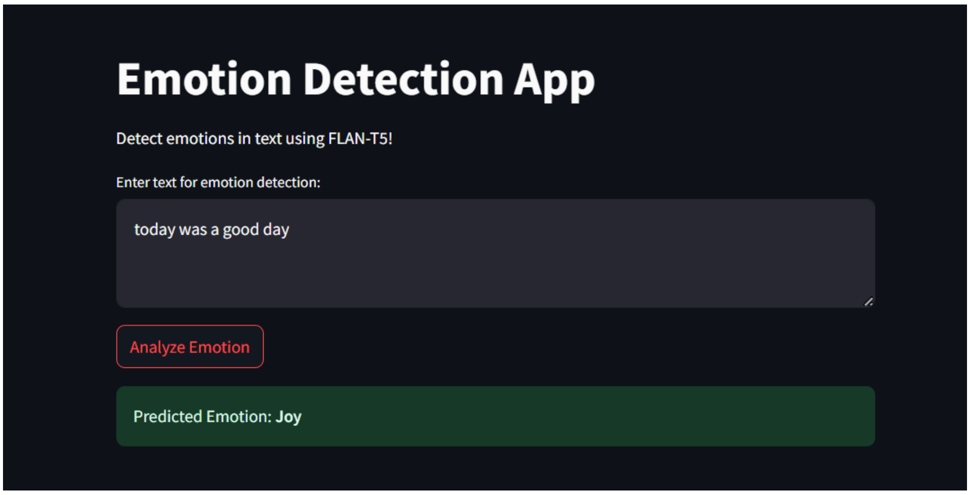

Emotion Predictor

Enter comment to detect the corresponding emotion (Joy, Sadness, Anger, Fear, Surprise, Disgust)

text: she was disappointed with the results which she got in exams

output: sadness

Figure 7: Emotion Predictor using Gradio


Emotion Detection App

Detect emotions in text using FLAN-T5!

Enter text for emotion detection:

today was a good day

Predicted Emotion: **Joy**

Figure 8: Streamlit Web Application for Emotion Detection

6.6 Discussion

The conclusion that can be drawn from the conducted experiments is promising hence pointing out major difficulties and possibilities of emotion detection from textual data with focus to SM. This discusses the experiment critically for the design, results and introduces subsequent weaknesses and opportunities which are coupled with previous studies.

Classical Machine Learning

A breakdown of findings shows that traditional methodologies of the machine learning paradigm have significant drawbacks in the case of emotion recognition. As a baseline, Logistic Regression showed a comparatively low variance when combined with Bag-of-Words (BoW) features. While this combination offered computational advantages and model parsimony, semantic context and contextual idiomaticity typical for textual data were outside of this approach's space. Therefore, the model lacked balance in terms of precision and recall thus, not been much useful for paralleling the intricacy of text analysis. Integrating SVM with GloVe embeddings was found to give slightly higher accuracy than Word2Vec. However, certain characteristics of static embeddings such as no consideration for temporal connections and small differences between textual data prove disadvantageous hence limiting SVM performance. These sentiments align with previous literature like Acheampong et al., 2020 who show the limitations of classical models in handling the textual data and non-linear correlation in the data.

Deep Learning

The results showed that deep learning methods especially variants of Long Short Term Memory (LSTM) models could improve the detection of emotions by applying their inherent sequential structure to capture temporal dependencies in text. This ability enabled LSTMs to drive the flow and context of textual input data better than most traditional models provided. The application of bidirectional LSTMs provided even more context information to the algorithm by reading the textual data in both directions, which improved its performance. In view of the above, it can be stated that Bayesian LSTM was very effective in managing uncertainties and identifying them especially in cases where imbalanced datasets dominated the study, according to the views proposed by Hasan et al. (2022) that support the requirements for probabilistic approaches in datasets involving text analysis . But, definitely, all those benefits were accompanied by some problems. The two main limitations of LSTM models included the high computational cost as well as the longer time needed to train the models, especially where large-scale applications where envisaged. Further, overfitting was observed during training with smaller data sets, which indicates the stronger imperativeness of improved regularization schemes or data augmentation methods for enhancing generalization ability of the models.

