Harnessing Emotions Using Language Processing in detecting cyberbullying

By

CHEN KOK CHUNG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS)
INFORMATION SYSTEMS ENGINEERING
Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

This Final Year Project proposal is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Systems (Honours) Information Systems Engineering at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project proposal represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project proposal may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

I would like to extend my sincerest appreciation and acknowledgement to my supervisor, Ts Dr Tong Dong Ling, for her superb mentorship, encouragement, and constant support throughout my Final Year Project, titled "Harnessing Emotions Using Language Processing to Detect Cyberbullying." This project has been an incredible experience that has allowed me to explore the areas of natural language processing, sentiment analysis, and deep learning, thus enhancing my academic as well as personal growth considerably.

Finally, I would like to extend my deepest gratitude to my family and parents for their unconditional love, support, and constant encouragement throughout my studies. This success would not have been possible without their efforts.

ABSTRACT

Cyberbullying represents a widespread challenge across social media platforms, frequently resulting in considerable emotional and psychological distress for individuals. Although current detection systems are geared towards recognizing harmful language, they fall short in comprehensively understanding the emotional consequences of such content. This project introduces Advanced Emotion Detection, a system aimed at categorizing particular emotions and assessing their intensity within comments on social media. This project centers on Instagram, a platform characterized by visual and textual interactions that often result in cyberbullying, with the objective of refining a large language model (LLM) by utilizing data obtained from publicly available posts and comments. The final dataset collected will be processed and used to fine-tune an LLM to locate subtle expressions of emotions within text. The system will go further than the basic sentiment analysis in detecting the severity and type of emotional impact, thus allowing the correct cyberbullying incident classification. The outcome of the project is to create a fine-tuned LLM model that capable to detect and classify the severity and types of cyberbully emotional impact.

Area of Study: Sentimental Analysis, Large Language Model

Keywords: Cyberbullying, Fine-tuning, web scrapping, Social Media Analysis, Text Classification.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AUC	Area Under the Curve
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
IDE	Integrated Development Environment
KNN	K-Nearest Neighbors
LLM	Large Language Model
LSTM	Long Short-Term Memory
MLM	Masked Language Modeling
NB	Naïve Bayes
NLP	Natural Language Processing
NSP	Next Sentence Prediction
PMI	Pointwise Mutual Information
RF	Random Forest
RoBERTa	Robustly Optimized BERT Pretraining Approach
SA	Sentimental Analysis
SHAP	SHapley Additive exPlanations
SVM	Support Vector Machine
TF-IDF	Term Frequency–Inverse Document Frequency
TPR	True Positive Rate
XAI	Explainable Artificial Intelligence

CHAPTER 1

Introduction

Social media has finally altered how people communicate and receive information around the world, but with all its benefits, it has also raised new concerns of harassment in view-cyberbullying. Cyberbullying is a pervasive issue that causes serious emotional and psychological harm to its targets. Most incidents of cyberbullying contain subtle and context-dependent forms of toxic behavior; hence, the detection and mitigation are complex. Cyberbullying is more commonly occurring among teenagers and children than other age groups since in fact a quarter of children and teens have cyberbullied someone in the life in the last five year in 2021. Statistics shown 16 percent of high school student encounter bullies electronically in a year, the girls have 20 percent more chance to encounter cyberbully online than the boys which has 11 percent [1]. The most recent approaches toward the detection of cyberbullying have put their focus mainly on offensive language or the polarity of sentiment, such as positive, negative, or neutral [2]. Although such systems show some promise, they fail to capture the fine-grained emotional effects related to vicious comments. Feelings related to fear, anger, and humiliation, often accompanying cyberbullying incidents, are either ignored or poorly categorized. The identification of these emotions and their strength is critical for an effective strategy toward the issue at hand and providing useful feedback for both victims and administrators. A new intervention is proposed project will make use of publicly available data from Instagram to fine-tune a Large Language Model in competently identifying and classifying various classes of emotions throughout textual interactions. Instagram is a domain that holds a peculiar juxtaposition between visual and textual discourse; hence, it becomes a fitting context to consider emotionally charged interaction, which frequently leads to incidents of cyberbullying.

1.1 Problem Statement and Motivation

Cyberbullying constitutes a sophisticated behavior, including not only direct insults and obscenities but even more sophisticated forms such as sarcasm, irony, and emotional nuance. Most modern-day techniques rely on static keyword-based approaches or lexicon-dependent sentiment analysis that seek to detect explicit markers of antagonism. However, such techniques fall short when dealing with communications that use implicit cues. For instance, a tweet full of sarcasm can include individually inoffensive terms when analyzed in a vacuum, yet together, in concert, they can convey hurt and contempt. Such a weakness generates high false-negative rates—situations in which damaging communications go undetected, depriving victims of timely intervention during critical moments. Such a weakness is stressed in current work: Van Hee et al. [3] showed that explicit cases of bullying can effectively be detected with use of linear classifiers with n-gram features, yet such approaches miss out on sophisticated emotional nuance that characterizes many cases of cyberbullying. In addition, studies comparing emotion-sensitive approaches reveal that including sentiment and affect analysis can deliver significant performance improvements, through increased sensitivity to the faint markers that traditional approaches miss out on. By focusing on sophisticated emotional awareness—using deep learning techniques that model both polarity and intensity of emotion—substantial improvements in the accuracy of cyberbullying detection algorithms can be achieved. Application of such a system enables concealed malice to be detected and reported, even in cases of no explicit terminologies of bullying, providing increased protection for high-risk groups.

Cyberbullying is not an individualistic phenomenon; instead, it occurs in a multifaceted social context with multiple participants. Current detection methods are prone to examining each message in a vacuum, without taking into account larger conversational structures and the various roles participants take on—e.g., aggressors, victims, and observers—within the developing incidents of cyberbullying. A single message can be misinterpreted when it is not considered in conjunction with its antecedent and subsequent messages. For instance, a seemingly harmless remark can be interpreted as aggressive or bullying in the context of a heated exchange. In addition, the implications and severity of cyberbullying can vary greatly depending on an individual's role; what may be interpreted as a harmless remark by an observer may be seen as a deliberate attack when uttered by a recognized bully. Empirical studies have supported that the inclusion of

contextual and role information in detection algorithms improves their performance. Research by Zhao et al. [2] and other researchers has investigated methods that combine networked information with participant role labels, establishing that knowledge of relational dynamics and conversational structures is essential for distinguishing between innocuous discourse and abusive bullying behavior. By combining conversational history with role awareness, sophisticated algorithms can more effectively detect "hidden" indicators of cyberbullying with greater accuracy. This method not only seeks to enhance detection accuracy but also allows for the prioritization of cases in which potential threats may be compounded by multi-participant interactions, sequences of escalating aggression, and other such situations. Thus, context-aware detection algorithms are essential in providing a more informed and timely intervention strategy, leading to a safer online environment.

1.2 Research Objectives

The aim of the thesis is to propose a SA based cyberbullying detection system by using fine-tuned Large Language Models in order to analyse the textual content for the variations in the emotional impact of social media, taking Instagram as the case study. The objectives are:

1. Design and deploy a cyberbullying detection system within scheduled project time which outperforms keyword-based systems through the use of advanced sentiment analysis combined with emotion detection by LLMs.

2. Develop a classification system that divides the different forms of disparate manifestations of cyberbullying, which are insults, threats, and defamation, using textual characteristics.

3. Design and implementing a system that not only includes isolated message but also the overall conversation.

1.3 Project Scope and Direction

This current goal of this project is to develop a cyberbullying detection that transcends limitations of traditional keyword-based approaches through the use of advanced sentiment and emotion analysis models. It further considers the broad conversation context and participant roles within online interactions. Instagram is the main subject of exploration due to its high volume of user-generated data and proneness to emotional exchanges between its users. English captions and comments are the main focus of this system since it does not process image-based content or video-based content nor multilingual content.

The project aims to design a cyberbullying detection tool that builds upon traditional keyword-based methods by combining sophisticated sentiment analysis along with emotion detection mechanisms. This tool will be particularly designed to scrutinize text data obtained from Instagram, such as captions, comments, and replies, due to the vast use of the platform and the high rate of emotionally charged interactions that tend to include cyberbullying instances.

The project involves in creating a simple user dashboard to gain insight of the cyberbullying topic and generation of charts and graph to show the trend among the post for further analysing. The dashboard also will be act as the cyberbullying detection interface.

The project involves the application of advanced Large Language Models (LLMs) that can recognize not only overt expressions of bullying but all manner of subtle, implicit forms of emotional abuse, like irony, sarcasm, and passive aggression. These models will be trained to evaluate the intensity and polarity of the emotions expressed in text, thus gaining further insight into the emotional impact that is involved in online communications.

Additionally, the project focus to create a classification system that measures episodes in cyberbullying based on the resulting psychological impact. This approach will enable the

prioritization of the cases that require more prompt action due to the severity of the emotional harm suffered.

In addition, the system is designed to examine not only individual messages independently but also to investigate the wider conversational context of exchanges, specifically sequences of messages, and identify the social roles the interaction partners take, such as aggressor, victim, or bystander. With the combination of conversational context and social dynamics, the system aims to make detections that are not only more accurate but sensitive to context nuances.

The data collection period should lead to collecting about 10,000 Instagram comments for model training and evaluation. The entire project data will come from self-collected web scraping operations instead of using public external datasets. The research gathers Instagram comments from specifically selected posts through a combination of HAR files and Stevesie software which runs on Google search results. The platform limitations restrict the data to minimal volumes and reduced diversity since it only gathers accessible public content.

The direction of the project involves several key phases. These include conducting a literature review on existing methods of cyberbullying detection and sentiment analysis, collecting and preprocessing relevant Instagram data, and fine-tuning a suitable LLM for emotion-aware and context-sensitive detection. The system will be implemented and tested using appropriate evaluation metrics, such as precision, recall, and F1-score. Finally, the project will be documented in a comprehensive report detailing the development process, system architecture, results, and potential areas for future improvement.

1.4 Contributions

This paper proposes a cyberbullying detection mechanism that moves beyond traditional keyword methods utilizing fine-tuned large language models like RoBERTa to enable efficient sentiment and affect analysis. These models are capable of detecting subtle emotional cues such as implicit hostility and sarcasm, which are mostly not detected by static approaches. This approach aligns with the work that has been done by Ptaszynski et al. [5], which proved to enhance the effectiveness of cyberbullying detection with the integration of emotional processing.

A unique aspect of this research project is the creation of classification methodology that arranges incidences of cyberbullying based on the emotional impact that each present. This methodology assists policymakers in prioritizing the more severe incidences to be acted upon quickly. Evidence acquired through research by Salawu et al. [6] supports the fact that emotional parameters drastically improve detection accuracy and assist with determining the severity of cyber abuse.

The detection process incorporates not only the isolated, individual discrete messages but also the overall conversational context, along with the different participant roles played, i.e., aggressors, victims, and bystanders. This process improves the detection accuracy of patterns of escalating abuse and covered aggression. Zhao et al.'s [4] research has proved that the combination of participant information with the context of the conversations greatly improves detection performance under real-world settings.

The system has been carefully developed for Instagram, which presents unique challenges due to the casual and largely visual nature of its content. By focusing on Instagram-specific captions and comments, the system offers a more targeted and contextually relevant detection system compared to more general approaches.

1.5 Report Organization

This report is organized to provide a comprehensive overview of the project of Harnessing Emotions Using Language Processing in detecting cyberbullying. Chapter one, Introduction, describe the project background, problem statement, objectives, scope, direction and contribution. This chapter also outline the limitations of current methods of detection of cyberbullying, thus highlighting the need for an approach that is more sensitive to the emotional aspects and context.

