Project Phase 2 Report On

Emosense

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Akshay Ajit (U2003018)

Crispin Mathew (U2003065)

David Johns Denny (U2003066)

Devananth H (U2003068)

Under the guidance of

Dr. Dhanya PM

Department of Computer Science and Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
April 2024

CERTIFICATE

This is to certify that the project report entitled ***"Emosense"*** *is a bonafide record of the work done by* ***Akshay Ajit (U2003018)***,***Crispin Mathew (U2003065)***, ***David Johns Denny (U2003066)***, ***Devananth H (U2003068)*** *submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2023-2024.*

Dr. Dhanya P M
Project Guide
Professor
Dept. of CSE
RSET

Dr. Sminu Izudheen
Project Coordinator
Professor
Dept. of CSE
RSET

Dr. Preetha K G
Head of the Department
Professor
Dept. of CSE
RSET

ACKNOWLEDGEMENT

We wish to express our sincere gratitude towards **Prof.(Dr). P. S. Sreejith**, Principal of RSET, and **Dr. Preetha K G**, Head of the Department of CSE for providing us with the opportunity to undertake our project, Emosense.

We are highly indebted to our project coordinator, **Dr. Sminu Izudheen**, Professor, CSE for her valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Dr. Dhanya P. M.** for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Akshay Ajit

Crispin Mathew

David Johns Denny

Devananth H

Abstract

This project is about making user experience with social media better, by developing the complete analysis tool that would parse textual data from different platforms like Instagram, Facebook or Reddit. By developing a custom browser extension, the user-generated content is captured in real time and can also be processed subsequently.

The project comprises two key components: Emotion Detection and Hate Speech Detection The Emotion Detection system uses sophisticated Natural Language Processing NLP techniques, such as BERT models to identify users' emotional state through their social media posts. This information is then used to develop a Monthly Mental State Report by which users are able to gain insights into their emotional health over time.

Further, the Hate Speech Detection system leverages machine learning algorithms to detect and mark words that may be harmful or offensive in user-generated content. This feature enhances positive and inclusive Internet environment.

The project goes beyond the analysis; it includes a pragmatic component suggesting specific tasks for users depending on their mental state reports. These recommendations are aimed at promoting positive interaction and enhancing the user's overall well-being.

In addition, it has incorporated a text summarization feature that allows for the extraction of critical information from long posts on social media. Users will be able to read content effectively and focus on specific details without being overloaded with information.

This project offers a comprehensive insight into users' emotional experiences on social media using state-of-the-art NLP and machine learning approaches. The combination of task recommendations and text summarization increases interaction among users, creating a pleasant atmosphere where people can enjoy using social media. The browser extension is a multipurpose tool that does not merely analyze but provides some improvements to the user's mental health and online activity.

Contents

List of Abbreviations

Acronym - Expansion

1. LSTM - Long Short-Term Memory
2. RNN - Recurrent Neural Network
3. NLP - Natural Language Processing
4. CNN - Convolutional Neural Network
5. SVM - Support Vector Machine

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In a rapidly evolving digital landscape, our project pioneers the integration of cutting-edge technology to create a versatile and robust platform—the Emotion and Hate Speech Detection Extension. This innovative endeavor seeks to address two critical aspects of online communication: understanding and interpreting human emotions, and curbing the dissemination of harmful content. Our Emotion Detection Extension stands as a testament to the fusion of artificial intelligence and emotional intelligence. Utilizing state-of-the-art machine learning models, the extension comprehensively analyzes textual content, voice inputs, and facial expressions to accurately decipher the user's emotional state. In parallel, our Hate Speech Detection Extension serves as a robust guardian against the proliferation of toxic content in the digital realm. Leveraging advanced natural language processing algorithms, the extension scans and evaluates textual data for signs of hate speech, discriminatory language, and offensive content. As a groundbreaking feature, both the Emotion and Hate Speech Detection functionalities seamlessly extend beyond conventional applications. Users can seamlessly integrate our extension into their preferred web browsers, enabling real-time analysis of content encountered online. Simultaneously, our dedicated website provides a comprehensive dashboard, allowing users to manage their preferences, view detailed analyses, and access insightful visualizations of their emotional patterns and content encounters.

1.2 Problem Definition

Design a powerful tool that can extract information from user's social media activities by harnessing the power of natural language processing (NLP) and machine learning techniques to generate comprehensive mental health reports and detect hate speech.

1.3 Scope and Motivation

The scope of the Emotion and Hate Speech Detection Extension is vast, transcending the boundaries of conventional online interactions. By seamlessly integrating with popular web browsers, the extension extends its reach to diverse digital spaces, including social media platforms, forums, and content-sharing websites. The real-time analysis of textual content, voice inputs, and facial expressions provides users with a holistic understanding of their emotional states during digital engagements. Furthermore, the extension's comprehensive hate speech detection capability contributes to a safer and more inclusive online environment by actively identifying and mitigating harmful content. The scope extends to empowering users with the tools to navigate the digital landscape responsibly, fostering empathy, and creating an atmosphere of respect in virtual spaces.

The motivation behind the Emotion and Hate Speech Detection Extension is rooted in a commitment to revolutionize online experiences. In an era dominated by digital communication, understanding and responding to user emotions play a pivotal role in enhancing the quality of interactions. The extension aims to empower individuals by providing insights into their emotional responses, fostering self-awareness, and enabling them to curate more personalized online experiences. Simultaneously, the project is driven by a strong sense of responsibility to combat the rising tide of hate speech and toxic content online. The motivation is to create a digital landscape where users can express themselves freely, without fear of encountering harmful expressions, ultimately contributing to a more harmonious and respectful online community. Through the integration of emotion detection and hate speech moderation, the project aspires to redefine the norms of digital communication and set a new standard for empathetic and secure online interactions.

1.4 Objectives

- Implement advanced emotion detection algorithms to interpret and respond to user emotions during online interactions.
- Enable users to effortlessly manage and customize the extension's settings through a dedicated website, offering a centralized hub for control and insights.

- Design the extension with a focus on inclusivity, recognizing and respecting the diversity of user expressions and emotions.
- Implement features that encourage positive online communication by providing users with tools to understand and manage their emotional impact on digital conversations.
- Raise awareness about responsible digital communication by providing users with clear insights into their online interactions.

1.5 Challenges

Developing the Emotion and Hate Speech Detection Extension confronts several formidable challenges. Crafting accurate emotion detection algorithms that interpret diverse digital inputs, addressing algorithmic complexity, and achieving adaptability across various web browsers are key technical hurdles. Ensuring ethical and culturally sensitive hate speech detection, balancing user privacy concerns with data collection, and meeting real-time processing demands are critical considerations. The extension must grapple with the challenge of training data quality to effectively recognize emotions and combat hate speech. Additionally, fostering user education, acceptance, and promoting responsible digital communication pose significant challenges. The dynamic nature of online content requires continuous updates to the hate speech detection mechanisms. Successfully navigating these challenges demands a holistic, multidisciplinary approach that encompasses machine learning expertise, user experience design, ethical considerations, and cultural sensitivity to deliver a robust and responsible extension.

1.6 Assumptions

The Emotion and Hate Speech Detection Extension project operates on the assumptions of user consent for data processing, availability of diverse training data, continuous user engagement for feedback and improvement, compatibility with major web browsers with allowance for variations, adaptability to the dynamic nature of online content, ethical use by users, responsible data handling in compliance with privacy regulations, and minimal impact on browsing performance. These foundational assumptions collectively underpin the development, implementation, and user acceptance of the extension, emphasizing the need for ethical user behavior, transparent communication, and ongoing collaboration to ensure its effectiveness in promoting positive online interactions and mitigating hate speech.

1.7 Societal / Industrial Relevance

The Emotion and Hate Speech Detection Extension project holds significant societal and industrial relevance by actively contributing to enhanced online safety and fostering a positive digital culture. Addressing issues of cyberbullying and harassment, the extension plays a pivotal role in supporting diversity and inclusion within the online sphere. This initiative aligns with corporate social responsibility goals, providing technology companies and platforms with an opportunity to demonstrate their commitment to ethical and responsible digital practices. The project not only contributes to legal compliance and risk mitigation but also positions itself as an educational tool, raising awareness about the consequences of hate speech and promoting responsible online behavior. As organizations adapt to evolving social norms, the extension becomes a crucial component in ensuring market competitiveness by offering advanced moderation tools that prioritize user safety and positive experiences. Its multifaceted impact encompasses media relations, educational initiatives, and proactive measures against emerging challenges, positioning it as a key player in shaping a safer and more inclusive digital landscape.

1.8 Organization of the Report

This report sheds light on the various methodologies by which the intended results of the project can be obtained through a thorough literature survey. It also elaborates on the proposed methodology and a well detailed functioning of the modules involved in the project. The division of labor and a well planned Gantt chart is also included.

Chapter 2

Literature Survey

2.1 BERT-CNN: A Deep Learning Model for Detecting Emotions from Text[1]

The base paper "BERT-CNN: A Deep Learning Model for Detecting Emotions from Text" introduces a comprehensive deep learning model, BERT-CNN, which comprises three key components: data preprocessing, the BERT base model, and a CNN classifier. The data preprocessing stage involves cleaning the text by removing noise and uninformative sections, followed by tokenization into word pieces. The BERT base model is then employed to generate contextual vector representations for each token in the text, leveraging a bidirectional transformer architecture with 12 layers and 12 heads of self-attention. Subsequently, the CNN classifier extracts high-level features from the BERT output using filters of various sizes, max-pooling, dropout mechanisms, and fully connected layers.

To validate the efficacy of the proposed model, extensive training and evaluation are conducted on two diverse datasets: the Semeval 2019 task 3 dataset and the ISEAR dataset. These datasets consist of texts labeled with various emotions, providing a robust testbed for emotion detection. The model's performance is benchmarked against several state-of-the-art models, and notably, BERT-CNN consistently achieves the highest accuracy and F1-score on both datasets. This highlights the model's superior ability to accurately identify and classify emotions in text, positioning it as a leading solution in the field of emotion detection and sentiment analysis.

2.1.1 Pre-processing data

The initial step in text classification, known as data preprocessing, is crucial for enhancing the efficiency of the classification process. This phase involves cleaning the text by eliminating noise data and uninformative sections, such as hashtags. The key objective is to refine the original text and optimize it for subsequent analysis. It also includes tokenization, which has the following processes:

- Normalizing URLs, emails, money, date, time, percentage, expressions, and phone numbers.
- Annotating all capitalized letters, censoring phrases, and words with emphasis.
- Annotating and reducing elongated and repeated words.
- Unpacking hashtags and contractions.

2.1.2 Bidirectional Encoder Representations Form Transformers (BERT)

BERT, a foundational element in Natural Language Processing (NLP), stands as the lowest layer language model. Through comprehensive pre-training on diverse corpora, BERT achieves unparalleled global and local feature representations for sequences. The network structure of BERT[4], depicted in Fig. 2.1, is rooted in the transformer architecture, featuring 12 layers that collectively enable robust contextual understanding.

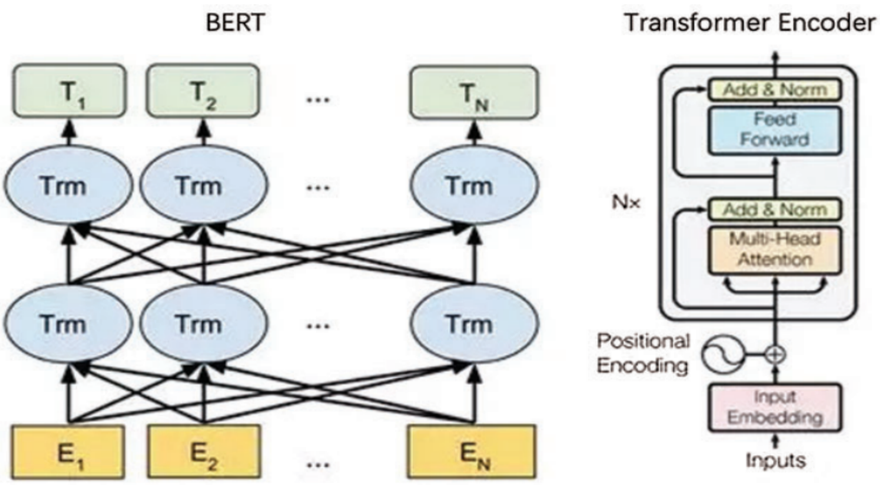


Figure 2.1: Architecture of the BERT model

Input Representation on BERT

- Token Embeddings: This layer transforms individual words into dense 768-dimensional vectors. The input text undergoes tokenization, and additional tokens ([CLS] and

[SEP]) are inserted to signify the beginning and end of tokenized phrases, aiding classification tasks.

- Segment Embeddings: Tailored for tasks involving pairs of input texts, segment embeddings facilitate BERT's ability to comprehend and distinguish between different segments within a sequence.
- Position Embeddings: Recognizing the importance of word order, positional embeddings are introduced to convey the position of words in a sentence. These embeddings are instrumental in overcoming the transformer's inherent limitation in capturing sequential information.

Encoder layer

The heart of BERT lies in its Encoder layer, comprised of 12 transformer blocks. Each block integrates 12 heads of self-attention, allowing the model to capture intricate relationships between tokens within a sequence. This self-attention mechanism is vital for understanding the internal structural nuances of the input text. The mechanism of multi-head attention involves matrices of queries (Q), keys (K), and values (V), enabling BERT to discern and learn token relationships effectively. Multiple linear projections facilitate parallelized computation, enhancing efficiency without compromising the model's depth.

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$

Figure 2.2: Scaled dot-product attention

$$MultiHead(Q, K, V) = Conc(Head_1, .Head_n)W^o$$

$$Head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Figure 2.3: Multihead attention

In addition to self-attention, each encoder layer incorporates a fully connected feed-forward network (FFN). This network operates independently on each position and is applied iteratively. The output from the FFN is subsequently passed to the next layer until reaching the final layer for classification.

$$FFN = \max(0, X^{W1} + b1)W2 + b2$$

Figure 2.4: FFN

2.1.3 Convolutional Neural Network (CNN)

The CNN structure comprises Convolutional layers, Max Pooling, and Fully Connected layers, as depicted in Fig. 2.5 . Each layer plays a crucial role in extracting and processing features from the BERT output.