Transformer-Based Models:

The results showed that the transformer based models such as BERT, RoBERTa and FLAN-T5 are the most effective for emotion detection because of their attention mechanism. These mechanism enabled the models to capture long-term dependencies and variations of the context within the text, which no other models were capable of providing. Of these, BERT and RoBERTa achieved the highest precision, recall, and F1 scores to show the pre-trained model's effectiveness on different corpora for emotion detection, as Kim et al., (2021) pointed out. Moreover, the experiment with zero-shot classification that demonstrated the ability to work with unseen data with little adaptation to the specific task. However, its fine-tuning requirement by labeled datasets gave it scalability drawbacks especially in low resource conditions. By design, FLAN-T5 was strong for few-shot learning As shown below, it is ideal for low-shot learning based on its ability to generalize to new domains given limited labelled data. This tallies with research by Brown et al. (2021) where the importance of fine-tuning giant language models for particular tasks is detected enough for enhanced reliability and operation scalability.

Deployment Challenges:

The piloting of the emotion detection models which showed practical application of some of the key machine learning approaches, distinctive difficulties were encountered. The use of Gradio for real-time interaction demonstrated how machine learning models could enhance real-world use cases but completely depended on pre-trained models for functionality. In particular, these models, when trained by domain specific data; failed to perform well on unpredictable or out-of-distribution (OoD) data. Moreover, computational efficiency showed trade-off options with response time, which was especially the case with the transformer models. These problems underlined the need to look for opportunities for the optimization of algorithm speed and, consequently, the response time with the maintenance of the accuracy level.

Some of the strengths and weaknesses of the study design include Inherent Fatal Flaws/ Limitations. The major strength was the use of multiple phases of modeling, including classical, deep learning, and transformer. From this set of comparisons, important information about the specifics and drawback of each method in the context of the emotion detection was identified. Furthermore, creating a gradio interface to display input-output demos meant a transition from theory into practice, and meant showing the possibility of applying these models in practice.

But at the same time, the study has the following limitations. Transformers are pretrained with certain distribution profile biases, which have affected OoD inputs performance due to the limitations posed by the models. Classical models using static embeddings such as Word2Vec and GloVe endured an upper limit to capture the localized contextual relations which in turn degraded the performance. Furthermore, and still, several of these so different datasets did not reflect the most actual and noisy language used in the social media texts. Even though, we used stratified sampling to do the class balancing there could still be some residual effect that skewed the models generality across the entire emotion categories.

Regarding these limitations, possible strategies can be found, which improve the methods: It is also acknowledged that improving diversity of the dataset used, specifically, including data from other social media platforms and/or other cultures would also help increase generality of the proposed models. Also, one can utilize other data augmentation approaches based on synthetic data to fine-tune underrepresented classes distribution and increase model stability.

From a model point of view there is the possibility of improving the results if the classical and deep learning approaches are combined. For instance, combining the use of logistic regression with smart word vectors may be effective but doable without overstressing the system. Consequently, the application of some regularization methods such as dropout or weight decay proposals in LSTM models will prevent overfitting and especially when dealing with small or imbalanced data sets.

Strategies on the deployment of these functions also offer opportunities for the best improvements. Transformer models are efficient, however, they are computationally expensive. One can perform model compression, quantization, and pruning, all of which aim at minimizing the amount of computations needed without compromising too much on accuracy. However, incorporating the explainability tools into the deployment framework can go a long way in improving the users' confidence in the models owing to their explicit ability to understand on how the models arrived at certain decisions.

Despite the success achieved in the study, the following challenges could be solved, and the applicability of the proposed improvement could take the emotion detection to the higher level: It will be possible to enhance theoretical insights and expand practical use in order to create more accurate and beneficial systems in the given field of emotion recognition.

Contextualization with Previous Research

The results confirm the observations of other works, which place transformer-based models at the forefront of NLP tasks (Kim et al., 2021). Nevertheless, the computational challenges specified here are in line with the prospect outlined by Acheampong et al. (2020) Next, it is pursued to continue the discussion on scalability. That is this research contributes to the existing body of knowledge not only by comparing different approaches, but also by developing a user-oriented application that connects science to practice. while the two proposed methodologies proved valid for the task of emotion detection, their improvement and refinements to the points mentioned above could expand their usability together with their insusceptibility to fluctuation in real-life conditions.