Chapter two provides an integrated review of the relevant literature to the ongoing studies for the purposes of the detection of cyberbullying, emotion-aware systems, the measurement of conversational contexts, and the use of Large Language Models (LLMs). It critiques varied methodologies, tools, and data sets utilized in prior studies, highlighting the targeted research gaps to be bridged by this project.

Chapter three outlines the proposed approach and methodology. It discusses the system architecture, model improvement techniques, data acquisition and preprocessing strategies, and the use of emotion and context-sensitive detection. In addition, the chapter describes the proposed classifier to assess the severity of cyberbullying, along with the use of conversational context, and the interaction between the participants in the model.

Chapter four describe the system design for the proposed system, which shows the project system architecture in a systematic way, system modeling, deployment phase, and the dashboard wireframe. The system architecture design will provide a detailed architectural picture, which will explain how the data will be gathered, preprocessed, analysed, and then graphically represented. The modeling phase block diagram outlines the pipeline out the model training. For deployment phase block diagram explains how to implement the trained model to a real-time operational environment under which it can be used to make predictions and perform constant monitoring. To summarize, the dashboard wireframe will be a first-time design of the user interface that is

effective in presenting detection results and emotional insights in the format that is understandable and is user-friendly.

Chapter five outlines the implementation of the proposed system empirically by configuring and integrating the hardware, as well as software. The hardware arrangement outlines the computing capabilities required to run model training and deployment. The software configuration contains the details on the programming tools, frameworks, and libraries used in the implementation. Next, the system operation section shows each of the interface module and it function of the proposed system. Implementation issues is also mentioned in this chapter. Finally, a conclusion that summarizes the results of the implementation phase and pre-installs the basis of the further system evaluation.

Chapter six provides the assessment of the developed system and its efficiency to achieve the project goals. Testing set-up and results are described in the testing structure and results section as being the datasets, evaluation metrics, and experimental results that support the work of the system. The project challenges section is a reflection on the problems experienced during the evaluation and includes control of subtle emotional nuances and the resolution of precision versus recall. The objectives evaluation goes back to the aims of the project and evaluates how well each of the objectives was met in the implementation and testing process. Lastly, the final commentary is a summary of the evaluation results with the strengths and opportunities of improvement highlighted.

Chapter seven conclude and summarize the overall progress, emphasizing how the proposed system successfully, Next further provides recommendations for future work to make the system more transparent and practical for real-world deployment.

CHAPTER 2

Literature Reviews

2.1 Previous Works on Detection of Cyber-Bullying with Sentiment Analysis

2.1.1 Traditional Machine Learning Methods

The paper from [2] [7] [8] [9] conducted their research using traditional machine learning approaches. Traditional machine learning to perform detection of cyberbullying with SA on social media. Traditional machine learning methods, namely NB, SVM, and Random Forest, have been one of the most common ways to solve it.

a. Naive Bayes Classifier

The NB is a probabilistic grouping technique that relies on Bayes' theorem with strong independence assumptions between features. It has seen relatively wide application to text classification. The research in [8] focuses on developing a model that capable to perform classification of cyberbullying in comments on Instagram, the research used NB machine learning method. It also uses TF-IDF method for feature extraction to improve the model efficiency and accuracy. After the data collection, preprocessing and training. It is evaluated using K-Fold Cross Validation. It gives an average accuracy of 83.53% when stemming is applied to the pre-processing while it is 83% without stemming. The paper [9] also applies NB for the detection of cyberbullying. The paper selected Twitter as the social media dataset. The paper focuses on classification of severity of cyber bullying. In some works, it has been reported together with other techniques: PMI for feature extraction. The model performed with very promising results, although often outperformed by SVM concerning the accuracy of the results. The result of the paper shows that NB perform in result of 75% accuracy in multi-class settings.

b. Support Vector Machines (SVM)

SVM is a supervised learning model that analyse the data for classification and also regression analysis. It is especially effective in high-dimensional spaces and is robust to overfitting in cases where the number of dimensions more than the number of samples. SVM has seen extensive use in cyberbullying detection on Twitter. In the paper [7], SVM was found to outperform NB with higher n-grams features. SVM classifiers gave better performance measures in Accuracy, Precision, Recall, F-measure and Roc per N-gram. for the detection of cyberbullying texts when compared with NB. Another independent study [2] employed SVM in the prediction of sentiments of comments on YouTube. The classifier was embedded in a framework that first applied the optimization algorithms like AdaBoost and SGD to enhance the cyberbullying detection before applying SVM to classify the sentiments.

c. Random Forest

Random Forest is an ensemble learning method that is fit to be use for classification or regression task. It builds up of several decision trees in the course of training and delivering the mode of classes or the mean prediction of the single trees. The paper [9] study on the detection of the severity of cyberbullying, the paper tested on few machines learning algorithm and the most perform algorithm is the Random Forest with 90.363% after the base classifier model is improve with SMOTE to handled class imbalance distribution. Moreover, the final model that integrated with SMOTE, Embedding, Sentiment, Lexicon and PMI-SO features perform greatly compare with others model like SVM and NB. The model that used Random Forest perform 91.153% accuracy with 0.898 F-Measure. The dataset the model uses is taken from the publicly available git repository about harassment datasets.

2.1.2 Large Language Model Methods

The paper [10] [11] [12] conducted the research related to the transition from traditional machine learning approaches toward the adoption of large language models in detecting cyberbullying on different social media platform. Traditional machine learning methods do not manage to capture

the complexity of linguistic patterns or the context-dependent nature that abusive content carries with it. With the advent of LLMs like BERT, RoBERTa, and GPT variants, however, researchers started embracing such deep models that could handle deeper contextual understanding and higher performance.

The paper [10] explores the use of LLM such as BERT and RoBERTa for detecting cyberbullying across datasets, focusing on improving performance and addressing issues like class imbalance. The research utilized two datasets for training, one is from Formspring which contains cyberbullying and non-cyberbullying post. Another one is a curated dataset that combining the Formspring and Twitter dataset to address class imbalances. The study compares the performance of traditional machine learning models and LLM model. Since the research want to address the performance of imbalance dataset, the SMOTE was implemented to balance the datasets. Moreover, for Feature engineering, vectorization techniques help captured the contextual richness of the text. The model was then fine-tuned using both datasets separately and performance was measured using metrics like accuracy, F1 score, and precision recall balance. The result of the research is that RoBERTa emerged as the top performing model, achieving the highest F1-scores of 66% for case 1 and 87% across datasets. Application of the balanced second dataset was another major step in trying to solve the problem of class imbalance. On this dataset, the model RoBERTa increased its accuracy by 12%, hence generalizing better.

The paper [11] explored not only cyberbullying but also biases encapsulated in digital text. The paper presents methods for detecting biases at the crossroads of age, ethnicity, gender, and religion and the creation of artificial datasets which make the model stronger. It utilized both real datasets, collected from places such as Twitter, and synthetic datasets created with the most modern LLMs, such as ChatGPT, Pi AI, and Gemini. Each class contained 4,000 sentences generated based on designed triggers. Ethical development of data included filtering and moderation to keep the datasets relevant. Several variants were compared for performance between DeBERTa, RoBERTa, HateBERT, MobileBERT, ELECTRA, and XLNet. New applications will include the so-called "CyberBulliedBiasedBot" for bias and abusive content detection in real time. The models were also optimized by the quantization technique to let their use when computational resources are small. A significant aspect of the study was the use of multilabel classification, which allowed the models to detect both cyberbullying and biases within the same text. This approach

enabled a deeper understanding of the intersectionality of online abuse. The scores varied from the most performing models, HateBERT and RoBERTa, reaching over 90% F1-score on synthetic datasets and about 85% on real ones, thus underlining the gap produced by the noise and ambiguity of real-world data. These synthetic datasets marked a point of inflection in terms of model performance and provided diverse, contextually rich training examples. Mixing these with real-world datasets further increased the robustness on most metrics of evaluation. Another key contribution was the multilabel classification perspective that brought nuanced overlaps of biases together-things like age and ethnicity-framing online abuse in all its varied forms.

The paper [12] that explores how the Large Language Models, in particular, the GPT-3 model, can be used to detect and moderate cyberbullying in all types of social media. Although a significant part of the previous literature had turned its attention to the more traditional approaches of Support Vector Machine, Logistic Regression, and Neural Networks based on the concept of machine learning, the purpose of the study is to fill the existing gap between platform moderation and the state-of-the-art practices based on the use of LLMs. The study on the detection of cyberbullying with the help of Large Language Models (LLMs) was based on the fine-tuning of the GPT-3 Ada model with the Jigsaw Toxic Comment Classification dataset. Multi-label entries were reformatted into pairs of prompt-completions (singular labels) and duplicates and characters were eliminated by preprocessing the data. The model was optimized after four epochs with 0.2 percent of the training data as the batch sizes. The accuracy, precision, recall and F1-score were used to assess the performance with the results revealing an accuracy of 90 percent, precision of 92 percent and finally the F1-score of 93 percent. These findings prove the usefulness of the LLMs in detecting and controlling real-time cyberbullying and other activities on social networks.

2.2 Limitation of Previous Studies

The research on [10] identifies few limitations. The first one is it cannot detect more sophisticated forms of cyberbullying, such as sarcasm, coded language or implicit bullying. The use of binary classification approach only capable of determine whether a comment is “bullying” or “non-bullying”, but it does not account for more nuanced and context-dependent forms of bullying. while the study mentions Reddit and Facebook, it does not explicitly address the linguistic characteristics of Instagram, namely being heavily reliant on emojis, hashtags, and visual cues. Other than that, there is also some limitation related to the dataset and model constraints. The dataset lacks platform-specific slang and text styles like abbreviations, shortened words, and emojis thereby making it less effective at bullying detection on Instagram. Another critical limitation of this research is the lack of the classification of emotional intensity in bullying. The system succeeds only in detecting the presence or absence of bullying, not in classifying such bullying into levels of intensity: mild, moderate, severe. Finally, whereas the application of Large Language Models like GPT-3 holds promise in capturing context, this study is not about sentiment analysis as a way of quantifying the emotional impact of bullying.

The study [11] identifies few limitations, the first one is the inability to detect subtle forms of cyberbully particularly sarcasm, humour, and coded language. While the study employs models such as BERT and RoBERTa, which are context-aware, it relies on bag-of-words and TF-IDF-based methods to identify bullying. These models are effective at detecting explicit forms of bullying but fail to recognize the implied meanings behind sarcastic or coded language. This challenge underscores the need for context-aware LLMs that can detect the intent behind implicit and coded messages. Other than that, this study is also lack of classification of emotional intensity. Similar to research of [10], this study only focuses on binary classification, categorizing posts as bullying or non-bullying, without considering how intense or harmful the bullying is. This binary approach fails to capture the range of emotional impact that cyberbullying can have on victims. For instance, a mild insult might not have the same emotional impact as a severe threat or public humiliation. The paper also does not incorporate any form of sentiment analysis (SA), which could have been useful for measuring emotional intensity. Lastly, it also has limitations related to datasets and training data. One notable issue is class imbalance where only a small percentage of the dataset contains actual cyberbullying posts, while the rest are non-bullying. This imbalance

biases the model toward predicting "non-bullying," resulting in more false negatives (i.e., failing to detect bullying). Additionally, the datasets used for training (like Formspring and Twitter) are not well-suited for Instagram.