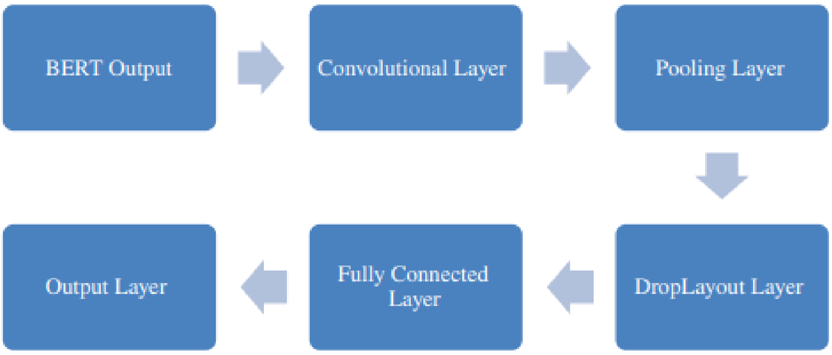


Figure 2.5: Architecture of the Convolutional Neural Network

Convolution layer

The Convolutional layer employs filters of different sizes to extract higher-level features from the input matrix. Filters, characterized by width (m) and height (h), are applied with a non-linear function (ReLU). Multiple filters with varying heights enhance feature coverage[5].

$$Sj = g(\omega \cdot [vi : vi + h - 1] + b)$$

Figure 2.6: Convolution layer

Pooling layer

Following the Convolutional layer, the Max-pooling layer minimizes and down-samples features in the feature map[6]. A max-pooling operation selects the highest value for each vector dimension, followed by dropout to mitigate overfitting.

$$h^p = MaxPooling \quad (h^c | h, w)$$

Figure 2.7: Pooling layer

Fully Connected and Output Layer

The high-level features extracted by the Convolutional layer are then forwarded to the Fully Connected layer. Here, the softmax function calculates the probability of each class.

$$\mathbf{p_i} = \frac{\mathbf{e^j}}{\sum^j \mathbf{e^j}}$$

Figure 2.8: Softmax

2.2 A BERT based dual-channel explainable text emotion recognition system[2]

This paper introduces a novel approach to multi-class text emotion recognition through a dual-channel system, offering a distinctive methodology for training and prediction explication. Leveraging a pre-trained BERT model, the paper extracts embedding vectors from input sentences, forming the foundation for the dual-channel network. This network comprises two channels, integrating a convolutional neural network (CNN) for feature extraction and a bidirectional long short-term memory (BiLSTM) network for capturing sequential information. The synergistic use of these channels aims to enhance the model's ability to comprehend both local features and broader contextual dependencies.

The outputs from the CNN and BiLSTM channels are concatenated and forwarded to the emotion classification module, where the system learns and projects emotion embeddings onto a hyperplane, effectively forming clusters. A key innovation lies in the development of a novel explainability technique, elucidating the training and prediction processes. This technique involves analyzing inter-cluster and intra-cluster distances, as well as the intersection of clusters across various emotion classes.

To validate the proposed methodology, the paper evaluates the system on four Text Emotion Recognition (TER) datasets, benchmarking its performance against existing approaches in terms of accuracy, precision, recall, and F1 score. Additionally, the paper provides a comprehensive qualitative and quantitative analysis of the emotion embedding clusters, offering insights into the system's ability to discern and represent various emotional states in textual data.

2.2.1 Methodology

The paper, titled "A BERT-based dual-channel explainable text emotion recognition system," outlines a comprehensive methodology for developing a novel deep-learning-based Text Emotion Recognition (TER) system. The approach integrates a dual-channel architecture, harnessing Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) components for multi-class text emotion recognition. A key emphasis is placed on the use of pre-trained BERT embeddings to capture rich contextual information and semantic meaning from input sentences.

The methodology begins with the employment of the BERT model to extract language-agnostic vector representations (embeddings) of input sentences. These embeddings serve as a foundation for the dual-channel architecture, allowing the system to effectively model sequential and spatial information in the text. The BERT model is fine-tuned specifically for the task of emotion recognition, ensuring that the embeddings are optimized for capturing emotional nuances in the text.

The paper introduces a novel explainability technique, a significant contribution to the methodology. This technique aims to enhance the interpretability of the model's predictions. By analyzing the layer-wise convergence of emotion clusters, inter-cluster and intra-cluster distances, and intersection matrices, the explainability module provides insights into the decision-making process of the deep neural network. This layer-wise analysis allows for a nuanced understanding of how the model processes and interprets emotion-related information across different stages of its architecture.

The qualitative and quantitative interpretation of the Text Emotion Recognition

(TER) results is facilitated through the layer-wise analysis of emotion embedding clusters. The paper emphasizes the importance of analyzing inter-cluster and intra-cluster distances, providing valuable insights into the separability and convergence patterns of emotion clusters at various layers of the network.

Acknowledging the limitations of the proposed approach, such as the implicit nature of inferring decision-making and the need for explicit reporting of feature importance, the paper outlines potential avenues for future research. This includes addressing these limitations and extending the system's capabilities to analyze multimodal data (text, speech, visual) while further enhancing its explainability.

In summary, the methodology encompasses the development of an advanced dual-channel TER system, leveraging pre-trained BERT embeddings, and introduces an innovative explainability technique to enhance transparency and interpretability in the context of emotion recognition.

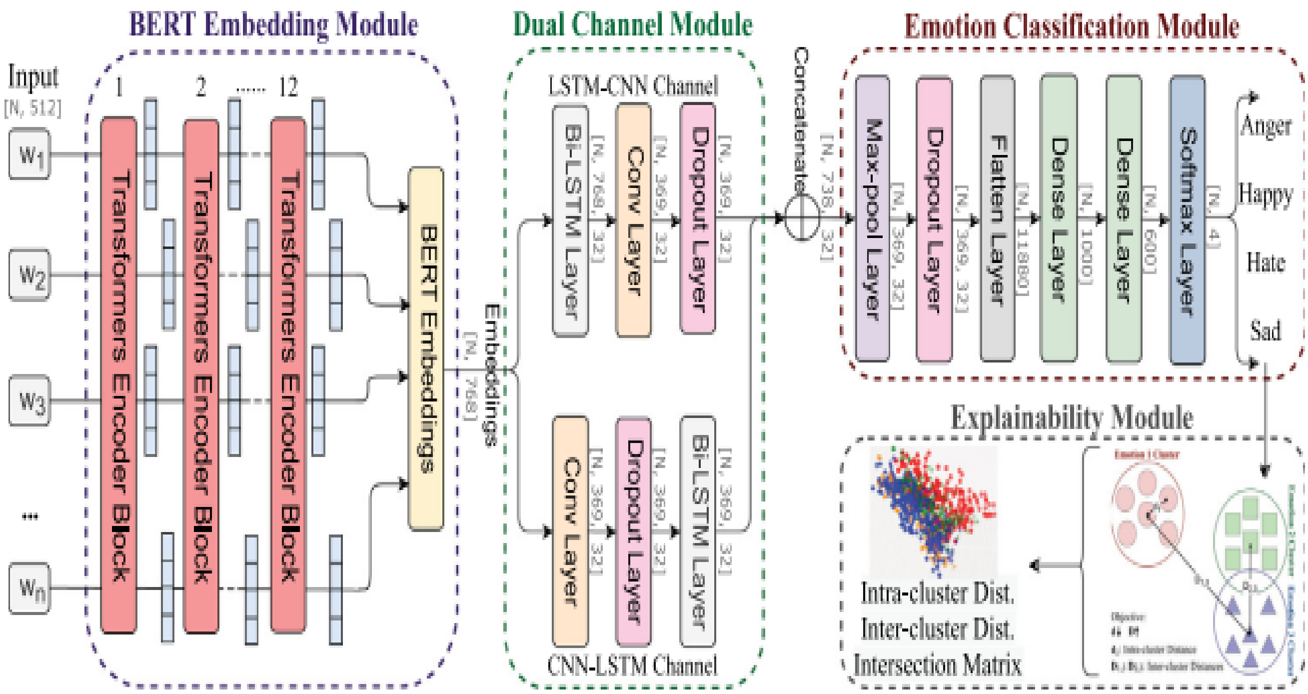


Figure 2.9: Architecture of the dual channel model

2.3 Emotion detection using LSTM and FastText

The paper "Emotion detection from Twitter using LSTM and FastText" dives deep into the methodology of emotion detection using the sequential neural network LSTM and the application of FastText to convert words into embeddings.

The dataset used in the study was sourced from Twitter and consisted of 1304 tweets with six emotional labels: happiness, sadness, anger, fear, disgust, and surprise. The data was collected using web scraping techniques from several influencers as the primary data and trending data for one week as supporting data. The tweets were labeled by students who actively use Twitter to ensure that the data obtained for the training model was not subjective. The preprocessing stage involved case folding, removing punctuations and numbers, tokenizing, stop word removal and stemming.

2.3.1 FastText

Word embedding is a technique for mapping words based on an existing dictionary to produce numeric vectors containing real numbers.They capture semantic and syntactic relationships between words based on their distributional properties in a given text corpus. FastText employs n-gram techinique to capture morphological information and handle out-of-vocabulary words efficiently.

FastText employs the n-gram technique to represent words as bags of character n-grams, where n represents the number of contiguous characters in each subword sequence.[7] In this approach, words are decomposed into overlapping character n-grams, and each n-gram is treated as an individual unit. For instance, with trigrams (n=3), the word "example" would be represented as "ex", "exa", "xam", "amp", "mpl", "ple". The word is then represented as the sum of the vector representations of its constituent character n-grams. FastText utilizes this subword information to capture morphological details, allowing it to effectively handle out-of-vocabulary words, accommodate misspellings, and capture rich morphological features during training. This approach is particularly advantageous in scenarios where subword information is critical, such as languages with complex morphology or situations involving numerous rare words in the training data.

2.3.2 Long Short-Term Memory (LSTM):

Long Short-Term Memory (LSTM) is a specialized type of recurrent neural network (RNN) architecture developed to address the vanishing gradient problem that often hinders the training of traditional RNNs when processing long sequences of data. LSTMs are designed to capture and retain information over extended sequences, making them particularly effective for handling sequential and time-series data. Central to the LSTM architecture are memory cells equipped with three crucial gates: the input gate, the forget gate, and the output gate. These gates regulate the flow of information within the network, allowing it to selectively remember or forget specific pieces of information, thereby facilitating the capture of long-term dependencies.[8] LSTMs have proven to be instrumental in various applications, including natural language processing, speech recognition, and time-series prediction, where understanding context and relationships in sequential data is vital for accurate modeling and prediction [9]

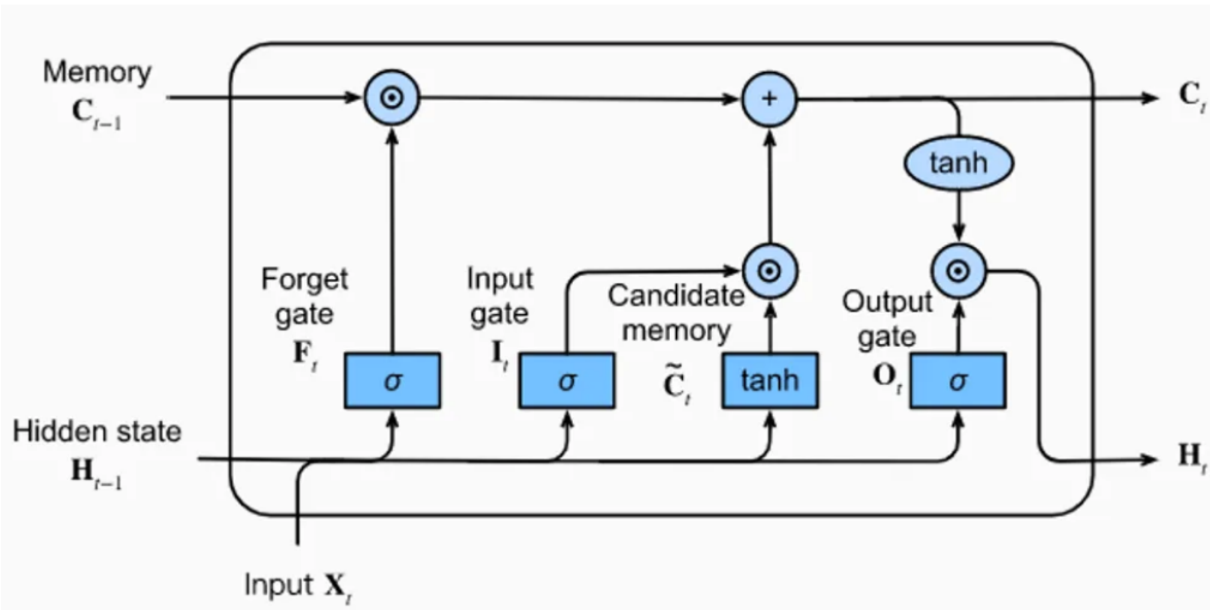


Figure 2.10: Nested LSTM Cell

An LSTM has a hidden state where H(t-1) represents the hidden state of the previous timestamp and Ht is the hidden state of the current timestamp. In addition to that, LSTM also has a cell state represented by C(t-1) and C(t) for the previous and current timestamps, respectively. The hidden state is known as Short term memory, and the cell state is known as Long term memory

The **Forget gate** decides whether the information from the previous time stamp should be kept or forgotten. where:

$$f_t = \sigma (x_t * U_f + H_{t-1} * W_f)$$

Figure 2.11: Forget Gate

X_t is the objective function,

U_f is the weight matrix for input

W_f is the weight matrix for H_{t-1}

This ft is later multiplied with the cell state of the previous timestamp. The contents are forgotten if the product equals zero and retained completely if the product equals one.

$$C_{t-1} * f_t = 0 \quad \text{...if } f_t = 0 \text{ (forget everything)}$$
$$C_{t-1} * f_t = C_{t-1} \quad \text{...if } f_t = 1 \text{ (forget nothing)}$$

Figure 2.12: Determes quantity of content to forget

The **Input gate** is used to quantify the importance of the new information carried by the input.

$$i_t = \sigma (x_t * U_i + H_{t-1} * W_i)$$

Figure 2.13: Input Gate

where:

X_t is the input at the current timestamp,

U_i is the weight matrix of input

H_{t-1} is the hidden state at the previous timestamp

W_i is the weight matrix of input associated with hidden state

The new information that needed to be passed to the cell state is a function of a hidden state at the previous timestamp *t-1* and input x at timestamp t.

$$N_t = \tanh(x_t * U_c + H_{t-1} * W_c)$$

Figure 2.14: New Information

The cell state is updated by

$$C_t = f_t * C_{t-1} + i_t * N_t$$

Figure 2.15: Updating Cell State

The **Output gate** determines the next hidden state of the LSTM and is influenced by the current input as well as the information stored in the memory cell. It controls the flow of information to the output of the LSTM.

$$o_t = \sigma\,(x_t * U_o + H_{t-1} * W_o)$$

Figure 2.16: Output Gate

In this paper six dense layers are implemented to accommodate for the six emotions detected. The dense layer gets input from the final output ht.[10]

$$H_t = o_t * \tanh(C_t)$$

Figure 2.17: Current hidden state

2.4 Accelerating automatic hate speech detection using parallelized ensemble learning models

[11]

With increasing number of social media users and online engagement, it is essential to study hate speech propagation on social media platforms (SMPs). Automatic hate speech detection on social media is of utmost importance as hate speech can create discomfort among users and potentially generate a strong reaction in society. Ensemble learning algorithms are helpful in addressing sentiment-based classification due to their fault tolerance and efficiency. However, a simple, scalable, and robust framework is required to deal with largescale data efficiently and accurately. Therefore, we propose parallelization to the standard ensemble learning algorithms to speed up the automatic hate speech detection on SMPs. In this study, we parallelize bagging, A-stacking, and random sub-space algorithms and test their serial and parallel versions on the standard highdimensional datasets for hate speech detection. The experiments are performed using six datasets that address hate speech propagation during events like the COVID-19 pandemic, the US presidential election (2020), and the farmers' protest in India (2021). Our parallel models observe a significant speedup with high efficiency, claiming that the proposed models are suitable for the considered application. Also, one of the main motivations of this study is to highlight the importance of generalization by testing the models under the cross-dataset environment. We observed that the accuracy is not affected while parallelizing the algorithms compared with serial algorithms executing on a single machine.