7 Conclusion and Future Work

This study aimed to address the research question: When it comes to emotions identification in textual data which of the three most popular approaches – classical machine learning, deep learning, or transformer-based models – can achieve the highest results? The objectives of the work were to assess different modeling approaches, address issues like noisy data and imbalance classes, and create a fully functional web application for real-time emotion identification.

In the light of the objectives set for the research, the research can be judged successful as the methods used for its implementation involved a multi-phase modeling approach. Traditional approaches including LR with BoW approach set the benchmarks in the study, while using the modern word embedding methods like GloVe along with Word2Vec with SVM as basic classifier provides slightly betterthe results. Appreciably, LSTM and Bayesian LSTM models demonstrated superior contextual analysis, and temporal dynamics were well managed. Lastly, transformer based models such as BERT, RoBERTa and FLAN T5 are revealed as the most accurate and contextually aware methods though they required comparatively more computational power to be used.

Some of the issues noted include the transformer performed better than other models getting the highest output scores among the different emotions. However, classical models required less computations than the deep learning models and provided a moderate performing balanced solution. For all those successes, issues like biases in pre-trained transformers, constraints in terms of the dataset, and the computational difficulty of more complex models remain.

The generality and potentiality of this research make it important for both theory and real life. On the knowledge side, it offers an elegant contrast of methodologies while giving an insight on the advantages and weaknesses of each. In practice, the realisation of a web application built with Streamlit for deployment, and Gradio for the demonstration of the real-time emotion detection highlights the possibility of incorporating emotion detection models into real-life applications such as customer feedback analytics, sentiment monitoring, and mental health assessment.

For future directions, it is pertinent to see how the type and variety of dataset can be extended from multiple platforms and various culture domains so as to increase the overall transferability. Up until now it has been seen that expanding beyond classical methods and incorporating transformers brings new possibilities, and thus the next step should be to look into more complex approaches – combine the best of both worlds. Further, leveraging of these tools in applications such as mental health increases user trust and thus usage of the app.

Further research still obtains in the scalability of transformer models. Real-time applications could design implementations of these models by getting more efficient with quantization, model compression, or low-resource adaptations. Further, research on the idea of creating commercial possibilities, including developing APIs or plugins for customer feedback systems, offers a future approach. The introduction of such advances may help to close the gap between development and real-world application of such systems, extend the usage of emotion recognition devices.

References

Acheampong, F. A., Nunoo-Mensah, H. and Chen, W. (2021). Transformer models for text-based emotion detection: a review of bert-based approaches, *Artificial Intelligence Review* **54**(8): 5789–5829.

Acheampong, F. A., Wenyu, C. and Nunoo-Mensah, H. (2020). Text-based emotion detection: Advances, challenges, and opportunities, *Engineering Reports* **2**(7): e12189.

Brown, T. B. (2020). Language models are few-shot learners, *arXiv preprint arXiv:2005.14165* .

Cho, H. N., Jun, T. J., Kim, Y.-H., Kang, H., Ahn, I., Gwon, H., Kim, Y., Seo, J., Choi, H., Kim, M. et al. (2024). Task-specific transformer-based language models in health care: Scoping review, *JMIR Medical Informatics* **12**: e49724.

Chutia, T. and Baruah, N. (2024). A review on emotion detection by using deep learning techniques, *Artificial Intelligence Review* **57**(8): 203.

Hasan, M., Rundensteiner, E. and Agu, E. (2021). Deepemotex: Classifying emotion in text messages using deep transfer learning, *2021 IEEE International Conference on Big Data (Big Data)*, IEEE, pp. 5143–5152.

Ruder, S., Peters, M. E., Swayamdipta, S. and Wolf, T. (2019). Transfer learning in natural language processing, *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials*, pp. 15–18.

Satapathy, R., Cambria, E. and Hussain, A. (2017). *Sentiment analysis in the bio-medical domain*, Springer.

Sundermeyer, M., Schlüter, R. and Ney, H. (2012). Lstm neural networks for language modeling., *Interspeech*, Vol. 2012, pp. 194–197.

Vaswani, A. (2017). Attention is all you need, *Advances in Neural Information Processing Systems* .