The study [12] identified few limitations, one of it is about the limitation in ability to detect subtle bullying. While the study discussed bias detection, it has not put into consideration how the hidden, sarcastic, or coded appearances of bullying could be differentiated. Additionally, most of the LLMs used in this study fail to process emoji-based bullying, which is prevalent on Instagram. Another limitation involves the categorization of emotional intensities. While the current study investigates multilabel classification in terms of bias and cyberbullying detection, the approach does not try to quantify emotional intensity. Lastly, although the study uses a mix of synthetic and genuine data, the synthetic ones can hardly mimic the depth and complexity of human language at Instagram and since the research is based on data from Twitter, which really doesn't relate to the environment on Instagram.

Table 2.0 Summary Table for Literature Review

Title	Year	References	Model/ Classifier	Dataset	Feature Engineering	Evaluation Metrics	Key Results
Cyberbullying Detection on Instagram using Naive Bayes	2019	[8]	Naive Bayes (NB)	Instagram comments dataset	TF-IDF, Stemming, K-Fold Cross Validation	Accuracy = 83.53% (with stemming)	Stemming improves performance by 0.53%
Cyberbullying Severity Detection on Twitter	N/A	[9]	Naive Bayes (NB)	Twitter dataset	PMI for feature extraction	Accuracy = 75% (multi-class)	SVM outperforms NB on multi-class tasks
Cyberbullying Detection on Twitter using SVM	N/A	[7]	Support Vector Machines (SVM)	Twitter dataset	N-gram features	Accuracy, Precision, Recall, F1-Score, ROC	SVM outperforms NB across all metrics

Sentiment Prediction on YouTube using SVM	N/A	[2]	SVM with AdaBoost & SGD	YouTube comments dataset	AdaBoost, SGD, Sentiment Embedding	Accuracy, Precision, Recall, F1-Score	Improved detection with optimization methods
Cyberbullying Severity Detection using Random Forest	N/A	[9]	Random Forest	Harassment dataset (GitHub repository)	SMOTE, Embedding, Sentiment, Lexicon, PMI-SO	Accuracy = 91.153%, F1-Score = 0.898	Outperforms SVM and NB models
Cyberbullying Detection using BERT and RoBERTa	N/A	[10]	BERT, RoBERTa	Formspring, Formspring + Twitter (combined)	SMOTE for imbalance, Vectorization	F1-Score = 66% (case 1), 87% (combined	RoBERTa outperforms other LLMs
Bias and Cyberbullying Detection using Multiple LLMs	N/A	[11]	RoBERTa, DeBERTa, HateBERT, MobileBERT, XLNet	Twitter (real) + Synthetic datasets (4,000 samples/class)	Quantization, Multilabel Classification	F1-Score = 90% (synthetic), 85% (real)	HateBERT, RoBERTa achieve best performance
Cyberbullying Detection Using GPT-3	N/A	[12]	GPT-3 (Ada)	Jigsaw Toxic Comment Classification dataset	Data Cleaning, Prompt-Completion Formatting	Accuracy = 90%, Precision = 92%, F1-Score = 93%	Demonstrates high accuracy for real-time use

2.3 System and Dashboard review

2.3.1 JPInfotech – Rule-Based Cyberbullying Detection System

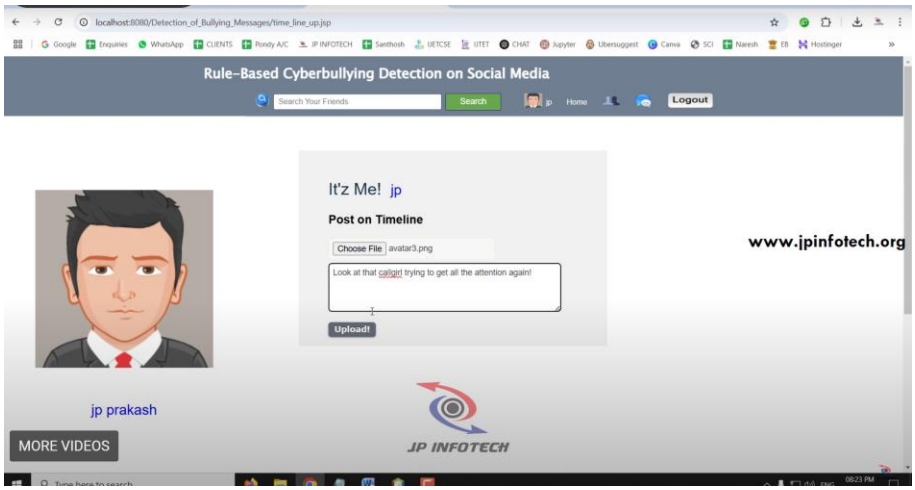


Figure 2.3.1 JPInfotech System Interface

A detection system based on rules developed by JPInfotech [13] evaluates offensive social media content through predefined rules alongside Java, JSP and MySQL technologies. The system enables real-time text search to locate harmful content by utilizing a set of adjustable evaluation rules for assessment purposes. The system depends heavily on pre-defined rules which create obstacles when identifying novel forms of cyberbullying because languages and slangs transform rapidly. The detection system fails to identify subtle bullying methods which results in reporting nonexistent bullying situations among actual harmful instances.

2.3.2 Sundaresan0502 – Machine Learning-Based Detection System

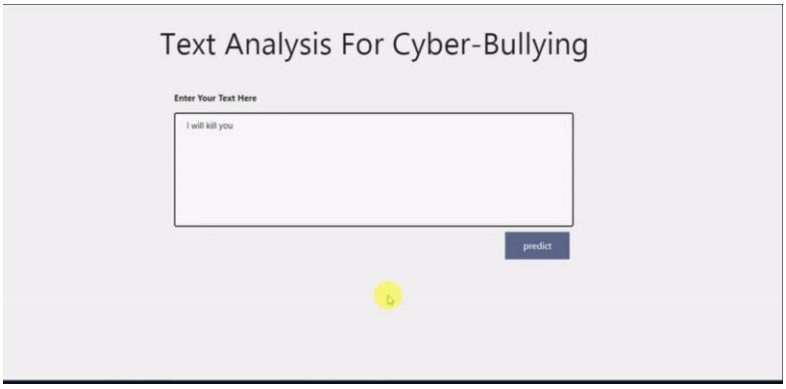


Figure 2.3.2.1 Sundaresan0502 landing page

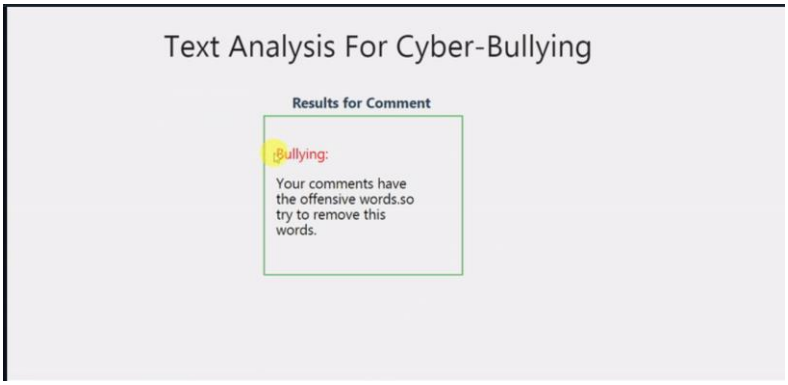


Figure 2.3.2.2 Sundaresan0502 prediction result page

Sundaresan0502 [14] developed the Cyberbullying Detection System which applies Multinomial Naive Bayes classifier for identifying dangerous or safe content in social media. The system extracts elements from text data then uses this data to determine when cyberbullying happens and to refine its knowledge base for continuous learning. The model operates based on exceptional training data and various forms of cyberbullying content representation. Naive Bayes classifier struggles to detect complex text interactions thus resulting in incorrect classifications particularly during cases of vague or delicate bullying occurrences.

2.3.3 Lightspeed Systems – AI-Powered Cyberbullying Prevention Platform

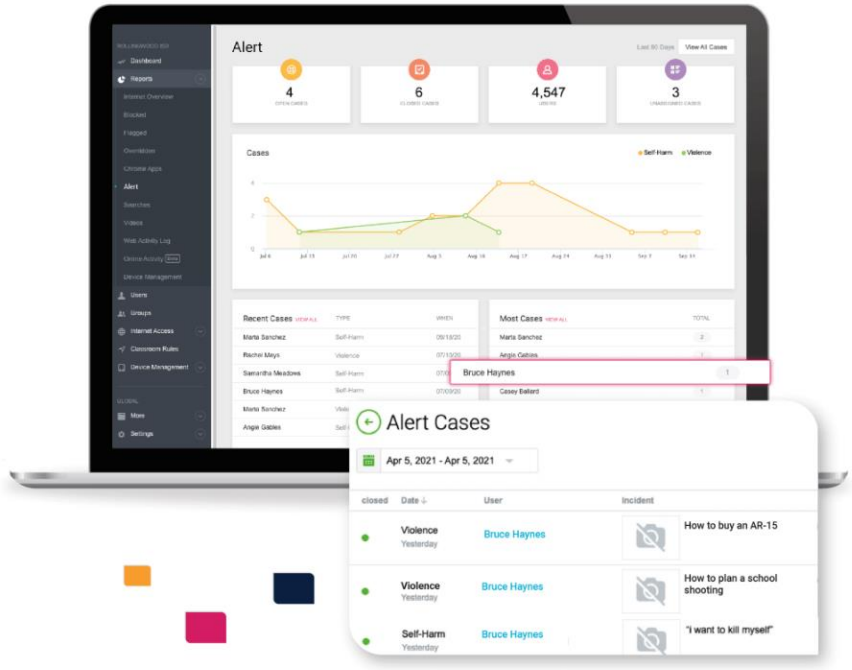


Figure 2.3.3 Lightspeed Systems Dashboard interface

Lightspeed Systems [15] delivers a full suite of tools enabled by AI to scan digital spaces and stop cyberbullying behaviour mainly targeting educational institutions. The system scans current digital interactions to detect dangerous material before notifying school authorities immediately. The defined search protocols of the platform prevent it from effectively adapting to evolving bullying strategies among online users and new digital harmful behaviours. Student privacy faces ethical issues because of thorough monitoring platforms.

2.3.4 Ahmad Ndmz – Cyberbullying-Detection-System

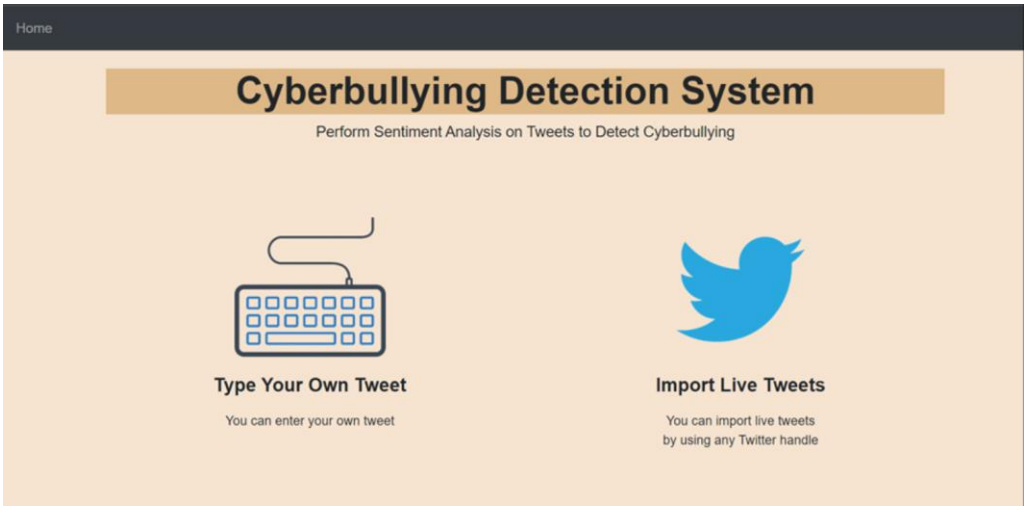


Figure 2.3.4 Ahmad Ndmz Dashboard Interface

A system utilizing Linear Support Vector Machines (SVM) together with Natural Language Processing (NLP) techniques detects cyberbullying incidents on social media according to Ahmad Ndmz [16]. Computer programs using NLP obtain semantic data and contextual meanings in text to detect various bullying forms including both direct and indirect subtle expressions. The system encounters reduced accuracy when it handles training data of poor quality and experiences difficulties with multilingual content as well as internet slang.