2.4.1 Parallel Ensemble Learning Models

The methodology initiates by introducing a parallel architecture designed to optimize the training process of base classifiers. This parallelization is explicitly introduced to accelerate the training phase and improve overall efficiency. The base classifiers, trained individually in parallel, are later integrated to make the final decision. The ensemble algorithms considered include Parallelized Bagging, Parallelized A-Stacking, and Parallelized Random Subspace.

Parallelized Bagging (P-Bagging)

P-Bagging is presented as an algorithm requiring training data (D_{Train}), test data (D_{Test}), and a classification algorithm (K). The algorithm involves the sampling of sets of instances, and training base classifiers in parallel. Each process operates independently, utilizing the classification algorithm K to generate base classifiers. The final output of the algorithm is an ensemble Z of these base classifiers, which is then used to classify query instances.

Parallelized A-Stacking (P-AStacking)

P-AStacking combines a clustering algorithm (C) with a meta-classifier (M). Instances are divided into clusters; for each cluster parallel processes generate base classifiers. Each of these base classifiers makes its own individual decision, and then the meta-classifier (M) integrates them to determine whether or not a test instance is positive. Careful choice of the best classifier for each cluster, according to its performance on validation data, is an important feature.

Parallelized Random Subspace

Algorithm 3 describes the method for parallelizing Random Subspace. The algorithm lets you sample features from the original training set to create sets of instances. These sets are assigned a number of processes, each tasked with developing the base classifier by means of a learning algorithm (L). These base classifiers pass on the decisions they make, and then the ensemble comes to a final decision based upon these individual outputs.

2.4.2 Experimental Setup

The study uses an extremely strict experimental environment to test the effectiveness of proposed models on different datasets. Selected datasets include Waseem and Hovy, SemEval2019, Covid-HL, Covid-ML; US Election a dataset just created for the Farmers

Protest. All of these datasets are preprocessed to exclude information that is irrelevant. The study describes the detailed steps taken in compilation for each dataset.

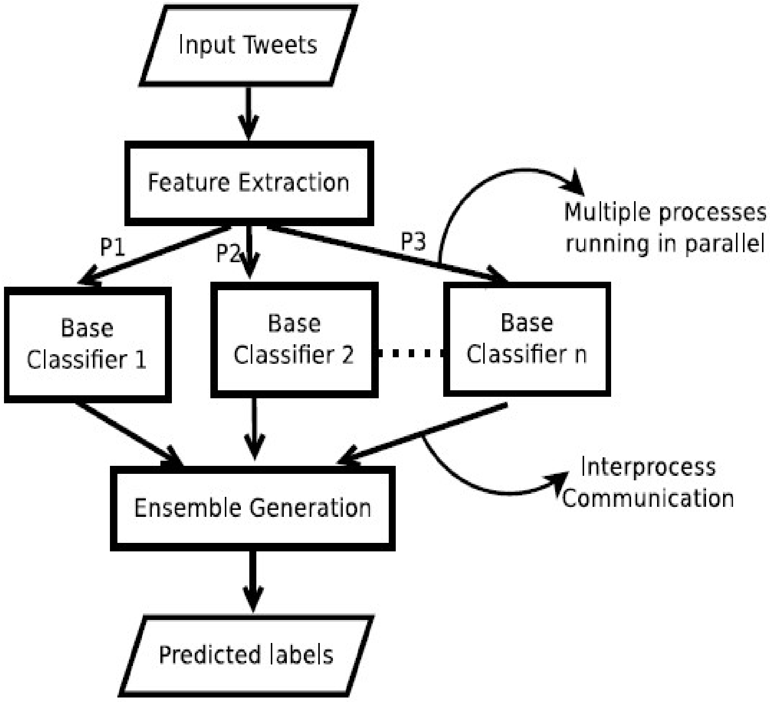


Figure 2.18: Schematic Representation Of Parallelized Ensmeble Model

Dataset Creation

The generation of datasets is a difficult process, and one recently introduced dataset Farmer Protest (2021) aims to capture hate speech in India's farmers 'protest. It is manually tagged with a definition of hate against delete, and annotators–research scholars from the Department of CSE at IIT (BHU), India provides labels based on majority voting. This process of annotation guarantees a balanced dataset, and some examples of hateful tweets are given. Preprocessing eliminates URLs and converts emoticons as well, but retains user mentions for useful information.

Experimental Setup and Evaluation Metrics

The experiments are conducted on a supercomputer, providing details of the hardware specifications. Python3, along with the joblib library, is used to implement the easy multiprocessing functionality. Evaluation metrics include precision, recall, F1-score, serial time (T_s), parallel time (T_p), speedup (S), and efficiency (E). The study adopts two experimental designs: within-dataset and cross-dataset, to assess model performance and generalization abilities.

2.4.3 Results

Within-Dataset Performance

The study evaluates the performance of existing state-of-the-art hate speech detection models on six datasets: Waseem and Hovy, SemEval2019, Covid-HL, Covid-ML, US Election, and Farmers Protest. The models exhibit varying performance, achieving F1-scores ranging from 72.32 percent to 90.12 percent. The parallel versions of ensemble algorithms, including A-Stacking, Bagging, and Random Subspace, showcase improved efficiency and speedup, particularly notable in larger datasets.

Cross-Dataset Performance

This shows a big drop in the performance of models compared to within-dataset situations, stressing the importance of cross-dataset evaluation. Models trained on one dataset and tested on another have lower F1-scores, showing the difficulty of generalization. The study offers a detailed analysis of the effects of cross-dataset evaluation on hate speech detection models.

Analysis

The analysis section examines the details of time usage, looking at serial and parallel execution times for ensemble classifiers. The results demonstrate a considerable decrease in the time needed for parallel versions on different datasets. It proves that using parallelization is effective at improving the training process of hate speech detection models.

2.5 Hate speech detection using static BERT embeddings [3]

With increasing popularity of social media platforms hate speech is emerging as a major concern, where it expresses abusive speech that targets specific group characteristics, such as gender, religion or ethnicity to spread violence. Earlier people use to verbally deliver hate speeches but now with the expansion of technology, some people are deliberately using social media platforms to spread hate by posting, shar- ing, commenting, etc. Whether it is Christchurch mosque shootings or hate crimes against Asians in west, it has been observed that the convicts are very much influenced from hate text present online. Even though AI systems are in place to flag such text but one of the key challenges is to reduce the false positive rate (marking non hate as hate), so that these systems can detect hate speech without undermining the freedom of ex- pression. In this paper, we use ETHOS hate speech detection dataset and analyze the performance of hate speech detection classifier by replacing or integrating the word embeddings (fastText (FT), GloVe (GV) or FT + GV) with static BERT embeddings (BE). With the extensive exper- imental trails it is observed that the neural network performed better with static BE compared to using FT, GV or FT + GV as word embed- dings. In comparison to fine-tuned BERT, one metric that significantly improved is specificity.

2.5.1 BERT Architecture

Its architecture is a radical departure from traditional transformer models. BERT does without decoders, and only consists of encoder structures. During training, BERT achieves bidirectionality by using encoders that process input data in both left-to-right and right-to-left directions. This allows it to capture the sometimes subtleties of contextual relationships between words. With layers of transformers using self-attention mechanisms, BERT assigns different degrees of weight to each word in a sentence based on its variation. Thus the goal is that language meaning can be understood more completely and widely across linguistic nuances. BERT's flexibility is further enhanced by the addition of WordPiece Tokenization, which breaks words down into subword units to help resolve out-of-vocabulary (OOV) problems. This flat, bidirectional context-learning architecture with profoundly inductive understanding and transfer learning capabilities has made BERT one of the most impactful tools around today for natural language processing; its impressive performance consistently demonstrates that it can handle a wide range of NLP tasks. BERT's revolutionary architecture, featuring bidirectional context learning, self-attention mechanisms, and transfer learning capabilities, has not only redefined language understanding models but also set a new standard for contextualized word representations in the field of natural language processing. The pre-training phase of BERT involves exposure to vast corpora of text, imbuing the model with a rich understanding of language semantics. This pre-learned contextual knowledge, stored in the form of embeddings, forms the basis for BERT's remarkable adaptability in detecting hate speech. During the fine-tuning phase, the model is tailored to task-specific objectives, such as hate speech detection, allowing it to generalize its pre- learned knowledge to the intricacies of identifying harmful language. BERT's attention to context, bidirectional processing, and transfer learning capabilities collectively simplify the complex task of hate speech detection, enabling the model to navigate the diverse and evolving landscape of online communication with unparalleled accuracy and efficiency.

2.5.2 Evaluation of BERT Model

Assessing the effectiveness of BERT (Bidirectional Encoder Representations from Transformers) [2] entails an overview on several natural language processing tasks. A popular evaluation benchmark is the General Language Understanding Evaluation (GLUE) benchmark, which covers a variety of NLP tasks including sentiment analysis and text similarity. All previous approaches thus far have been bested by BERT on the GLUE tasks. This indicates just how outstanding BERT is at representing and contextualizing relations within a sentence, which relies upon its bidirectional cross-attention mechanism to comprehensively analyze this information from different directions all in one go. It's also worth mentioning that often the Stanford Question Answering Dataset (SQuAD) is used to test BERT on question-answering tasks. BERT's performance on SQuAD, where it achieves human-level accuracy at extracting answers from context, clearly demonstrates that the model does a good job of understanding complex linguistic context. Further, BERT's assessment outstrips task-specific benchmarks to reach real appli- cations in information retrieval, document classification and sentiment analysis on social media. And thanks to its bidirectional context learning, BERT shines when it needs to understand the relationships between words. Application to industrial tasks The wider ap- plicability of BERT can be further enhanced by fine-tuning on domain-specific datasets. But the assessment of BERT is not an easy task. In some deployment scenarios, the model's large size and resource-hungry nature may be a constraint. There will have to be optimization for efficiency of performance. In addition, dealing with possible biases in the training data and maintaining fairness of predictions are important parts of a comprehensive assessment plan for using BERT responsibly across different scenarios.

2.5.3 Static BERT Embedding Matrix

The embedding matrix [4] in the proposed BERT model encompasses word embeddings for every word within the dataset. Unlike traditional embeddings, BERT provides contextualized representations, meaning the same word can have distinct embeddings contingent on its usage in a sentence. These context-dependent embeddings are organized in a dictionary, where each unique word corresponds to its respective vector. During processing, each contextualized embedding is added to the vector associated with the unique word in the dictionary. This approach enables the model to efficiently store and retrieve diverse word embeddings, accommodating variations in meaning based on the word's context in different sentences. This dynamic utilization of contextual embeddings enhances the model's capacity for nuanced understanding and contributes to its efficacy in discerning hate speech across varied linguistic contexts. The static BERT embedding matrix serves as a foundational element in natural lan- guage processing obligations, presenting a fixed illustration of phrases in a manner that captures the contextual intricacies of language. At its middle, the embedding matrix is a research table that maps every phrase within the vocabulary to a high-dimensional vector illustration. This static nature distinguishes it from dynamic embeddings, in which the phrase representations are updated at some point of the training method. In the context of BERT, the embedding matrix is pre-trained on large corpora, gaining knowledge of rich semantic relationships and contextual nuances. The ensuing embeddings encapsulate the meaning of words not as remoted entities but as additives intricately tied to their surrounding linguistic context. This property is specially vital for tasks like hate speech detection, wherein expertise the broader context is essential in discerning the nuanced that means of phrases and phrases inside potentially harmful language.

2.6 Summary and Gaps Identified

Method	Advantages	Disadvantages
BERT-CNN: A Deep Learning Model for Detecting Emotions from Text	• Harnesses the power of BERT pre-trained models for contextual understanding. • Integrates CNN for local feature extraction. • Achieves state-of-the-art results on emotion detection tasks.	• May require substantial computational resources for training BERT-CNN. • Fine-tuning BERT models can be time-consuming.
A BERT based dual-channel explainable text emotion recognition system	• Utilizes a dual-channel architecture for a nuanced approach to emotion recognition. • Incorporates explainability techniques for model interpretation.	• May introduce complexity in the model architecture. • Explainability methods might add computational overhead.

Emotion Detection from Text using LSTM and FastText	• Rich Word Embeddings: FastText provides subword embeddings, capturing morphological information. • Sequential Context: LSTM captures sequential dependencies in the text, preserving temporal information. • Transfer Learning: Pre-trained FastText embeddings can enhance model performance.	• Computational Intensity: Training LSTM models can be computationally expensive. • Data Dependency: Requires a sufficiently large and diverse dataset for effective training.

Hate Speech Detection Using Ensemble Learning	• Focuses on the acceleration of hate speech detection through parallelized ensemble learning models. • Demonstrates improved efficiency and reduced computation time.	• Specific to acceleration aspects, might lack in-depth analysis of model interpretability. • The applicability of parallelized models to various hate speech contexts may require further investigation.

Hate speech detection using static BERT embeddings	• BERT (Bidirectional Encoder Representations from Transformers) embeddings capture contextual information, allowing for a more nuanced understanding of language. • BERT's pre-trained embeddings leverage transfer learning, benefiting from knowledge gained across a wide range of tasks and domains.	• BERT models, including their static embeddings, have relatively large memory requirements. This can be a limitation in resource-constrained environments, especially on devices with limited RAM or in scenarios where memory optimization is critical. • Hate speech often involves the use of slang, memes, and evolving internet language.