2.3.5 Shabroz Kamboj – Cyberbullying-Detection-System

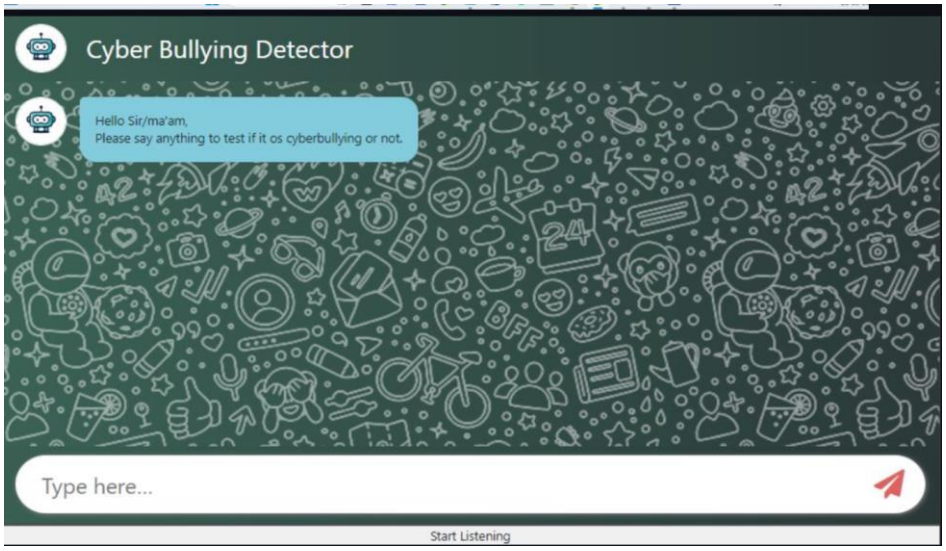


Figure 2.3.5 Shabroz Kamboj Website Interface

Shabroz Kamboj developed the Cyberbullying Detection Bot which uses Machine Learning NumPy Stack tools including NumPy and Pandas and TensorFlow and Keras and Sci-Kit Learn and NLTK to detect bullying in social media comments [17]. The bot operates through a frontend built with React which emerges from Material UI and Bootstrap design elements integrating Google Translate as its multilingual support mechanism. Through its deep learning process, the system can detect intricate bullying patterns because of its high computational demands. The primary concern about deep learning models today exists in preserving model transparency because their classification reasons often remain unintelligible to researchers.

Table 2.3 Summary Table for Reviewed System and Dashboard

System	Machine Learning Approach	Strengths	Weaknesses
JPInfotech	Rule-Based (Java, JSP, MySQL)	Customizable rules, real-time monitoring	The system demonstrates awkwardness when facing contemporary linguistic changes while encountering problems with covert bullying practices.
Sundaresan0502	Multinomial Naive Bayes (NB)	Simple, interpretable, efficient with small datasets	The system faces difficulties when trying to handle sophisticated or discreet bullying situations because it relies on simple relationship connections.
Lightspeed Systems	AI-Powered Detection (Proprietary)	Real-time alerts, human review, comprehensive monitoring across platforms	In certain situations that involving privacy risks exist along with the possibility of missing subtle meaning during interactions.

Ahmad Ndmz	Linear Support Vector Machine (SVM)	High-dimensional data poses no problem for this approach and it demonstrates great resistance to overfitting.	Complex patterns represent a challenge because linear decision boundaries are its only operational boundary
Shabroz Kamboj	Deep Learning (LSTM, TensorFlow, Keras)	This method captures sophisticated data patterns and remains suitable for different kinds of information.	The system needs substantial computational strength while keeping its internal workings non-transparent to users.

2.4 Structure of BERT and RoBERTa Architectures

BERT is a transformer based language model that uses simultaneous processing of all tokens in a sequence to obtain bidirectional contextual embeddings of a text, eliminating the need to use unilateral left-to-right or right-to-left readings [18]. Its input schema includes token embeddings, segment embeddings and positional embeddings and then become channels of several layers of transformer encoders. The encoder layers have a feed forward neural network and a multi head self attention mechanism on each, thus allowing the model to integrate dependencies across the length of a sentence [18]. BERT is trained using two major goals, Masked Language Modeling (MLM) in which a specific part of the tokens is masked and then the prediction is made and Next Sentence Prediction (NSP) in which the model either predicts whether the next sentence follows the current sentence in the natural text or not [18]. BERT has been successfully fine-tuned in the area of cyberbullying detection and, as such, is able to recognize threatening expressions and context-specific features in social media posts to attain strong performance on detecting material of bullying nature [19].

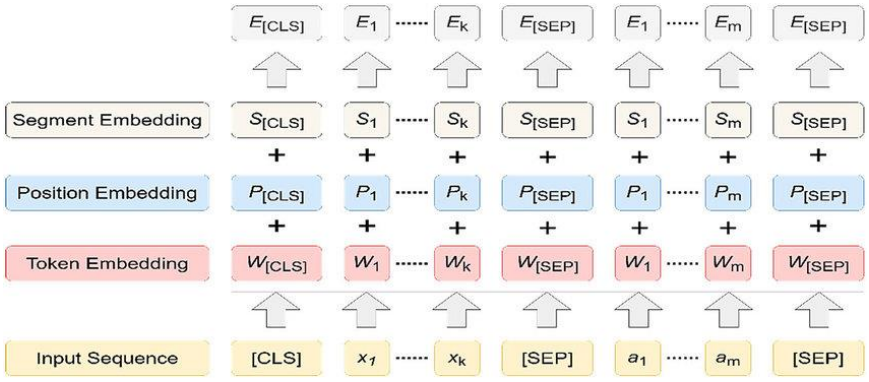


Figure 2.4.2 BERT Input Representation [22]

RoBERTa is builds on BERT by improving its pre-training paradigm [20]. In contrast to BERT, RoBERTa completely does away with the NSP task and focuses solely on the MLM. It is trained with significantly larger mini-batches and higher learning rates and with longer training sequences, which prepares the model to learn more complex textual dependencies [20]. These improvements make RoBERTa more robust and more easily accessible on the variety of natural language processing tasks [20]. Recent studies also highlight that they are useful in participant-level classification, in which models do not just need to identify the presence of bullying but also the role of various users in a conversation (e.g., bully, victim, or bystander). Ratnayaka et al [21] examined this by applying role identification to detecting cyberbullying on ASKfm data such that the classification performance is highly optimized, with F1-scores of role classification of around 0.76, showing that role-aware cyberbullying detection based on transformer-based models is a strong base of classification. The most thorough awareness of the architectural differences between BERT and RoBERTa will prepare the researchers to make the most relevant decisions regarding the optimal model to use in the efforts of detecting cyber-bullying, where subtle meaning of words and context is the most important factor.

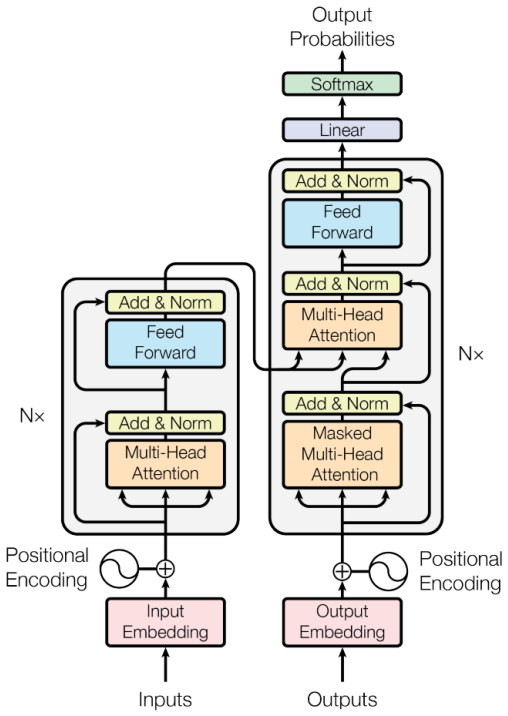


Figure 2.4.1: Transformer Architecture Diagram [23]

2.5 Explainable AI (XAI) and Attention Mechanisms in Cyberbullying Detection

The fast growth of language models based on transformers to detect cyberbullying has significantly improved predictive accuracy and strength, but it has also raised the issue of their opaque and black-box character. Considering that cyberbullying often deals with very sensitive situations, including that of victimization, harassment and attributing roles, there is an eminent need that such systems must demonstrate transparency in their decision-making processes. The necessity has prompted the inclusion of Explainable Artificial Intelligence (XAI) methods in the context of natural language processing (NLP).

XAI methods attempt to make model decisions more interpretable by finding those portions of input text that have the most significant impact on classification outputs. Such interpretability may be especially useful in the sphere of cyberbullying detection, where practitioners and interested parties, such as educators and social media sites, should be capable of determining why a specific post was detected. This will enable trust, accountability, and possible legal justifiability of automated systems [24].

Attention mechanism is one of the most used interpretability tools in transformer architectures. Attention weights and are computed as part of the layers of BERT or RoBERTa to offer a numeric representation of the relative significance of every token in the sequence compared to others. An example of this is that in a message like the one below, “Go kill yourself, loser”, the focus of the attention layers is often on the words, “kill”, “yourself”, and “loser”, which sheds light on the toxic elements that form the basis of a bullying category [23].

Recent progresses in explainable artificial intelligence (XAI) focus on the application of attention mechanisms, attribution procedures like SHAP and LIME, co-attention networks to highlight the foreground of particular lexical items, phrasal architecture, or interaction patterns in a conversational scenario which have the most significant impact on the decision of a model. As an example, HENIN by Chen and Li [25] is a heterogeneous neural interaction network that utilizes co-attention to learn to model user and content interaction. Their empirical findings indicate that the outcome attained attention weights can provide a form of insight into the conversational dynamics that promotes instances of cyberbullying, which thereby makes the process of detecting the same more interpretable.

On the same note, Mehta and Passi [26] used XAI algorithms in hate-speech detection by demonstrating how attention and feature-attribution algorithms can reveal an initial view of both obvious and hidden context signals. These results can be applied to the sphere of cyberbullying detection where indirect or sarcastic words often obscure ill motive.

Prama et al. [27] explored the topic of personalized explainability in the context of detecting the severity of cyberbullying, using SHAP and LIME to provide case-by-case explanations. In their article, the authors point out that the personalization process can help to understand how the same content can have different effects on different users, which is especially relevant when the extent of emotional damage is at stake.

Recent works also point to the classification at the participant level, in which models are capable of not only identifying the presence of bullying, but also the roles of the users in the interaction (i.e., bully, victim, bystander). This issue was explored by Sandoval et al. [28] in the frame of Identifying Cyberbullying Roles in social media with the use of AMiCA dataset. They optimized a set of transformer models (BERT, RoBERTa, T5, GPT 2), tackled the problem of class imbalance by oversampling, and discovered that a RoBERTa model had an overall F1-score of about 83.5 - score of about 89.3 score when a prediction threshold was used. These findings indicate that role-conscious detection is effective and possible.

To sum up, XAI and attention mechanisms can reduce the gap between the strong performance of deep learning models and the need to have interpretability in sensitive uses of deep learning like cyberbullying detection. Explaining the linguistic features that result in model predictions, XAI provides developers and practitioners with the tools to verify model behavior, reduce bias, and strengthen confidence in real-world applications.

CHAPTER 3
Methodology

This chapter outlines the method used to develop a sentiment-sensitive cyberbullying detection system, and the focus is on the pretrained Large Language Models to optimise them and process textual data retrieved on Instagram. The process is broken into specific steps: the preliminary project development, the preprocessing of the data, the model-training architecture design and the selection of the training data, the assessment of the predictive performance on the basis of a specific test data.

3.1 System Design Diagram

3.1.1 Use Case Design

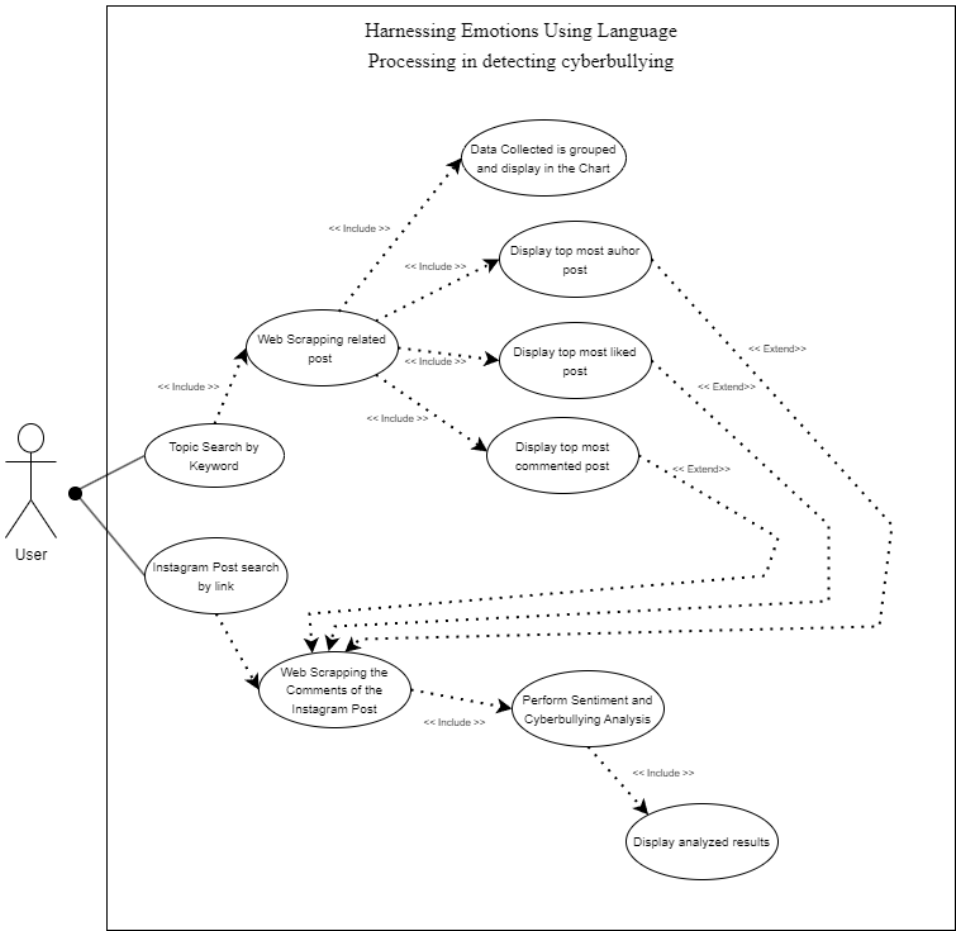


Figure 3.1.1 Dashboard Use Case Diagram

Figure 3.1.1 shows the use case diagram for the User Dashboard. The users are given two options to choose from. The first options is users can start Topic Search by Keyword through an interface that accepts input of pertinent keywords or phrases. The system commences web scraping processes for Instagram related posts whenever this directive action is executed. The system collects related posts for further use which enables the generation of three additional insights through the showcase of top most liked post, top most commented post, and top most author post display options. The system displays group data through charts which helps users better understand the information presented.

The Second options allow users access the Instagram Post Search by Link. The user just pastes in the Instagram post links. Each chosen post link will be investigated for its comments by the system processing. The analysis operation initiates here because system execution commences sentiment and cyberbullying assessment of obtained comments by employing an enhanced linguistic framework.

Users see the analysed results through a completed page which displays detected cyberbullying content with severity levels and emotional tones of the comments.

3.1.2 Activity Diagram

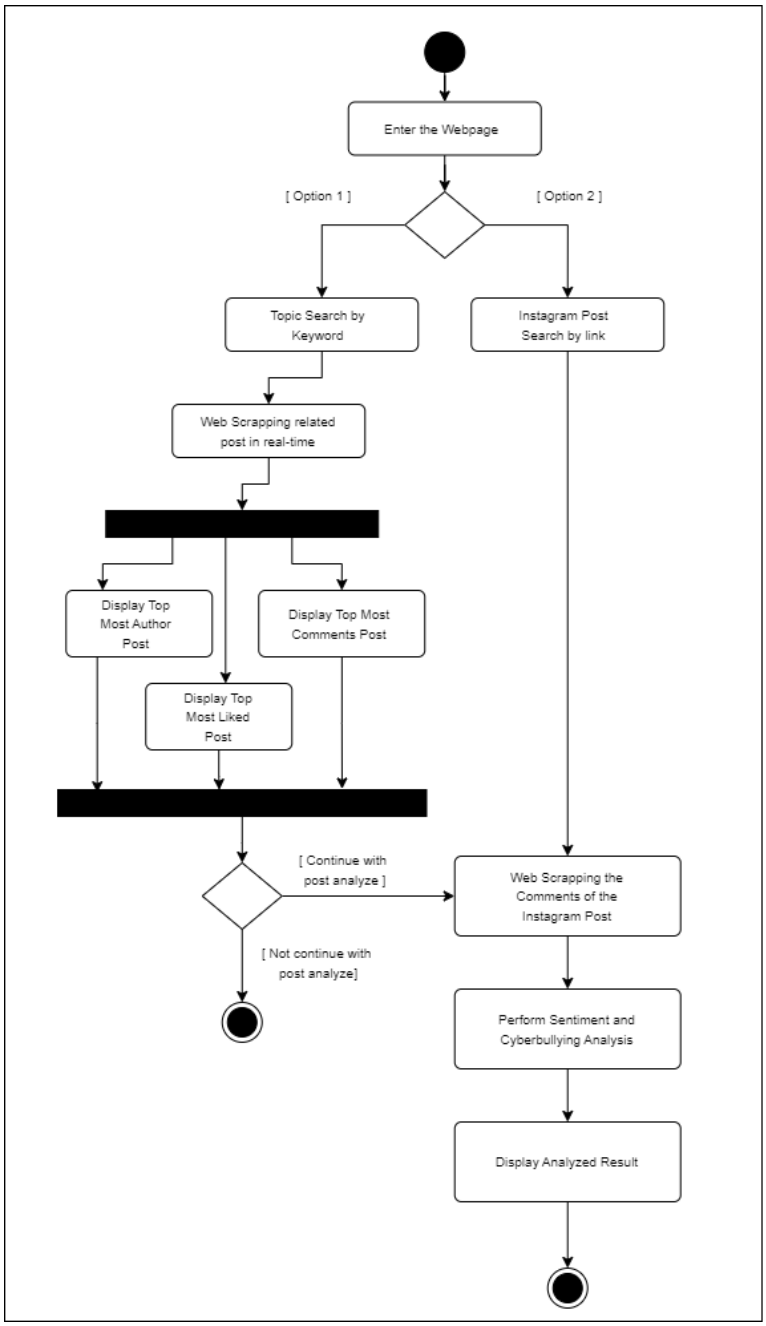


Figure 3.1.2 Dashboard Activity Diagram

Figure 3.1.2 shows about the Dashboard Activity Diagram of the system. Users encounter two entry points on the online webpage which enable them to start data retrieval and analysis processes.

The user has the option to perform Topic Search by Keyword. The system proceeds with live Instagram web scraping to gather posts which include the selected keyword. The system presents users with key post findings that are ranked according to specified criteria once it gathers appropriate content. The system presents three top post insights to users which include Top Most Author Post and Top Most Commented Post and Top Most Liked Post. Users can make a choice to perform post-analysis following their review of the summarized results. Proceeding with the activity depends on their decision. Moving forward with the analysis requires user consent after which the system advances to deeper assessment.

Beside than that, the user can directly enter an Instagram Post link as their second options. The system collects the comments from the particular post entered by the user. The user will reach this option by two paths: users who entered Option 1 can choose to analyze deeper or fresh users can initiate Option 2.

The system advances to execute sentiment analysis and cyberbullying detection through natural language processing algorithms after the collection of comments from any chosen route. The evaluated insights along with emotional metrics and bullying detection results appear to the user to signal flow completion.

3.2 Modified CRISP-DM Methodology

The traditional CRISP-DM (Cross-Industry Standard Process for Data Mining) was modified to accommodate the needs of utilizing pre-trained LLMs. The modified phase is as below:

3.2.1 Business Understanding

The first task during this phase involves defining the project's main purpose which consists of developing a detection system for cyberbullying that exceeds basic keyword analysis with emotional sensitivity and context functionality. The research conducts an analysis of online behaviour using Instagram comments selection as its main case study to discover harmful social interactions.

3.2.2 Data Understanding

Instagram data is collected during this phase through scraping techniques and APIs which focus on gathering user comments. The purpose here is to acquire authentic social media data which forms a wide-ranging collection of real-world examples. A first evaluation of the data helps researchers understand the characteristics of the content in addition to possible cyberbullying warning signs.

3.2.3 Data Preparation

A data preprocessing step takes place which cleans the raw material in order to make it usable for LLM fine-tuning. The initial preparation process eradicates duplicated data while filtering texts that are not in English and removes special symbols and emoji content along with standardizing text formatting. The prepared data contains labels under "cyberbullying" and "neutral" and "non-cyberbullying" categories with defined severity levels included.

3.2.4 Modeling

The prepared and labelled dataset serves to finetune a pretrained RoBERTa lls throughout the modeling phase. The model receives refined parameter adjustments during this stage to excel at recognizing cyberbullying occurrences specifically through understanding emotional content together with linguistic contextual details.

3.2.5 Evaluation

Testing performance of the fine-tuned model takes place on a distinct dataset used for evaluation. The model evaluation depends on accuracy, weighted F1 score and AUC to evaluate its capability to detect cyberbullying and measure its impact severity. The evaluation process establishes both operational effectiveness and dependability of the model in practical settings.

3.2.6 Deployment

The last development phase integrates the optimized model into a Flask web interface allowing users to submit comments for immediate assessment. Users receive predictions through the system which indicate if the input includes cyberbullying and show the identified severity level. The final deployment phase enables quick access and functional use of monitoring and intervention methods in the future.