Table 2.1: Comparison of existing models

Chapter 3

Requirements

3.1 Hardware and Software Requirements

3.1.1 Hardware Requirements

- **Processor:**

  A modern multi-core processor, such as the Intel Core i5 or AMD Ryzen 5 series, is recommended to efficiently handle concurrent tasks associated with natural language processing and machine learning.

- **Memory (RAM):**

  A minimum of 8 GB RAM is essential for smooth processing and to accommodate the memory-intensive requirements of machine learning tasks.

- **Storage:**

  Adequate storage space is required for storing collected data, pre-trained models, and application files. The use of Solid State Drives (SSDs) is recommended to ensure fast data access and retrieval.

- **Graphics Processing Unit (GPU):**

  While not mandatory, the inclusion of a Graphics Processing Unit (GPU) can significantly expedite machine learning tasks. A CUDA-enabled GPU, particularly from NVIDIA, is beneficial when working with frameworks like TensorFlow.

3.1.2 Software Requirements

- **Python 3.x:**

  Python serves as the primary programming language, providing the necessary flexibility for working with BERT and associated libraries.

- **Keras/TensorFlow Library:**

  The project utilizes the Keras library with TensorFlow backend for building and training the neural network models. This library provides a high-level interface for defining and training deep learning models, including text classification models.

- **Text Preprocessing:**

  The ktrain library handles text preprocessing steps, such as tokenization and encoding, to prepare the text data for input into the pre-trained models.

- **Pre-trained Model (ktrain):**

  The project leverages pre-trained models available through ktrain, such as BERT or other transformer-based models, which have been pre-trained on large text corpora and can be fine-tuned for the specific task of emotion detection or text classification.

Chapter 4

System Architecture

The Social Media Emotional Analysis and Task Recommendation System is a comprehensive platform designed to enhance users' well-being by providing deep insights into their emotional states during social media interactions. This system leverages advanced Natural Language Processing (NLP) techniques for emotion and hate speech detection, along with a text summarization feature, to generate detailed monthly mental state reports for users based on their social media activities.

4.1 System Overview

This is an all-inclusive social media analysis and mental health tracking system that aims to provide users with useful information on their emotional state while they interact through different online platforms. The system integrates emotion recognition, hate speech detection and text summarization to create a comprehensive perspective of the user's online activities. The process starts with a tailor-made browser extension where users can effortlessly capture and paste their social media text data into the system. This extension acts as a conduit between what the user does on their social media and how the system subsequently analyzes those activities.

The emotion detection module is the first key component of the system. This module uses advanced NLP techniques, possibly BERT-based models, to parse the user's textual content and determine their underlying emotions. Social media posts and comments are often filled with subtle emotional tones, and the system tries to detect them correctly by identifying their categories accurately. Emotion detection system gives a detailed analysis of how the user is feeling whether it's joy, sadness, anger or any other sentiment during their interactions on social media.

At the same time, there is a hate speech detection module integrated within the system to promote a safer and more inviting online space for users. This module uses machine learning models trained on hate speech databases to detect abusive language or sentiments in the user's social media content.

Data gathered from the emotion detection and hate speech detecting modules is what forms part of a complete monthly mental state report. This report gives users a comprehensive view of their emotional state over the month, highlighting emotional patterns and potential problem areas. By providing this data in an organized and easily understood way, users can educate themselves and make educated choices about how to use computer technology.

The system doesn't not simply analyze, but actively contributes to making the user better. The monthly mental state report becomes a useful instrument to recommend personalized tasks aimed at the user's emotional condition. These recommendations have the potential to positively impact a user's emotional health by proposing activities that are in line with their current mental state. For instance, if the user has had higher levels of stress then the system may propose calming actions or mindfulness activities.

Besides, this project also involves a text summarization algorithm to help deal with the information overload in social networking. This feature allows users to quickly and easily digest large portions of text content as it takes long pieces of information and shortens them into brief summaries. By employing text summarization algorithms, the system enables users to keep updated without being overwhelmed by all information present on social media platforms.

In conclusion, this project is a well-made and user oriented system that does not only analyze break down the emotional components of social media communication but take active part in influencing users' welfare by offering customized pieces of advice on how to avoid hate speech or unpleasant situations. Combining emotion detection, hate speech detection and text summarization makes it a powerful tool for the users who want their online experience to be more positive, better informed and conscious.

4.2 Architecture diagram

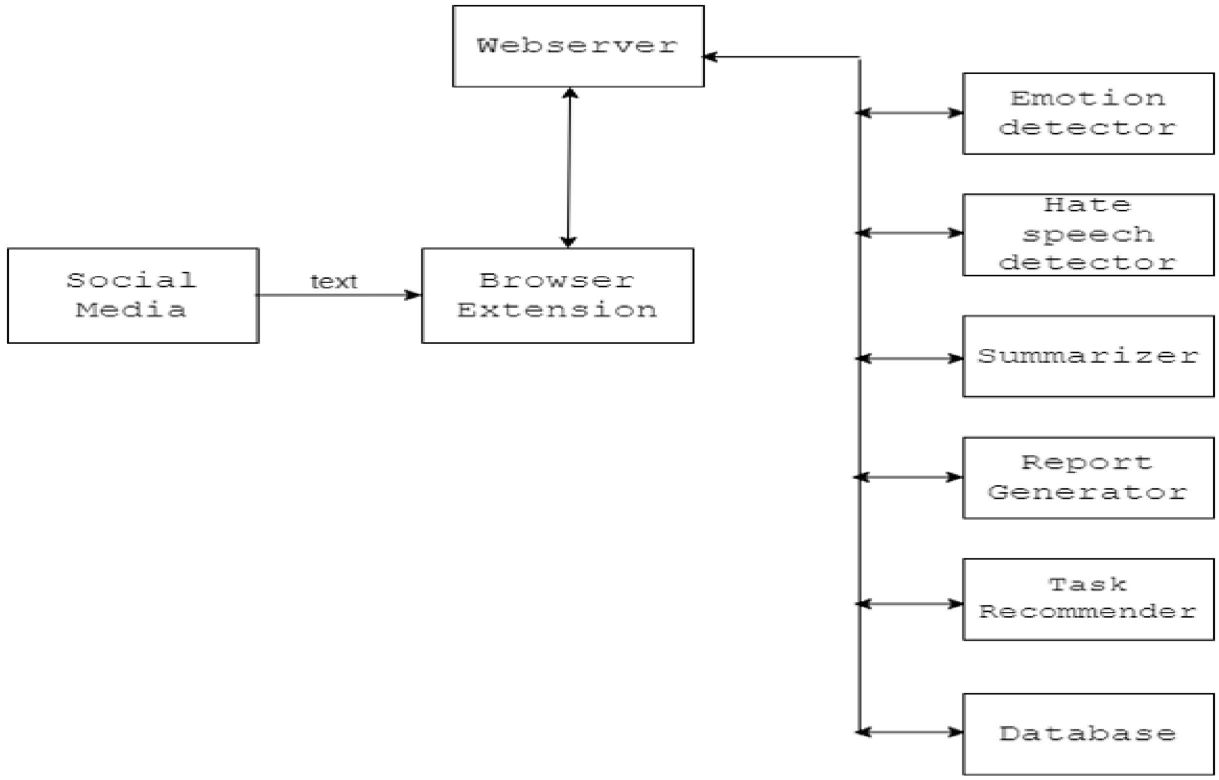


Figure 4.1: Architecture diagram

The browser extension captures text typed on various social media websites as intended. This text is then sent to the web server where it's emotion contents are analysed and quantified by the fine-tuned emotion detector and also a hate speech screening is done. The results are to be then stored in a database. The report generator accesses the results from the database on a weekly basis and generates a report showing the statistics of the week. The report is fed to the task reccommender which suggests a variety of tasks to elevate the user's mental state.

The extension also provides an additional feature of summarization. Whenever the user drags a piece of text on any webpage, the extension captures it, sends it to the back-end to be summarized the by a summarizer api, which is then returned back to the extension to be displayed in the form of a pop-up.

4.3 Modules

The project has the following modules:

1. Browser extension - To read data typed in a website and send it to server.
2. Server & Database - To process the text and store the resulting emotion.
3. Emotion detection - Uses NLP libraries to capture emotion specified in a sentence.
4. Hate speech detection - To detect if the text received through the extension constitutes hate speech. The results are sent to the database for report generation
5. Report Generation and task recommendation - To generate a mental state report from data in the database(emotions and hate speech) and suggest tasks to improve mental state.
6. Text summarization - To summarize text selected by the user and display the result in a pop up.

4.4 Modules in detail

4.4.1 Browser Extension

The extension is capable of extracting texts typed in various social media platforms such as Instagram, Facebook, Reddit etc. Extraction of text is facilitated via the application of event listeners provided in JavaScript. The event listener attentively caters to any keystrokes which are stored in a variable. The variable content is then periodically sent over to the web-server utilising the WebSocket api. Unlike traditional HTTP communication, WebSocket establishes a persistent, bidirectional connection, facilitating real-time, low-latency communication. This enables instant and efficient data transmission, making it ideal for applications that demand quick updates, such as online gaming, financial platforms, and live chat applications. Its stateful nature allows for continuous data flow, eliminating the need for frequent reconnection, and significantly reducing overhead. Additionally, the user will be required to enter his/her login credentials into the browser extension for the purpose of identification and ownership of texts in the back end.

4.4.2 Webserver

The webserver is responsible for establishing user credentials and the subsequent authentication requests. It is capable of handling bidirectional communication with the individual browser extension by means of ws library. The web server is responsible for triggering the action of emotion detector, hate speech detector and summarizer. It also stores into the database the intermediate results produced by the emotion detector with the corresponding User ID.

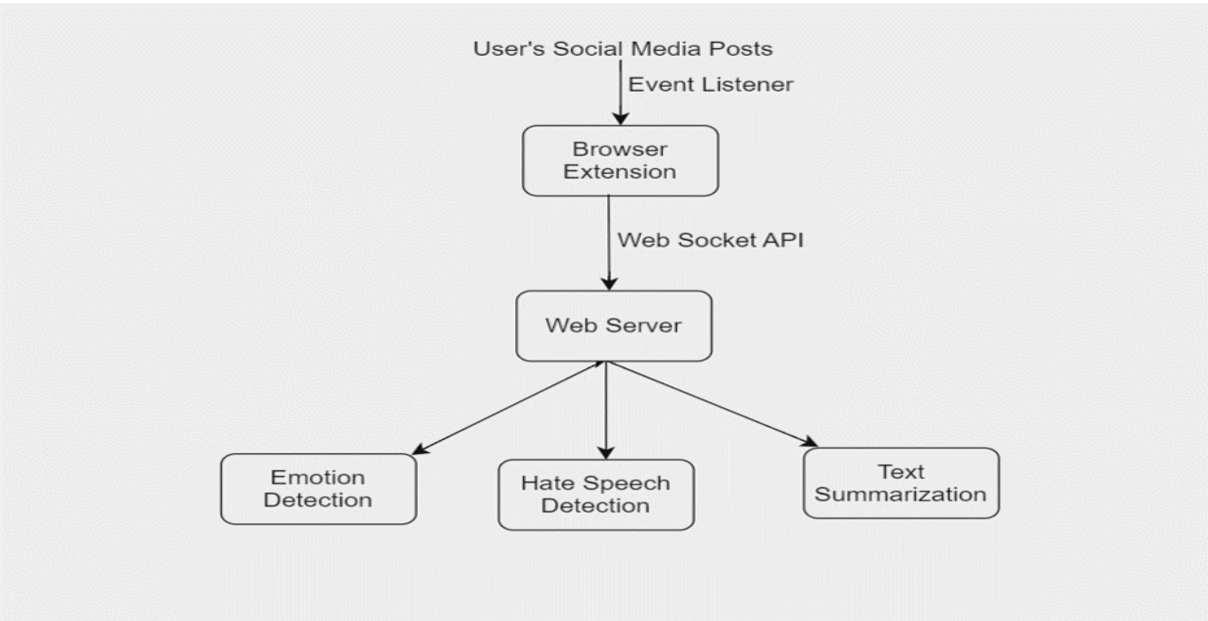


Figure 4.2: Browser extension and webserver

4.4.3 Text Summarizer

The browser extension provides the additional feature of summarizing for text larger than 200 characters. Whenever the user drags any text satisfying the size constraint mentioned before, it sent to the web server, summarized and returned back to the extension in the form of a popup.
A summarizer typically progresses through the following stages:

1. Text preprocessing: involves text cleaning, tokenization, stopword removal and lemmatization.

2. Sentence scoring: Sentence scoring in natural language processing (NLP) involves assigning a numerical score or weight to individual sentences within a document based on certain criteria. The goal is often to identify the most important or relevant sentences in a given context.

   TF-IDF is a numerical statistic that reflects the importance of a word in a document relative to a collection of documents. Sentence scores can be calculated by considering the TF-IDF values of the words present in each sentence.

$$\mathrm{TF}(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

Figure 4.3: TF scoring

$$\mathrm{IDF}(t,D) = \log\left(\frac{\text{Total number of documents in the corpus } N}{\text{Number of documents containing term } t \text{ in the corpus } D{+}1}\right)$$

Figure 4.4: IDF scoring

   Putting it all together, the TF-IDF score for a term t in a document d with respect to a document collection D is the product of the term frequency (how often the term appears in the document) and the inverse document frequency (how unique or rare the term is across the collection).

$$\textit{TF-IDF}(t,d,D)=\textit{TF}(t,d)\times \textit{IDF}(t,D)$$

TF scoring also provides the bonus of vectorizing each term in the document. Therefore, a sentence in the document in its vector form will be a collection of such vectors.

3. Cosine Similarity:

   Cosine similarity measures the angle between two vectors. In the context of summarization, it measures how similar each sentence vector is with all other sentence vectors. It ranges from -1 (completely opposite directions) to 1 (identical vectors). It's calculated using the following formula:

   $$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

   Figure 4.5: Cosine similarity

   where:
   A and B are the vectors representing two sentences
   A . B is the dot product of vectors A and B.
   $||A\ ||$ and $||B||$ are the magnitudes (lengths) of A and B

   Sentences with higher average similarity scores are considered more representative of the main content and are thus selected for the summary.