3.3 Project Timeline

Activity	Weeks													
FYP 2	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Initiation														
Meeting with Supervisor														
Research and Literature Review														
Prelimary Work														
Scrapping tools upgrade and update														
Data Collection and Annotation														
System Design Enhancement														
Model Development and Fine-tuning														
Model Evaluation														
Dashboard Integration														
Testing and Debugging														
Documentation of Progress														
Finalize Report														
Closure														
Submittion of FYP 2														
Presentation Slides Submition														
Oral Presentation														

Figure 3.3 Project Timeline

CHAPTER 4

System Design

4.1 System Architecture Design

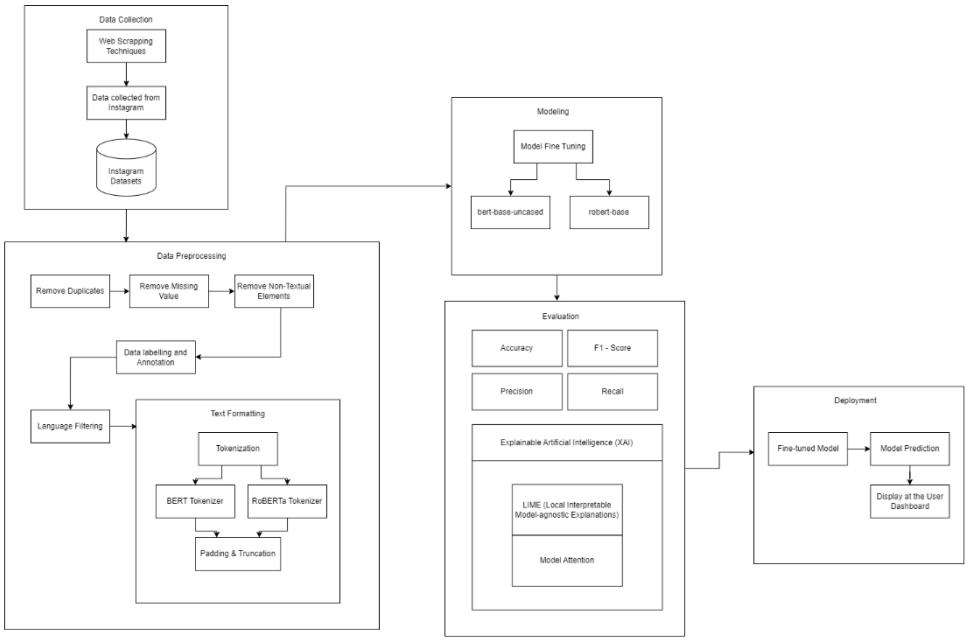


Figure 4.1 System Architecture Design Process for Model Fine-Tuning

Figure 3.5 provides a detailed overview of the system architecture design for the sentiment analysis and cyberbullying detection system. The diagram visually shows the flow of creating the model from Data Collection to deployment. Each Component and the process is described as below:

4.1.1 Data Collection

The system collects Instagram data using automation for scraping operations. The materials collected from Instagram consist of posts together with their metadata and comments that shape the core elements of the database. Model developers categorize the collection data as hate speech or non-hate speech for training purposes.

4.1.2 Data Preprocessing

The raw information from Instagram goes through cleaning steps for model training purposes.

4.1.2.1 Data Cleaning

- **Remove Duplicate:**
  A systematic process should identify duplicate data entries for their subsequent removal to establish unique training records.

- **Handle Missing Values:**
  The handling of missing data values can proceed through two strategies which are either the removal of incomplete data entries or the imputation of missing information.

- **Remove Non-Textual Elements:**
  Discard every element which does not include textual information from the analysis. Example: emoji, Hashtags and URLs.

4.1.2.2 Text Normalization:

- **Language Filtering:**
  Language Filtering eliminates non-target or irrelevant languages which appear in the data collection.

- **Data Labelling and Annotation**
  The process of supervised learning requires data labelling and annotation which helps categorize text content into cyberbullying or non-cyberbullying groups.

4.1.2.3 Text Formatting

- **Tokenization:**
  The text gets divided into smaller fragments such as words or phrases which serve as the basis for subsequent analysis.

- **BERT Tokenizer and RoBERTa Tokenizer:**
  The deep learning model requires preparation through the tokenizing process before it accepts textual input.

- **Padding and Truncation:**
  The model needs all input texts to have identical lengths through padding followed by truncation methods.

4.1.3 Modeling

- **Model Fine-Tuning:**
  The research used a method of fine-tuning a BERTand RoBERTa as a pretrained language model to enhance its capacities for recognizing text emotions and cyberbullying. BERT and RoBERTa functions as a robust and transformer-based model which receives data specification for better linguistic processing.

4.1.4 Evaluation

Performance Metric

- **Accuracy:**
  The model achieves accuracy through its capability to make right predictions during its operational phase

- **F1-Score (Weighted and Macro)**
  This metric shows the trade-off of the precision and recall of a balanced and imbalanced distribution of classes and thus gives measure of performance of a model in a heterogeneous environment.

- **AUC (Macro)**
  This is a statistic that measures the discriminative capacity of the classifier by counting the average area under the receiver operating characteristic curve of each class, which is then a cumulative measure of separation efficacy that does not vary with the class prevalence.

Explainable AI (XAI)

- **LIME**
  LIME enables users to obtain explanations about single predictions from a model to enhance human understanding of its decision-making process.

- **Attention Visualization**
  Highlights important tokens and phrases in the input text that influenced the model's decision.

4.1.5 Deployment

1. **Fine-Tuned Model:**

   The real-time prediction process begins using the fine-tuned model that has been deployed.

2. **Model Prediction:**

   Through emotional analysis the predicted model determines whether the text shows cyberbullying behaviour and additional sentimental analysis.

3. **User Interface:**

   Easy accessibility of model predictions exists through a user dashboard on the interface which enables end-users to understand the output.

4.2 Modeling Phase Block Diagram

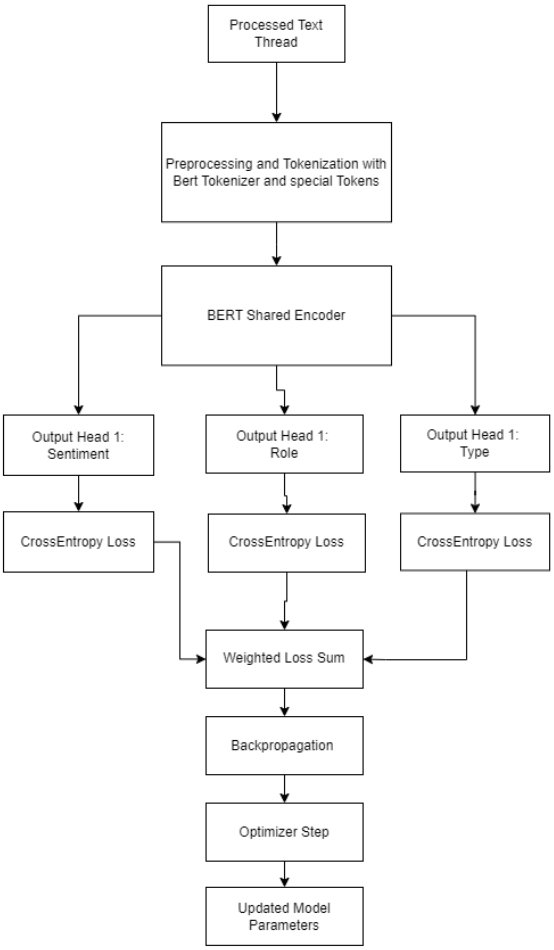


Figure 4.2: Bert Modeling Block Diagram

Figure 4.2 describe the modeling phase of the development, firstly the preprocessed comment text is formatted with special tokens in such format [CAPTION][PARENT][REPLY] and [CAPTION][PARENT] to ensure that the model learn the whole context. Next, each message is tokenized using the BERT tokenizer. Each special token is declared in the tokenizer to distinguish roles within a conversation. Tokenized sequence is then converted into input IDs and attention masks. Thirdly, after the text is tokenized with special tokens, the text is fed into the BERT Encoder with 3 head declared to predict Sentiment, Role and Type. The 3 head is sharing the same encoder to ensure each underlying language representations are consistent and context-aware. Fourthly, after each head produces logits for its respective prediction which are passed through a

SoftMax layer to generate probabilities. CrossEntropy Loss are computed and combined into a weighted sum to form the total loss, which is then backpropagated to update the encoder and task specific head. Lastly, there are two stages in training the BERT model, first stage is freezing the BERT model for stability of the model, this allow BERT to not forget the pretrain knowledge when my context specific data is given for training. The second stage then unfroze the BERT to fine-tuned for the model with the rest of the data. Additional optimization like optimized learning rate scheduling using AdamW and Early stopping when detecting model starts to overfitted. This setup allows the model to jointly learn sentiment, roles and cyberbullying type.

4.3 Deployment Phase Block Diagram

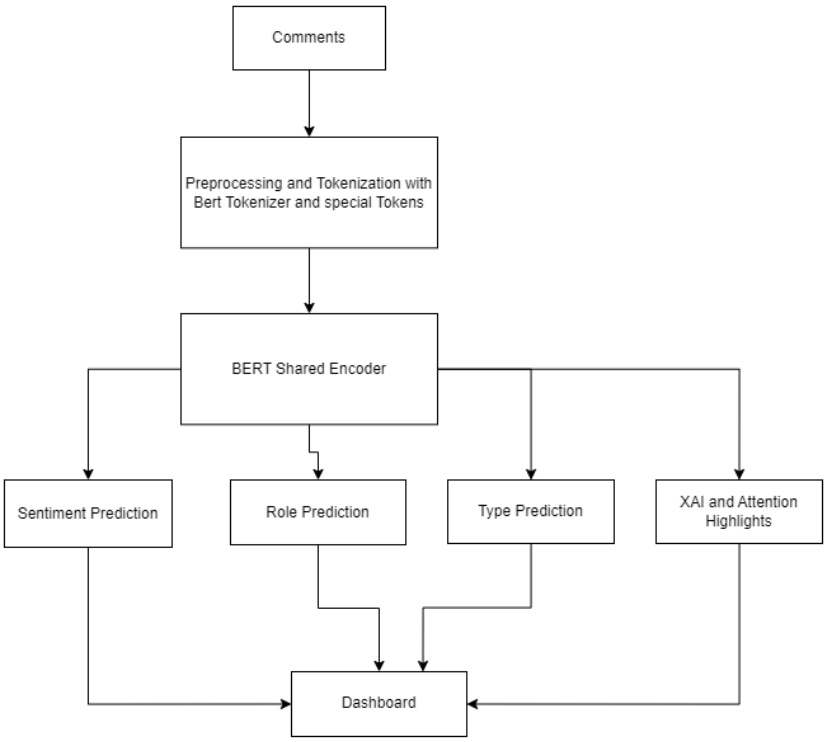


Figure 4.3: Deployment Phase Block Diagram

The Figure 4.3 shows the way the trained cyberbullying detection model is applied in practice. After saving the best model checkpoint (best_model.pt), a new conversation thread with a caption, parent comment or reply is inputted. The raw text is tokenized and preprocessed with the same BERT tokenizer that is used during training, special tokens [CAPTION], [PARENT], and [REPLY] are added in order to maintain the conversational context. The formatted text is then converted into attention masks and input IDs and then forward to the pretrained model. The shared encoder along with three classification heads results in sentiment, conversational role and cyberbullying type predictions. A softmax layer converts each raw logits into probabilities and the greatest probability class are chosen as the final prediction. An explainability module is used to extract attention weights to illustrate the words or phrases that informed the prediction in order to make the process more transparent. Lastly, the findings are presented on the Flask-based dashboard, where the user can observe that cyberbullying was identified, the role and sentiment expected, and the risky words mentioned.