4. Page Ranking Algorithm:

   The PageRank algorithm[12] is an algorithm used by the Google search engine to rank web pages in its search results.The main idea behind PageRank is to assign a numerical weight, or "PageRank score," to each web page based on the link structure of the web. The underlying assumption is that a page is more important if it is linked to by other important pages

In the context of text summarization, the pagerank algorithm treats the sentences as nodes in a graph, where edges between nodes represent the cosine similarity between the corresponding sentences. Higher cosine similarity results in stronger connections (edges) between nodes.

A similarity matrix S[i][j] represents the similarity scores between sentence i and j. A transition matrix T[i][j] is obtained by

$$T[i][j] = \frac{S[i][j]}{\sum_k S[i][k]}$$

Figure 4.6: Transition matrix

The transition matrix T[i][j] represents the probability of jumping from sentence i to sentence j based on their similarity.

A uniform page rank score is assigned initially to each sentence. To calculate page rank score of sentence i at iteration t+1:

$$PR_i^{(t+1)} = \sum_j \left(\frac{PR_j^{(t)}}{L_j} \cdot T[j][i] \right)$$

Figure 4.7: Page rank equation

where:

$PR_i^{(t+1)}$ is the updated PageRank score for sentence i at iteration $t+1$.

$PR_j^{(t)}$ is the PageRank score for sentence j at iteration t.

T_{ji} is the transition probability from sentence j to sentence i.

L_j is the outdegree of sentence j.

When the page rank scores appear to converge after a few iterations, those sentences which cross a custom threshold are selected to be included in the summary.

4.4.4 Emotion detection

To perform emotion detection, we leverage a pre-trained model from the transformers library, with a preference for BERT. Using the 'pipeline' module, we seamlessly load the pre-trained model into our Python environment. Subsequently, we embark on fine-tuning the model specifically for the task of emotion detection. Textual content harvested from websites is collected through a dedicated plugin, and this data is then sent for processing to the model, which operates on a server. The outcomes of the emotion detection task are meticulously stored in a database for further analysis and reference. It typically progresses through the following stages:

(a) Attention mechanism: Self-attention[13] is used to understand the relationships between tokens in a sentence and capture the inner structural details.

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$

Figure 4.8: Scaled dot product attention equation

Q-Query matrix of each word

K-Key matrix of each word

V- Value matrix of each word

dk- Dimension of K matrix

Mean pooling is used to convert the individual words into sentence representations.

(b) Classification layer:

Weight Matrix:

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1H} \\ w_{21} & w_{22} & \dots & w_{2H} \\ w_{31} & w_{32} & \dots & w_{3H} \end{bmatrix}$$

Bias Vector:

$$b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Linear Transformation:

$$z = W \cdot h + b$$

Where:

- W represents the weight matrix of the classification layer.
- b denotes the bias vector of the classification layer.
- h is the input from the attention layer.
- z signifies the output of the classification layer for each sentence.

The softmax function converts the linear transformation result z into probabilities for each emotion class.

(c) Loss function: The model is trained using a loss function, such as cross-entropy loss, which measures the difference between the predicted probabilities and the true labels. Categorical Cross-Entropy Loss:

$$\text{Categorical Cross-Entropy Loss} = -\sum_{i=1}^{C} y_i \cdot \log(p_i)$$

Where:

- C is the number of classes.
- y_i is the true probability distribution (one-hot encoded vector) for class i.
- p_i is the predicted probability distribution for class i.

Let's assume the True Label for sad sentence=[0,1,0]

Predicted Probabilities(Z)=[0.2,0.7,0.1]

Cross-Entropy Loss=[0log(0.2)+1log(0.7)+0log(0.1)]

Cross-Entropy Loss=0.3567

After back propagation, the weight and bias matrix are updated with the correct value.

4.4.5 Report generation

The output obtained from the emotion detection module is utilized to create a comprehensive social media mental health report for the user. At the close of each day, the system compiles the results from the database to generate a daily report, which is subsequently stored in the database for future reference. Upon a user's request, their individual report can be retrieved from the database after logging in. Additionally, weekly reports can be generated upon specific user requests, providing a more extended overview of their mental health trends over time.

4.4.6 Task recommendation

Upon the user's request, the system suggests tasks tailored to their mental health report. This functionality is integrated into the website where users access their generated reports. Utilizing various APIs, the system recommends tasks and even suggests songs based on the individual's report, enhancing the user experience with personalized recommendations aligned with their emotional well-being.

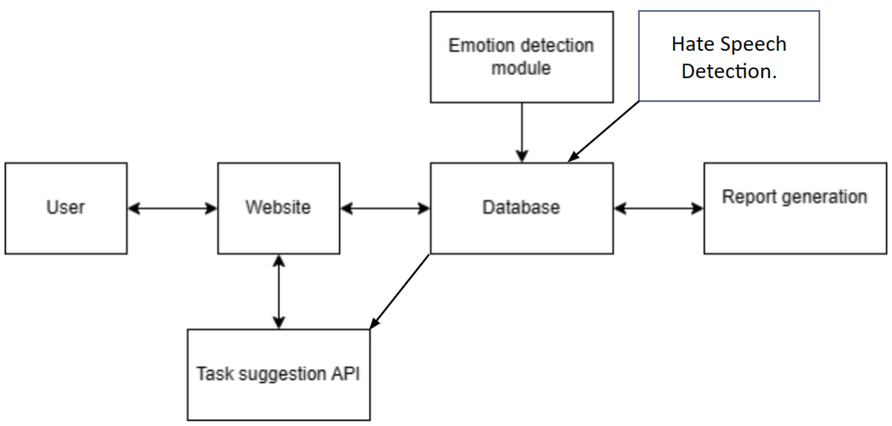


Figure 4.9: Report Generation and Task Recommendation

4.4.7 Hate Speech Detection

An essential part of our technology is hate speech identification, which finds and marks discriminating and harmful material in user-generated text. For the purpose of detecting hate speech, we have optimised a BERT (Bidirectional Encoder Representations from Transformers) model.

Training the BERT model on a labelled dataset that included instances of abusive language, hate speech, and regular language allowed researchers to fine-tune the machine's ability to discern between various content kinds. The weights and parameters of the model were changed during this process to increase its sensitivity to the linguistic subtleties connected to hate speech.

Our server-side processing pipeline incorporates the improved BERT model. The browser extension transmits text to the server for processing when it receives user-entered content from social media platforms. The BERT model reads the text and determines whether it contains hate speech or objectionable language.

A probability score indicating the text's propensity to contain hate speech is produced by the model. The text is marked as potentially having hate speech if the score is higher than a predetermined threshold. After being reported, the text is analysed further; depending on the context and overall analysis, it may go through additional checks or be added to the user's mental status report.

We hope to give consumers a safer online experience by utilising BERT's capability and optimising it for hate speech detection. This will help to warn users about potentially dangerous content and foster an online community that is more inclusive and courteous. This system element helps us achieve our overarching objectives of enhancing user welfare and encouraging constructive interactions on social media platforms.

4.5 Work Breakdown

- **Crispin Mathew**- Browser Extension, Emotion Detection, Text Summarization, Backend

- **David Johns Denny** – Website, Emotion Detection, Text Summarization, Report Generation
- **Akshay Ajit** - Website, Hate Speech Detection, Task Recommendation, Back-end
- **Devananth H** – Browser Extension, Hate Speech Detection, Task Recommendation, Report Generation

4.6 Gantt Chart

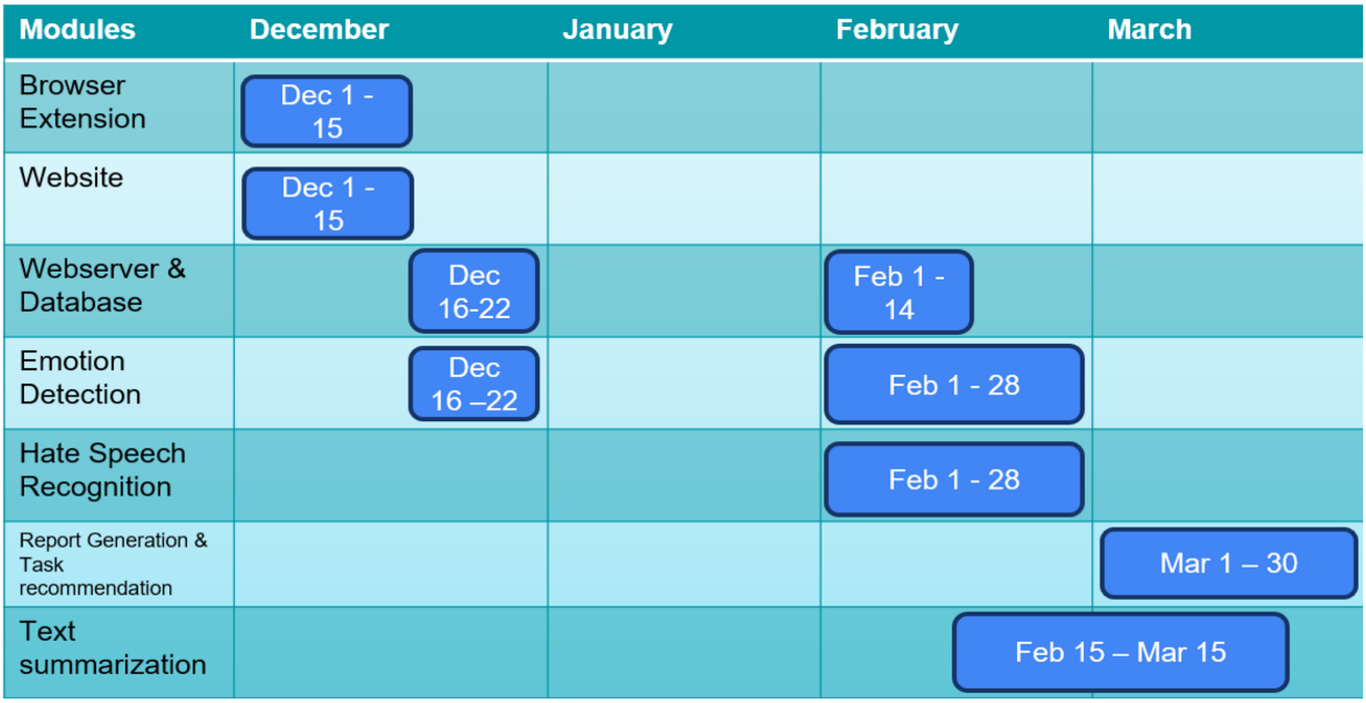


Figure 4.10: Proposed Gantt Chart

4.7 Conclusion

Finally, our system offers a novel method of improving users' social media engagements with the help of sophisticated NLP and machine learning technologies. As we use Emotion Detection and Hate Speech Detection modules within a custom browser extension, this allows for real-time analysis and creates periodical Mental State Reports on monthly basis. These reports provide users with a retrospective perspective at their emotional states, allowing for increased self-knowledge and promoting positive online spaces.

The interactivefeatures of thesystem, such as task recommendations tailored to emotional states that encourage positive activity and mental health go beyondcommonly used modelling. A text summarization tool adds value to users allowing them quickly know the main points within long posts helping with content digestion.

This all-encompassing mechanism not only scrutinizes and evaluates but also shapes users' digital wellness, transforming the social media sensation. As we progress, regular variations and user comments will determine subsequent enhancements so that the system remains relevant as well as useful in solving the problem of varied nature(dynamics) found with online interactions.

Chapter 5

Results And Discussions

5.1 Overview

The functionality described above was developed in it's entirety in the form of a browser extension. The extension is capable of reading text the user has entered into social media platforms and sending it to a remote server hosted on another computer for further processing.

Further processing involves the use of an emotion detection model (BERT) to find out the values of the following emotions in the text - joy, sadness, anger, fear, neutral. Also, hate speech detection is employed on the text to find out if there has been any instance of hate speech from the user's side in his/her social media activity. Hate speech detection is also done using a fine-tuned BERT model.

5.2 Backend

Each sentence that the user enters is processed individually first and stored into the database i.e, for each sentence, there is a separate value for the emotions joy, sadness, anger fear and neutral. So, if the user has entered 1000 sentences in social media on that day, there are 1000 records in the database. The browser extension is currently capable of capturing text from the following three platforms - facebook, reddit and instagram.

5.3 Combining Scores To Get Final Results

The process of combining the results from all platforms happens at the end of day. Text from different social media platforms go to different tables in the database. Firstly, we calculate emotion scores for sentences from reach social media platform individually and combine their scores using a normalization process to achieve the final score from that

Figure 5.1: Screenshot of website developed for the project

platform. Therefore, we now know how much impact a particular social media platform, for example instagram has had on the user's mental state for that day.

Now comes the process of combining scores from all three social media platforms. Here, we encounter a problem. The scores from the three different platforms may not deserve the same weightage in the final score. For example, text the user has encountered on a public forum like reddit may not be as indicative of his/her mental state like the text he/she entered on a much more private platform like instagram. From our research, we found that people are much more likely to share positive news on instagram compared to facebook and reddit. Also, negetive emotions are far more easily shared on facebook compared to instagram.