4.4 Dashboard Wireframe

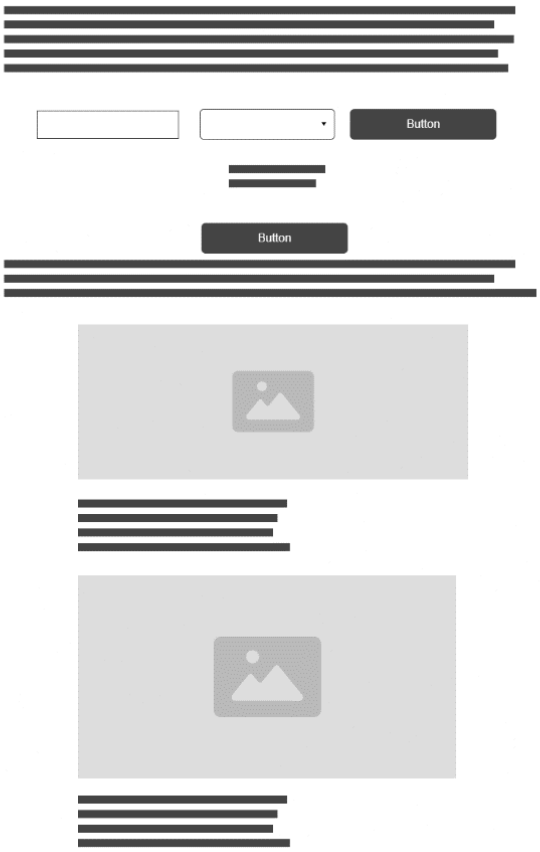


Figure 4.4.1 Main Page Wireframe

The web-based dashboard application has a wireframe diagram as shown in Figure 4.4.1. This page mark as the landing page. It allows user to get started to the system, the user can search for recently hot topic and gather information about it on Instagram. The page will scrape the web and organize and visualize it into chart. User can then go to next page to analyse the Instagram comment using Instagram post URL or copy from the result returned from the main page.

Figure 4.4.2 Analyse Page Wireframe

The wireframe of the Analyse page is shown in Figure 4.4.2. The page firstly will only have one input text box for the user to input Instagram post URL link. After that, the system will scrape all the comments and metadata of the post and display on a box below. The user also allows to modify the title to help the model for a more accuracy prediction. This is because sometimes the Instagram post does not have a clear title or it is a picture-based Instagram post.

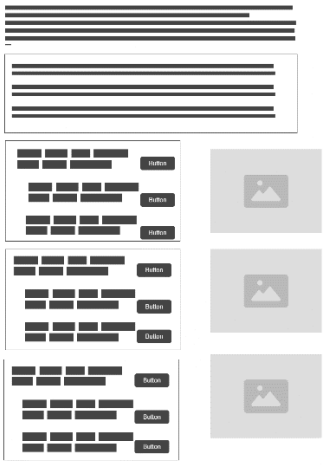


Figure 4.4.3 Analyse Result Page Wireframe

Figure 4.4.3 shows the wireframe of the Analyse Result Page. This page will take all the comments collected in Analyse Page and go through the model to make prediction. The model will return the prediction and the confidence of the prediction for each label. Each comment is display in a threaded to show the conversation level. Lastly, a graph is visualized on the right side to provide insights for each label.

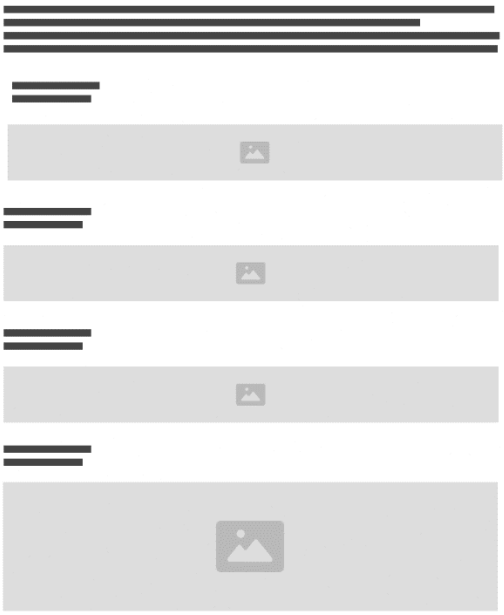


Figure 4.4.4 Explainable AI and Attention Page Wireframe

Figure 4.4.4 shows the wireframe of the Explainable AI and Attention Page. On this page, the user can see the visual representation of the way the model comes up with its predictions by indicating the words or phrases that had the greatest contribution to the decision. The identified words as highlighted based on the attention weights of the model or explainability module give the transparency and assist the user in knowing why a particular message has been selected as cyberbullying. Moreover, the page shows the estimated sentiment, role and the type of cyberbullying, providing a clear and interpretable picture of the system output.

Chapter 5

System Implementation

5.1 Hardware Setup

The developed system went through development work as well as testing using a desktop computer. This setup provided the computational power needed to support activities such as data preprocessing, model optimization, and performance evaluation. The desktop environment allowed the processing of large datasets efficiently along with the implementation of deep learning processes efficiently.

Table 5.1 Specifications of the Computer

Description	Specification
Motherboard	A520M-A Pro
CPU	Ryzen 5 5500
OS	Windows 11
Graphic Card	Radeon RX6600
Memory	16GB
Disk/Drive	500GB SATA HDD

5.2 Software Setup

The development of the project is divided into few parts, the process includes, the building of the dashboard, real-time web scraping, data preprocessing and model training. The following are the software tools and frameworks that were used to facilitate these processes:

5.2.1 Programming Language and Libraries

1) **Python** – The primary programming language used for implementing data preprocessing, annotate, model fine tuning, evaluation and deployment tasks.

2) **Transformer Library** It is being used for fine-tuning pre-trained Large Language Models (LLMs) such as Bert-base and RoBERTa for conducting effective sentiment analysis as well as emotion analysis

3) **Pandas and NumPy** - A library to manipulate data and numerical computations during data preprocessing.

4) **Scikit-learn** - for machine learning utility functions, i.e., data split, etc., evaluation metrics

5) **PyTorch** – A deep learning framework for training and fine-tuning of the model.

5.2.2 Web Scraping tools

1) **Selenium** - It is used for automating the interactions of the browser for the purpose to extract data from dynamic websites.

2) **BeautifulSoup** - Parsing the HTML content and extract the textual content from the webpage, in this case to extract the Instagram post.

3) **Stevesie.com** – Scraping the Instagram data from the Official Site without violating its Terms of Service using HAR Files. Used in FPY 1 and drop this method and found a better method.

4) **Instaloader** – A public repository from GitHub that allows for scrapping Instagram by simulating API calls from mobile devices.

5) **GraphQL Simulation** – Integrated along side with instaloader to emulate Instagram internal GraphQL request from a web.

5.2.3 Model Training and Evaluation

1) **Hugging Face Datasets and Trainer -** The implementation simplifies data management, batching processes and transformer architecture training.
2) **Explainable AI (XAI) Libraries -** Used for generating interpretability outputs such as attention heatmaps and highlighted influential words/phrases.
3)

5.2.4 Dashboard Development

1) **Flask** – used to building the dashboard and allow dynamic interaction with the interface. Users can get insight dynamically during that time. It is lightweight and can seamlessly host a machine model on the backend.
2) **Chart.js / JavaScript** – Integrated into the dashboard for visualizing model outputs, such as sentiment distribution, role classifications, and cyberbullying type frequencies.

5.2.5 Integrated Development Environment (IDE)

1) **Visual Studio Code -** The primary IDE employed throughout development of the source code of the project, with features of syntax highlighting, debugging, and integration of a version control system.

5.3 Setting and Configuration

This section outlines the setup and configuration of the development environment, library and framework required to develop the proposed cyberbullying detection system.

5.3.1 Development Environment

This project is mainly developed in python programming language. Thus, we need to configure a virtual runtime environment in order to run the programme. The runtime environment will be configured in Visual Studio Code.

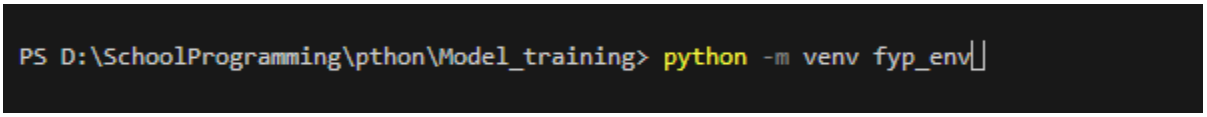


Figure 5.3.1.1 Creating the Python Virtual Runtime Environment

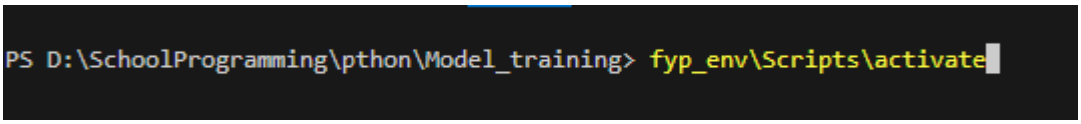


Figure 5.3.1.2 Activating the Python Virtual Runtime Environment

Based on Figure 5.3.1.1 and Figure 5.3.1.2, we first create the python virtual environment to isolate dependencies. After that, we need to access to it using the command otherwise the code will be run through global dependency instead of isolated dependency.

5.3.2 Required Libraries

```
pip install torch
pip install transformers
pip install scikit-learn
pip install pandas
pip install numpy
pip install matplotlib
pip install flask
pip install lime
pip install selenium
pip install requests
pip install beautifulsoup4
pip install notebook
pip install label-studio
```

Figure 5.3.2.1 Required library installation

The following library based on figure 5.3.2.1 is installed via python **pip** package manager, this is important to support the model training, evaluation, visualization and deployment of the system. While the training and fine-tuning of the model code is written on python file, the data processing is done by using jupyter notebook and the annotation of the cleaned dataset is label using label-studio.