Therefore, while calculating the final values (how all three websites have contributed to the user's mental state for the day), for arriving at the final value for the emotions joy and neutral, values from instagram where given more importance. Likewise, for the emotions anger, fear and sadness, values from facebook and reddit were given more importance.

```
mysql> select * from insta_data;
+---------------------------------+-------------------------+
| text                            | timestamp               |
+---------------------------------+-------------------------+
| hello cristiano, i am a huge fan | 2024-02-21T09:10:37.089Z |
+---------------------------------+-------------------------+
1 row in set (0.00 sec)
```

Figure 5.2: Text from instagram captured by extension

```
mysql> select * from insta_results;
+------------+-----------+----------------+-------------+-------------+-------------+
| date       | joy       | sadness        | fear        | anger       | neutral     |
+------------+-----------+----------------+-------------+-------------+-------------+
| 2024-02-21 | 0.9586112 | 0.00096629077  | 0.019724492 | 0.002208148 | 0.018489823 |
+------------+-----------+----------------+-------------+-------------+-------------+
1 row in set (0.00 sec)
```

Figure 5.3: Emotion score for sentences

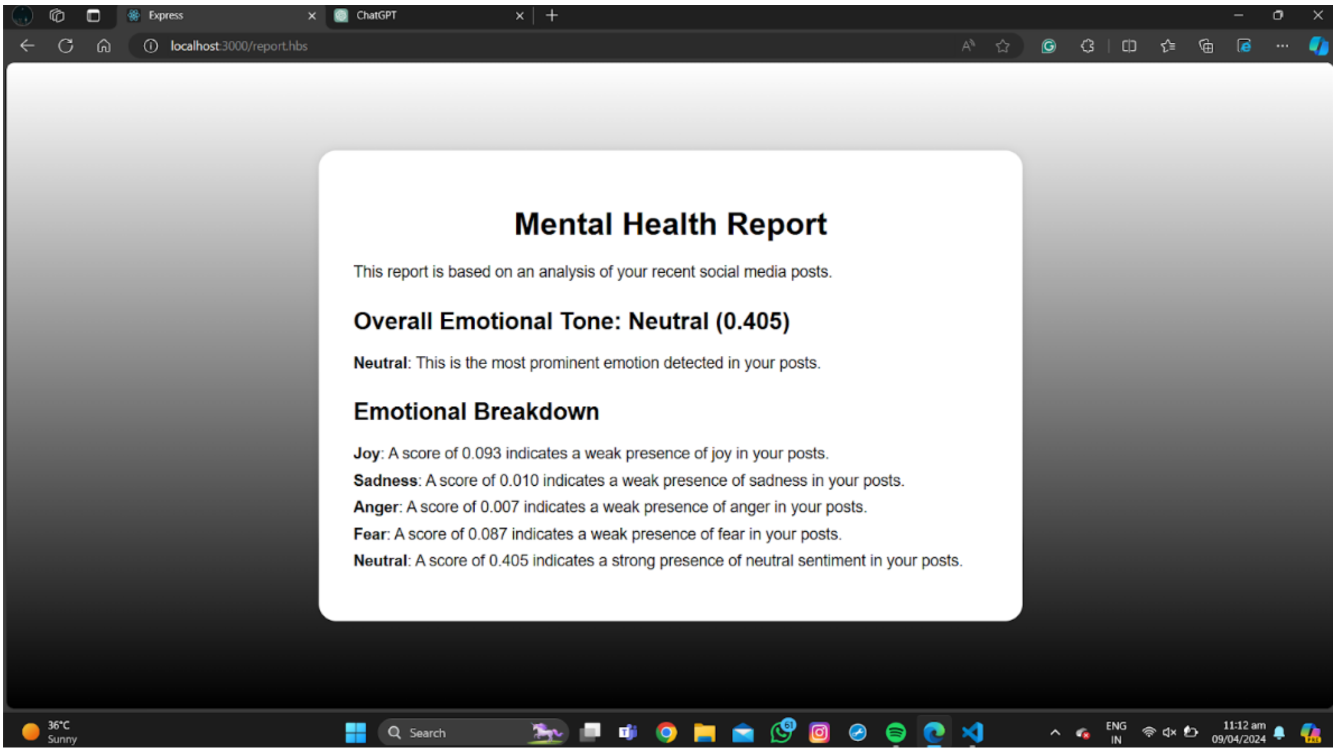


Figure 5.4: Final mental health report

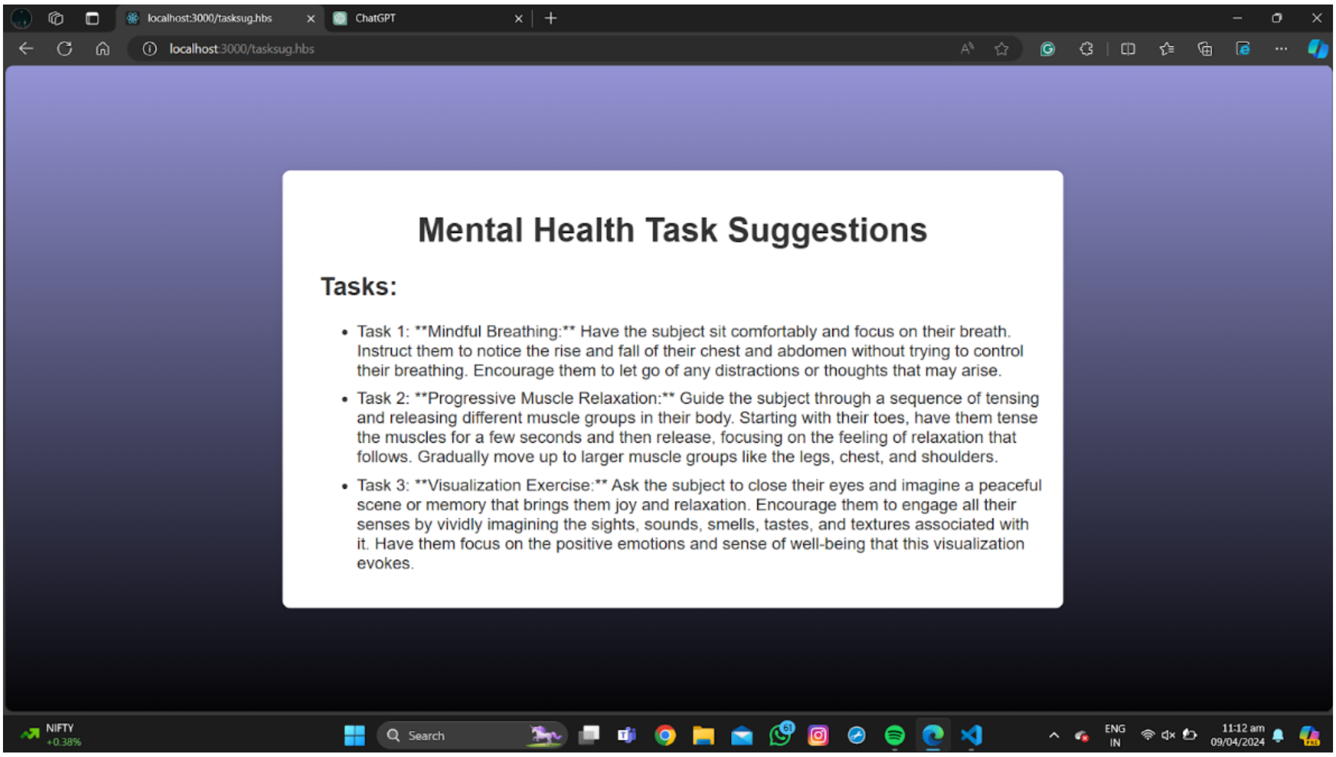


Figure 5.5: Task Suggestion

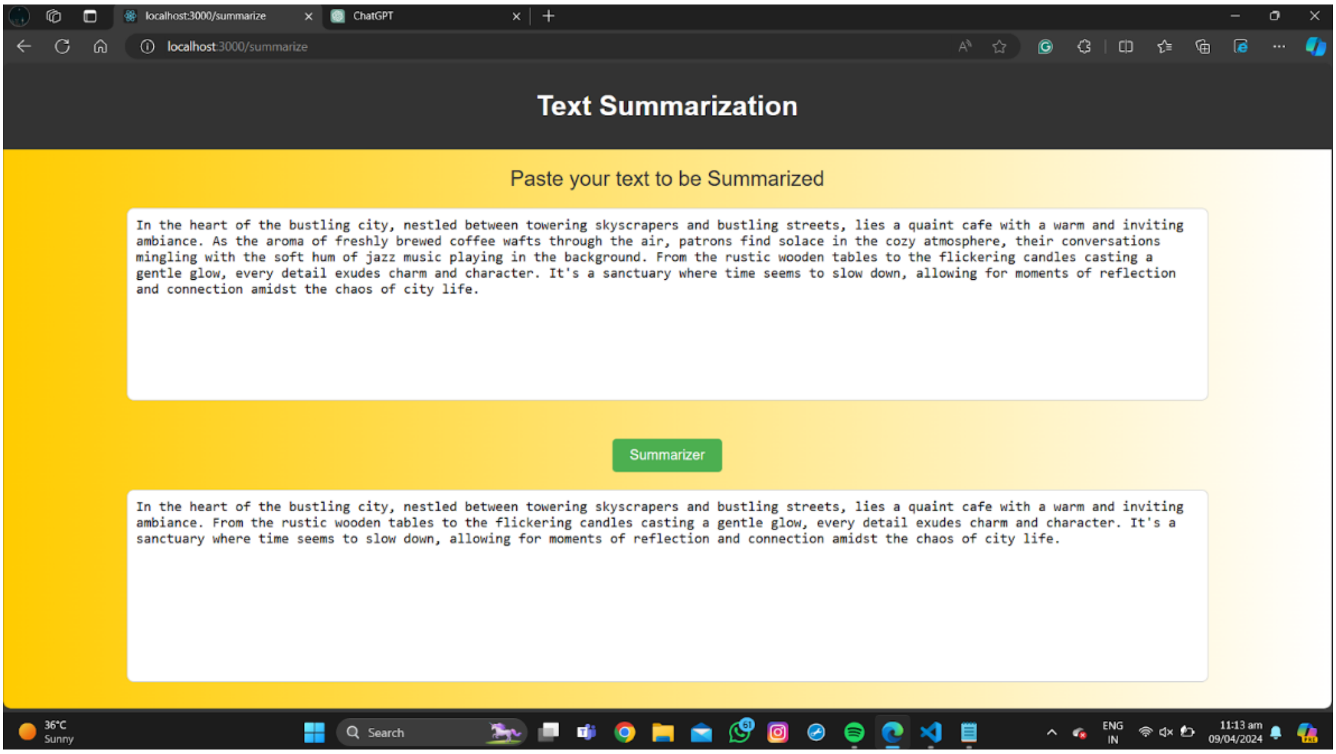


Figure 5.6: Text summarisation

5.4 Risks And Challanges

1. **Model Performance and Accuracy:** Making sure the BERT model performs and is accurate in identifying hate speech and detecting emotions is one of the main problems. Maintaining high accuracy rates might be difficult due to language use variability, context, and the dynamic nature of online speech. To meet this issue, ongoing model evaluation and enhancement are crucial.

2. **Data Privacy and Security:** Concerns regarding data security and privacy arise when user-generated text is collected and processed via social media networks. It is imperative to guarantee adherence to pertinent data protection laws (such the CCPA or GDPR) and to have strong security measures in place to shield user data from breaches or illegal access.

3. **User Engagement and Adoption:** It is difficult to persuade people to download and utilise the browser extension and to frequently visit the website where they can get their mental health report. It is possible to boost user engagement and adoption rates by offering a smooth and beneficial user experience, which includes transparent communication about the advantages of the tool and how user data is handled.

Chapter 6

Conclusions & Future Scope

In conclusion, the proposed browser extension represents a groundbreaking advancement in leveraging text analysis for multifaceted purposes, including emotion detection, hate speech identification, and text summarization. By seamlessly integrating these features, the extension offers users personalized mental health reports, fostering self-awareness and contributing to online safety. The incorporation of a summarization feature not only enhances user convenience but also streamlines the vast amounts of textual information encountered online. Moreover, the inclusion of task recommendations based on mental health reports provides actionable insights, creating a comprehensive solution for users seeking a positive online experience and enhanced well-being.

This innovative project is driven by the overarching goal of creating a tool capable of discerning users' mental states through textual analysis of their social media interactions. Beyond the crucial aspect of mental health awareness, the extension also addresses the pervasive issue of hate speech in social media, aiming to detect and control such harmful content effectively. The dual functionality of text summarization and task recommendations further enriches the user experience, promoting not only emotional well-being but also increased productivity and engagement.

1. Enhanced Language Support: Expand the project's capabilities by incorporating additional languages to cater to a more diverse user base, ensuring accurate emotion detection, hate speech identification, and summarization across various linguistic contexts.

2. Real-time Monitoring and Intervention: Develop real-time monitoring features that provide instant feedback to users, allowing for timely intervention and support. Implement features that can guide users towards positive online interactions and well-being in real-time.

3. Algorithmic Refinement: Future developments in the project could focus on refining the underlying algorithms for emotion detection, hate speech identification, and text summarization. Continuous research and improvements in natural language processing techniques may enhance the accuracy and effectiveness of the browser extension.

4. Expanded Platform Compatibility: To broaden its impact, the project could explore compatibility with additional social media platforms and online environments. Adapting the extension to various platforms would increase its reach, enabling a larger user community to benefit from the integrated features for mental health awareness and online safety.

5. Collaboration with Mental Health Professionals: Collaborating with mental health professionals and experts could enrich the project's capabilities. By integrating expert insights, the extension could provide more nuanced and context-aware mental health reports, offering users a more comprehensive understanding of their emotional well-being based on professional input.

References

[1] A. R. Abas, I. Elhenawy, M. Zidan, and M. Othman, "Bert-cnn: A deep learning model for detecting emotions from text." *Computers, Materials & Continua*, vol. 71, no. 2, 2022.

[2] P. Kumar and B. Raman, "A bert based dual-channel explainable text emotion recognition system," *Neural Networks*, vol. 150, pp. 392–407, 2022.

[3] S. K. S. Gaurav Rajput, Narinder Singh punn and S. Agarwal, "Hate speech detection using static bert embeddings," pp. 1–11, 2021.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[5] J. Wang, Z. Wang, D. Zhang, and J. Yan, "Combining knowledge with deep convolutional neural networks for short text classification." in *IJCAI*, vol. 350, no. 10.5555, 2017, pp. 3 172 077–3 172 295.

[6] A. R. Abas, I. El-Henawy, H. Mohamed, and A. Abdellatif, "Deep learning model for fine-grained aspect-based opinion mining," *IEEE Access*, vol. 8, pp. 128 845–128 855, 2020.

[7] I. N. Khasanah, "Sentiment classification using fasttext embedding and deep learning model," *Procedia Computer Science*, vol. 189, pp. 343–350, 2021.

[8] M. S. Saputri, R. Mahendra, and M. Adriani, "Emotion classification on indonesian twitter dataset," in *2018 International Conference on Asian Language Processing (IALP)*. IEEE, 2018, pp. 90–95.

[9] T. Daouas and H. Lejmi, "Emotions recognition in an intelligent elearning environment," *Interactive Learning Environments*, vol. 26, no. 8, pp. 991–1009, 2018.

[10] M. A. Riza and N. Charibaldi, “Emotion detection in twitter social media using long short-term memory (lstm) and fast text,” *International Journal of Artificial Intelligence & Robotics (IJAIR)*, vol. 3, no. 1, pp. 15–26, 2021.

[11] S. Agarwal, A. Sonawane, and C. R. Chowdary, “Accelerating automatic hate speech detection using parallelized ensemble learning models,” *Expert Systems with Applications*, p. 120564, 2023.

[12] W. Xing and A. Ghorbani, “Weighted pagerank algorithm,” in *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004.* IEEE, 2004, pp. 305–314.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

Appendix A: Presentation

100% CODE EVALUATION-EMOSENSE

PROBLEM DEFINITION

- Develop a tool capable of extracting text data from the user's social media activity and process it using BERT based emotion detection and hate speech detection modules to provide the user with a detailed report of how his/her mental state has been affected by social.

OBJECTIVES OF THE PROJECT

- Mental state recognition - The primary aim of the project is to recognize the user’s mental state from his/her online activity (social media and browser activity) using a extension.

- Mental state report generation - From the results of the first stage, a detailed mental state report is generated that can be accessed by the user through the website dedicated to the project.

- Hate Speech recognition - We aim to detect hate speech from the user’s side in his/her online activity.

OBJECTIVE OF THE PROJECT

- We aim to implement a task suggestion module that can recommend tasks to the user based on his/her mental state report.

- Text summarization feature will also be integrated in the plugin. This can be used to summarize long paragraphs we encounter on the internet.

NOVELTY AND SCOPE OF PROJECT

- Task suggestion based on the detected mood can help alleviate the mental state and enhances the overall quality of the day.
- Minimises chances of potential loss of reputation associated with emotional outbursts.
- We aim to detect hate speech from the user's side in his/her online activity.

GANTT CHART

Modules	December	January	February	March
Browser Extension	Dec 1 - 15			
Website	Dec 1 - 15			
Webserver & Database	Dec 16 – Jan 4		Feb 1 - 14	
Emotion Detection			Feb 1 - 28	
Hate Speech Detection			Feb 1 - 28	
Report Generation and Task Reccommendation				March 1 - 30
Text Summarization			Feb 15 – March 15	

WORK DONE DURING 30% EVALUATION

- Developed an extension capable of extracting text message from various social media websites.

- The extracted text was stored in a MySQL database for further processing.

- Developed the front-end of the website facilitating sign up and log in.

WORK DONE DURING 60% EVALUATION

- Completed the development of emotion detection and hate speech detection module.

- Both models were developed by fine tuning BERT.

- Accuracy of the models were:
    - Emotion detection - >80%
    - Hate speech detection - >80%

Both models were successfully loaded into our local machine.

WORK DONE DURING 60% EVALUATION

- Established connectivity between the trained models and the extension developed for the project.

- Both models are now capable of retrieving the data stored by the extension in the project database and make predictions on that data.

WORK PROGRESS

- Finished the report generation module by giving different weights to different social media platforms to generate a comprehensive mental state report.

- Developed a task recommendation system by making use of Google Gemini via API calls.

- Developed and integrated a summarizer module into the website by making use of API.

WORK PROGRESS

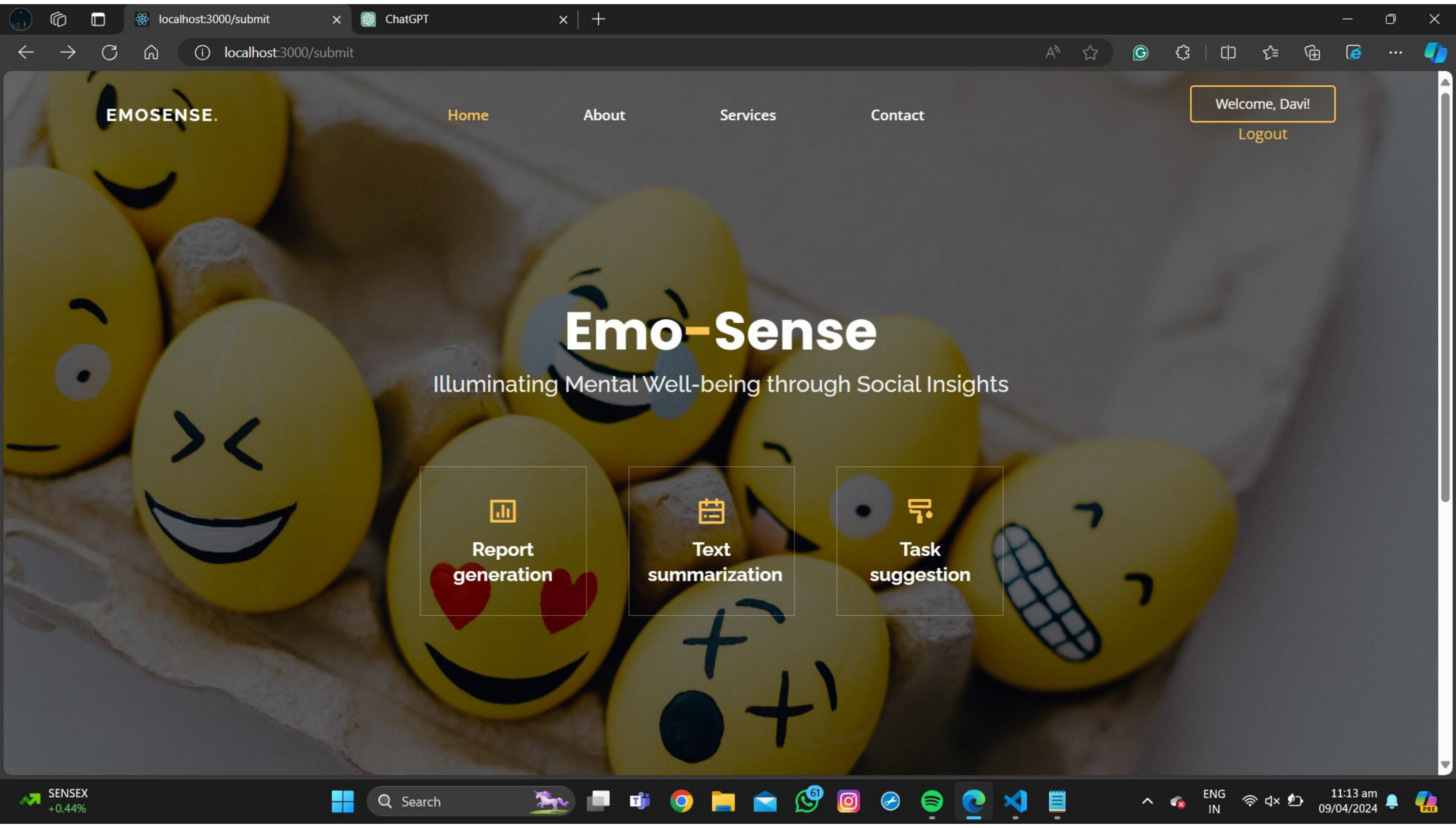
- It was found that users prefer to share positive emotions on instagram compared to facebook and reddit.
- Likewise, negative emotions were more openly shared on facebook.
- This was taken into consideration while generating the final mental state report.

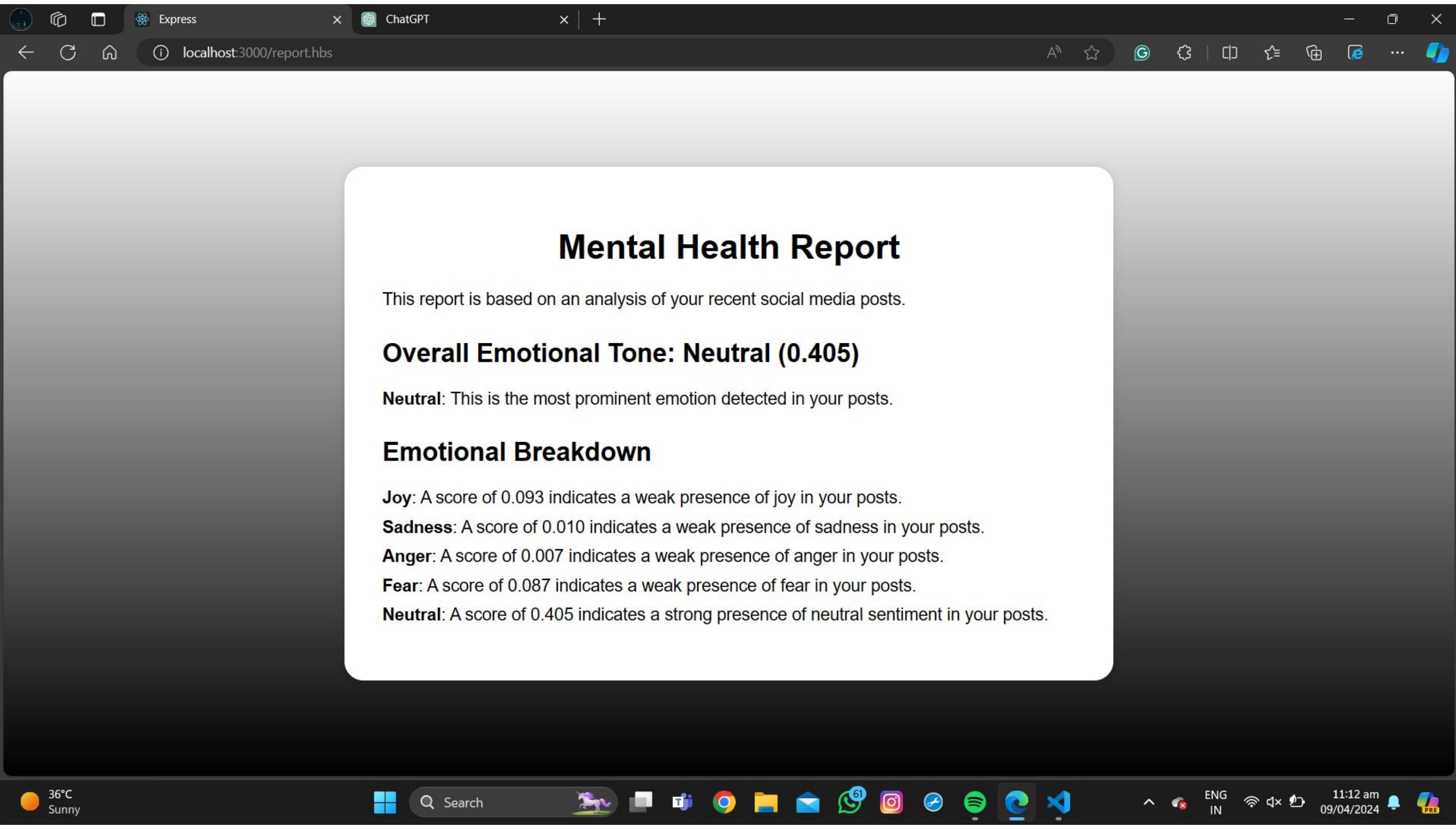
RESULTS

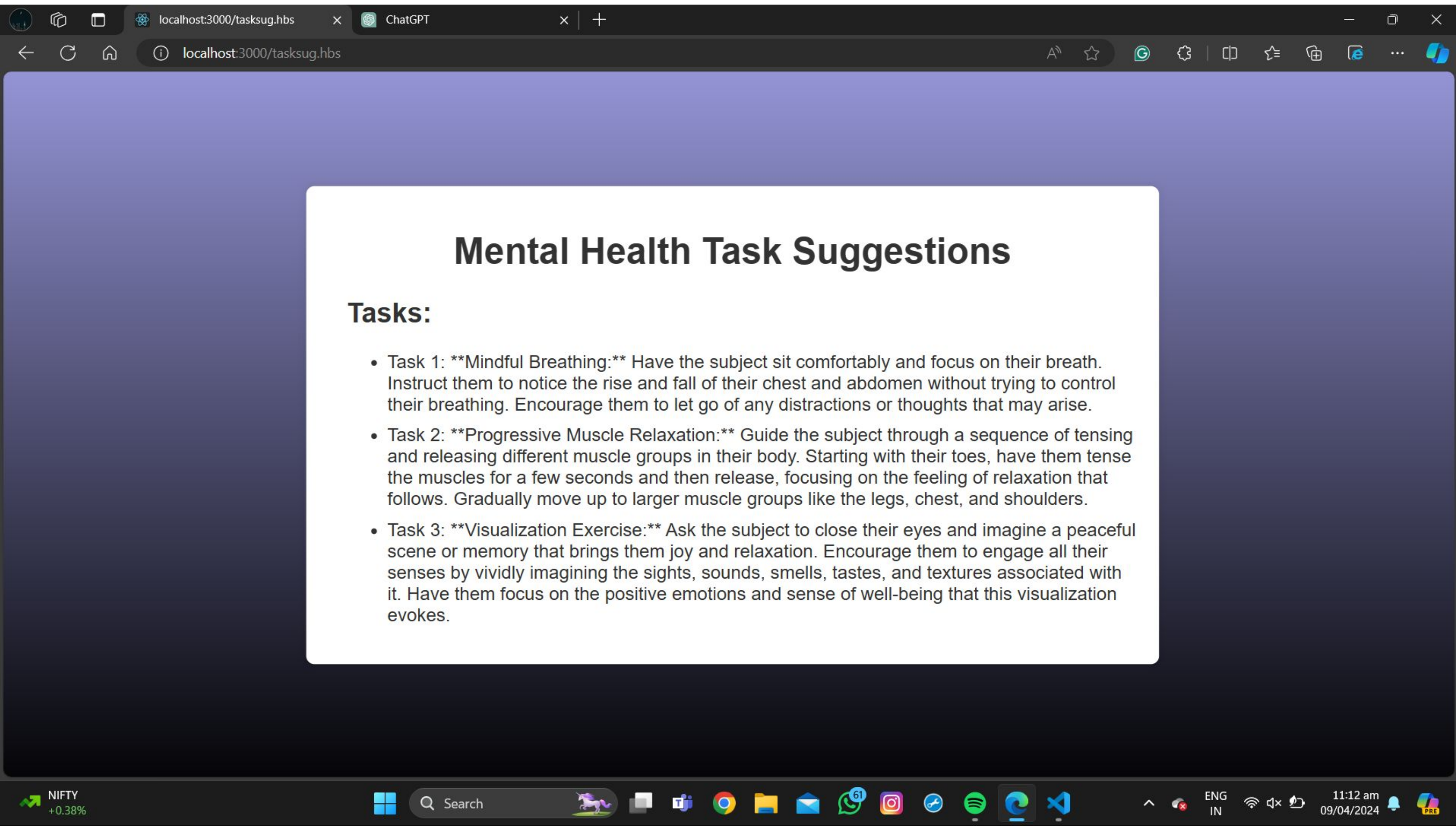
```
mysql> select * from insta_data;
+-----------------------------------+--------------------------+
| text                              | timestamp                |
+-----------------------------------+--------------------------+
| hello cristiano, i am a huge fan  | 2024-02-21T09:10:37.089Z |
+-----------------------------------+--------------------------+
1 row in set (0.00 sec)
```

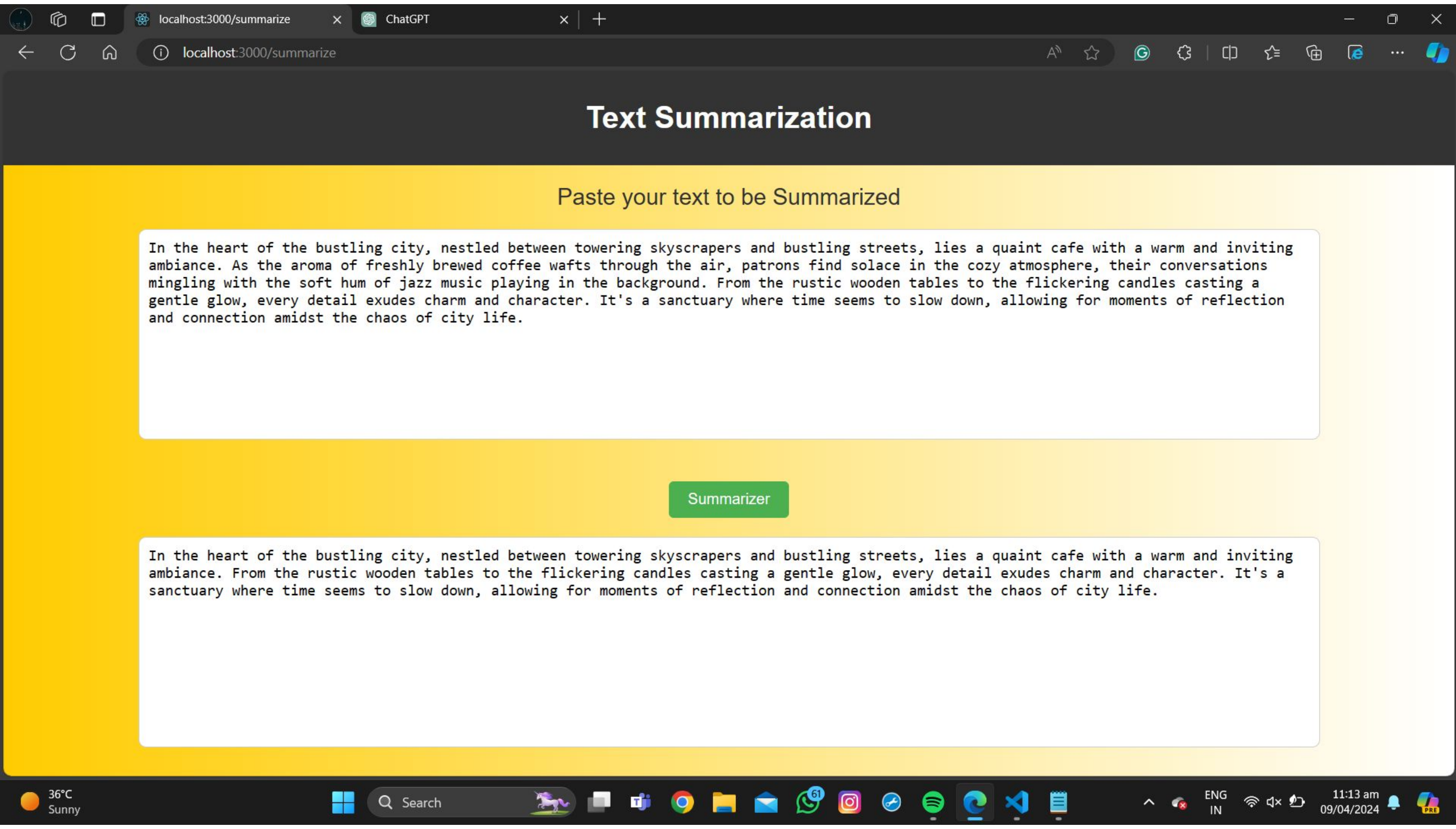
```
ect * from insta_results;
---+-----------+---------------+-------------+-------------+-------------+
   | joy       | sadness       | fear        | anger       | neutral     |
---+-----------+---------------+-------------+-------------+-------------+
21 | 0.9586112 | 0.00096629077 | 0.019724492 | 0.002208148 | 0.018489823 |
---+-----------+---------------+-------------+-------------+-------------+
et (0.00 sec)
```

```
mysql> select * from insta_hate;
+---------------------+----------------------+-------------+
| date                | text                 | hate_speech |
+---------------------+----------------------+-------------+
| 2024-02-21 21:56:10 | hello how are you    | Neither     |
| 2024-02-22 09:05:50 | good game yesterday  | Neither     |
+---------------------+----------------------+-------------+
2 rows in set (0.00 sec)
```




Express
ChatGPT
localhost:3000/report.hbs
Mental Health Report
This report is based on an analysis of your recent social media posts.
Overall Emotional Tone: Neutral (0.405)
Neutral: This is the most prominent emotion detected in your posts.
Emotional Breakdown
Joy: A score of 0.093 indicates a weak presence of joy in your posts.
Sadness: A score of 0.010 indicates a weak presence of sadness in your posts.
Anger: A score of 0.007 indicates a weak presence of anger in your posts.
Fear: A score of 0.087 indicates a weak presence of fear in your posts.
Neutral: A score of 0.405 indicates a strong presence of neutral sentiment in your posts.
36°C
Sunny
Search
ENG
IN
11:12 am
09/04/2024


localhost:3000/tasksug.hbs
ChatGPT
localhost:3000/tasksug.hbs
Mental Health Task Suggestions
Tasks:
Task 1: **Mindful Breathing:** Have the subject sit comfortably and focus on their breath. Instruct them to notice the rise and fall of their chest and abdomen without trying to control their breathing. Encourage them to let go of any distractions or thoughts that may arise.
Task 2: **Progressive Muscle Relaxation:** Guide the subject through a sequence of tensing and releasing different muscle groups in their body. Starting with their toes, have them tense the muscles for a few seconds and then release, focusing on the feeling of relaxation that follows. Gradually move up to larger muscle groups like the legs, chest, and shoulders.
Task 3: **Visualization Exercise:** Ask the subject to close their eyes and imagine a peaceful scene or memory that brings them joy and relaxation. Encourage them to engage all their senses by vividly imagining the sights, sounds, smells, tastes, and textures associated with it. Have them focus on the positive emotions and sense of well-being that this visualization evokes.
NIFTY
+0.38%
Search
ENG
IN
11:12 am
09/04/2024



FUTURE SCOPE

- Aid therapists in diagnosis of mental health ailments
- Multi language operability
- Real time hate speech detection and intervention

TASK DISTRIBUTION

- Web extension: Crispin Mathew
- Website: David Johns Denny
- Database: Devananth H
- Emotion detector: Crispin Mathew, David Johns Denny
- Hate Speech detector: Akshay Ajit, Devananth H
- Text summarizer: Akshay Ajit
- Report generator: David Johns Denny
- Task recommender: Crispin Mathew

CONCLUSION

- All the proposed modules – web extension, emotion detector, hate speech detector, text summarizer, report generator and task recommender have been successfully developed

- Integration of the modules into the web server yet to be completed.

- Researched how users prefer different social media platforms to share varying emotions and generated the mental state report accordingly.

REFERENCES

1. Kumar, P., & Raman, B. (2022). A BERT based dual-channel explainable text emotion recognition system. Neural Networks, 150, 392-407.

2. Riza, M. A., & Charibaldi, N. (2021). Emotion Detection in Twitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text. International Journal of Artificial Intelligence & Robotics (IJAIR), 3(1), 15-26.

3. Agarwal, S., & Chowdary, C. R. (2021). Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19. Expert Systems with Applications, 185, 115632.

4. Rajput, G., Punn, N. S., Sonbhadra, S. K., & Agarwal, S. (2021). Hate speech detection using static bert embeddings. In *Big Data Analytics: 9th International Conference, BDA 2021, Virtual Event, December 15-18, 2021, Proceedings 9* (pp. 67-77). Springer International Publishing.

STATUS OF PAPER PUBLICATION

Currently writing a research paper to be submitted to a conference on the 15th of April.

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills
The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills
The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills
The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes (CO)

Course Outcome 1: Model and solve real world problems by applying knowledge across domains (Cognitive knowledge level: Apply).

Course Outcome 2: Develop products, processes or technologies for sustainable and socially relevant applications (Cognitive knowledge level: Apply).

Course Outcome 3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: Apply).

Course Outcome 4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: Apply).

Course Outcome 5: Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: Analyze).

Course Outcome 6: Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: Apply).

Appendix C: CO-PO-PSO Mapping

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO 1	2	2	2	1	2	2	2	1	1	1	1	2	3		
CO 2	2	2	2		1	3	3	1	1		1	1		2	
CO 3									3	2	2	1			3
CO 4					2			3	2	2	3	2			3
CO 5	2	3	3	1	2							1	3		
CO 6					2			2	2	3	1	1			3

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/ MEDIUM/HIGH	JUSTIFICATION
100003/ CS722U.1-PO1	M	Knowledge in the area of technology for project development using various tools results in better modeling.
100003/ CS722U.1-PO2	M	Knowledge acquired in the selected area of project development can be used to identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions.

100003/ CS722U.1-PO3	M	Can use the acquired knowledge in designing solutions to complex problems.
100003/ CS722U.1-PO4	M	Can use the acquired knowledge in designing solutions to complex problems.
100003/ CS722U.1-PO5	H	Students are able to interpret, improve and redefine technical aspects for design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/ CS722U.1-PO6	M	Students are able to interpret, improve and redefine technical aspects by applying contextual knowledge to assess societal, health and consequential responsibilities relevant to professional engineering practices.
100003/ CS722U.1-PO7	M	Project development based on societal and environmental context solution identification is the need for sustainable development.
100003/ CS722U.1-PO8	L	Project development should be based on professional ethics and responsibilities.
100003/ CS722U.1-PO9	L	Project development using a systematic approach based on well defined principles will result in teamwork.
100003/ CS722U.1-PO10	M	Project brings technological changes in society.

100003/ CS722U.1-PO11	H	Acquiring knowledge for project development gathers skills in design, analysis, development and implementation of algorithms.
100003/ CS722U.1-PO12	H	Knowledge for project development contributes engineering skills in computing & information gatherings.
100003/ CS722U.2-PO1	H	Knowledge acquired for project development will also include systematic planning, developing, testing and implementation in computer science solutions in various domains.
100003/ CS722U.2-PO2	H	Project design and development using a systematic approach brings knowledge in mathematics and engineering fundamentals.
100003/ CS722U.2-PO3	H	Identifying, formulating and analyzing the project results in a systematic approach.
100003/ CS722U.2-PO5	H	Systematic approach is the tip for solving complex problems in various domains.
100003/ CS722U.2-PO6	H	Systematic approach in the technical and design aspects provide valid conclusions.
100003/ CS722U.2-PO7	H	Systematic approach in the technical and design aspects demonstrate the knowledge of sustainable development.

100003/ CS722U.2-PO8	M	Identification and justification of technical aspects of project development demonstrates the need for sustainable development.
100003/ CS722U.2-PO9	H	Apply professional ethics and responsibilities in engineering practice of development.
100003/ CS722U.2-PO11	H	Systematic approach also includes effective reporting and documentation which gives clear instructions.
100003/ CS722U.2-PO12	M	Project development using a systematic approach based on well defined principles will result in better teamwork.
100003/ CS722U.3-PO9	H	Project development as a team brings the ability to engage in independent and lifelong learning.
100003/ CS722U.3-PO10	H	Identification, formulation and justification in technical aspects will be based on acquiring skills in design and development of algorithms.
100003/ CS722U.3-PO11	H	Identification, formulation and justification in technical aspects provides the betterment of life in various domains.
100003/ CS722U.3-PO12	H	Students are able to interpret, improve and redefine technical aspects with mathematics, science and engineering fundamentals for the solutions of complex

		problems.
100003/ CS722U.4-PO5	H	Students are able to interpret, improve and redefine technical aspects with identification formulation and analysis of complex problems.
100003/ CS722U.4-PO8	H	Students are able to interpret, improve and redefine technical aspects to meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
100003/ CS722U.4-PO9	H	Students are able to interpret, improve and redefine technical aspects for design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/ CS722U.4-PO10	H	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools for better products.
100003/ CS722U.4-PO11	M	Students are able to interpret, improve and redefine technical aspects by applying contextual knowledge to assess societal, health and consequential responsibilities relevant to professional engineering practices.
100003/ CS722U.4-PO12	H	Students are able to interpret, improve and redefine technical aspects for demonstrating the knowledge of, and need for sustainable development.

100003/ CS722U.5- PO1	H	Students are able to interpret, improve and redefine technical aspects, apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
100003/ CS722U.5- PO2	M	Students are able to interpret, improve and redefine technical aspects, communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
100003/ CS722U.5- PO3	H	Students are able to interpret, improve and redefine technical aspects to demonstrate knowledge and understanding of the engineering and management principle in multidisciplinary environments.
100003/ CS722U.5- PO4	H	Students are able to interpret, improve and redefine technical aspects, recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
100003/ CS722U.5- PO5	M	Students are able to interpret, improve and redefine technical aspects in acquiring skills to design, analyze and develop algorithms and implement those using high-level programming languages.
100003/ CS722U.5- PO12	M	Students are able to interpret, improve and redefine

		technical aspects and contribute their engineering skills in computing and information engineering domains like network design and administration, database design and knowledge engineering.
100003/ CS722U.6-PO5	M	Students are able to interpret, improve and redefine technical aspects and develop strong skills in systematic planning, developing, testing, implementing and providing IT solutions for different domains which helps in the betterment of life.
100003/ CS722U.6-PO8	H	Students will be able to associate with a team as an effective team player for the development of technical projects by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/ CS722U.6-PO9	H	Students will be able to associate with a team as an effective team player to Identify, formulate, review research literature, and analyze complex engineering problems
100003/ CS722U.6-PO10	M	Students will be able to associate with a team as an effective team player for designing solutions to complex engineering problems and design system components.
100003/ CS722U.6-PO11	M	Students will be able to associate with a team as an effective team player, use research-based knowledge and research methods including design of experiments, analysis and interpretation of data.

100003/ CS722U.6-PO12	H	Students will be able to associate with a team as an effective team player, applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
100003/ CS722U.1-PSO1	H	Students are able to develop Computer Science Specific Skills by modeling and solving problems.
100003/ CS722U.2-PSO2	M	Developing products, processes or technologies for sustainable and socially relevant applications can promote Programming and Software Development Skills.
100003/ CS722U.3-PSO3	H	Working in a team can result in the effective development of Professional Skills.
100003/ CS722U.4-PSO3	H	Planning and scheduling can result in the effective development of Professional Skills.
100003/ CS722U.5-PSO1	H	Students are able to develop Computer Science Specific Skills by creating innovative solutions to problems.
100003/ CS722U.6-PSO3	H	Organizing and communicating technical and scientific findings can help in the effective development of Professional Skills.