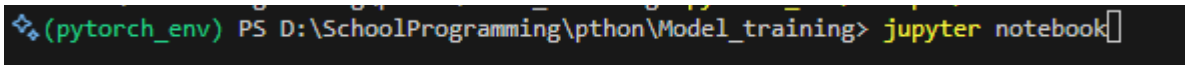


Figure 5.3.2.2 Command line to run local jupyter notebook server

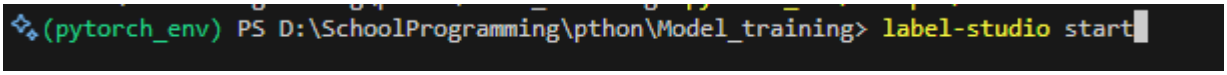


Figure 5.3.2.3 Command line to run local label studio

5.3.3 Data Cleaning and preprocessing

```python
import re
import emoji

# Text Processing Function
def clean_text(text):
    text = re.sub(r'@[^\s]+\s*', '', text)          # mentions
    text = re.sub(r'http\S+', '[URL]', text)        # urls
    text = re.sub(r'#\w+', '', text)                # hashtags
    text = emoji.replace_emoji(text, replace="")    # remove emojis
    text = re.sub(r'([!?.]){2,}', r'\1', text)      # repeat punctuation
    text = re.sub(r"[°•·~*…]+", " ", text)          #Remove decorative symbols
    text = re.sub(r"\s+", " ", text)                #Remove excessive newlines, tabs, and extra spaces
    return text
```

Figure 5.3.3.1: Code Snippet of Data Cleaning

Extract and Transform Clean Dataset to Label-Studio ready Dataset for annotation

```python
folder_path = "Cleaned"
all_flattened = []

for file_name in os.listdir(folder_path):
    if file_name.endswith(".json"):
        file_path = os.path.join(folder_path, file_name)
        with open(file_path, "r", encoding="utf-8") as f:
            post = json.load(f)

        flattened_data = transform_dataset(post)
        all_flattened.extend(flattened_data)

#Save everything into one file
with open("flattened_comments_full.json", "w", encoding="utf-8") as f:
    json.dump(all_flattened, f, indent=2, ensure_ascii=False)

print(f"Saved {len(all_flattened)} comments into flattened_comments.json")
```

```
Saved 13388 comments into flattened_comments.json
```

Figure 5.3.3.2: Flattened data for annotation

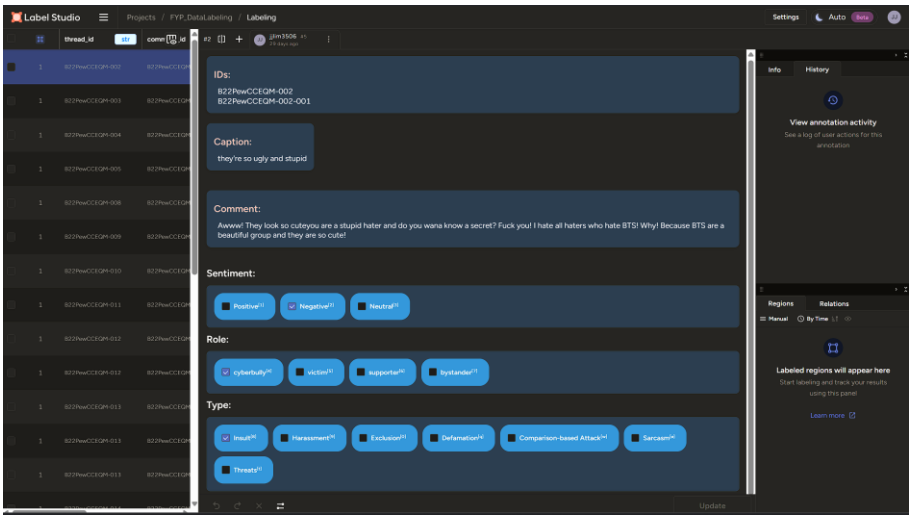


Figure 5.3.3.3: Label-Studio labeling process

Convert Back from Label-Studio Json format

```python
import json

def transform_data(input_data):
    # Group data by post_id (shortcode)
    posts_data = {}
    for item in input_data:
        post_id = item["data"]["post_id"]
        if post_id not in posts_data:
            posts_data[post_id] = []
        posts_data[post_id].append(item)

    # Process each post separately
    output = []

    for post_id, post_items in posts_data.items():
        # Get the first item to extract post-level data
        first_item = post_items[0]["data"]

        # Create the base post structure
        post_obj = {
            "shortcode": post_id,
            "post_url": first_item["post_url"],
            "caption": first_item["post_caption"],
            "comments": []
        }

        # Add optional fields only if they exist
        optional_post_fields = ["likes", "comments_count", "date"]
        for field in optional_post_fields:
            if field in first_item and first_item[field] not in ["", None]:
                post_obj[field] = first_item[field]

        # Organize comments by thread within this post
        comments_by_thread = {}

        for item in post_items:
            data = item["data"]
            thread_id = data["thread_id"]

            if thread_id not in comments_by_thread:
                comments_by_thread[thread_id] = []

            # Build comment object
            comment_obj = {
                "id": data["comment_id"],
                "text": data["text"],
                "owner": {
                    "username": data["user_id"]
                },
                "sentiment": data["sentiment"],
                "role": data["role"],
                "type": data["Type"],
                "replies": []
            }

            # Add parent_id if it exists (for reply handling)
            if data.get("parent_id") and data["parent_id"] not in ["", None]:
                comment_obj["parent_id"] = data["parent_id"]

            # Add optional fields
            if "created_at" in data and data["created_at"] not in ["", None]:
                comment_obj["created_at"] = data["created_at"]

            comments_by_thread[thread_id].append(comment_obj)

        # Process each thread to build comment hierarchy
        all_comments = []
        for thread_comments in comments_by_thread.values():
            all_comments.extend(thread_comments)
```

Figure 5.3.3.4: Convert label-studio format back to normal

Flatten the dataset for model training

```python
import json

# Load your dataset
with open("FinalDataset_ReadyForTraining.json", "r", encoding="utf-8") as f:
    data = json.load(f)

flattened_data = []

def flatten_comment(caption, parent_text, comment):
    """Create flattened entries for parent+reply comments"""
    # Ensure type is always a list
    comment_type = comment["type"]
    if not isinstance(comment_type, list):
        comment_type = [comment_type]

    # Parent alone
    flattened_data.append({
        "input_text": f"[CAPTION] {caption} [PARENT] {parent_text}",
        "sentiment": comment["sentiment"],
        "role": comment["role"],
        "type": comment_type
    })

    # Replies
    for reply in comment.get("replies", []):
        reply_type = reply["type"]
        if not isinstance(reply_type, list):
            reply_type = [reply_type]

        flattened_data.append({
            "input_text": f"[CAPTION] {caption} [PARENT] {parent_text} [REPLY] {reply['text']}",
            "sentiment": reply["sentiment"],
            "role": reply["role"],
            "type": reply_type
        })
        # Recursive if replies have nested replies
        if reply.get("replies"):
            flatten_comment(caption, f"{parent_text} [REPLY] {reply['text']}", reply)

# Process all posts
for post in data:
    caption = post["caption"]
    for comment in post["comments"]:
        flatten_comment(caption, comment["text"], comment)

# Save flattened dataset
with open("flattened_dataset_modelTrainingStage.json", "w", encoding="utf-8") as f:
    json.dump(flattened_data, f, ensure_ascii=False, indent=2)

print(f"Flattened dataset saved. Total samples: {len(flattened_data)}")
```

Figure 5.3.3.5: Flattened data for annotation

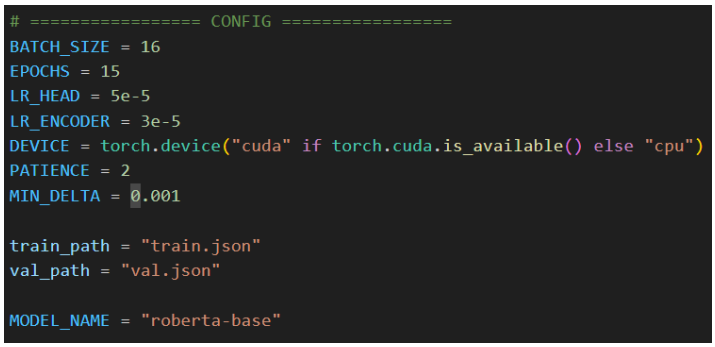
```
[
  {
    "input_text": "[CAPTION] they're so ugly and stupid [PARENT] Awww! They look so cuteyou are a stupid hater and do you wana know a secret? Fuck you! I hate all haters who h
    "sentiment": "Negative",
    "role": "cyberbully",
    "type": [
      "Insult/Threat"
    ]
  },
  {
    "input_text": "[CAPTION] they're so ugly and stupid [PARENT] this is my acc yall cant take a joke im a bts stan too",
    "sentiment": "Neutral",
    "role": "bystander",
    "type": [
      "not_cyberbully"
    ]
  },
  {
    "input_text": "[CAPTION] they're so ugly and stupid [PARENT] first of all Look at yourself in the mirror",
    "sentiment": "Negative",
    "role": "cyberbully",
    "type": [
      "Insult/Threat"
    ]
  },
  {
    "input_text": "[CAPTION] they're so ugly and stupid [PARENT] I LOVE BTS!",
    "sentiment": "Positive",
    "role": "bystander",
    "type": [
      "not_cyberbully"
    ]
  },
  {
    "input_text": "[CAPTION] they're so ugly and stupid [PARENT] Well show your face then huh?",
    "sentiment": "Negative",
    "role": "cyberbully",
    "type": [
      "Insult/Threat"
    ]
  },
```

Figure 5.3.3.6: Snippet of final dataset

5.3.4 Model Training

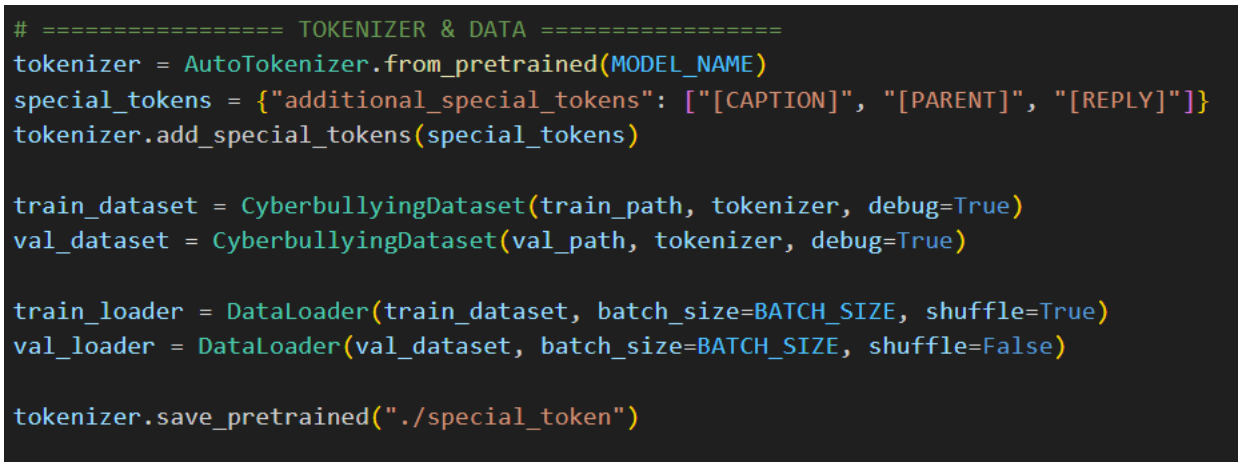
```python
# ================ EARLY STOPPING CLASS ================
class EarlyStopper:
    def __init__(self, patience=5, min_delta=0.001, mode='min'):
        self.patience = patience
        self.min_delta = min_delta
        self.mode = mode
        self.counter = 0
        self.best_metric = None
        self.early_stop = False

    def __call__(self, current_metric):
        if self.best_metric is None:
            self.best_metric = current_metric
            return False

        if self.mode == 'min':
            if current_metric < self.best_metric - self.min_delta:
                self.best_metric = current_metric
                self.counter = 0
            else:
                self.counter += 1
        else:
            if current_metric > self.best_metric + self.min_delta:
                self.best_metric = current_metric
                self.counter = 0
            else:
                self.counter += 1

        if self.counter >= self.patience:
            self.early_stop = True

        return self.early_stop
```

Figure 5.3.4.1: Early Stop Function Class



```python
# ================ CONFIG ================
BATCH_SIZE = 16
EPOCHS = 15
LR_HEAD = 5e-5
LR_ENCODER = 3e-5
DEVICE = torch.device("cuda" if torch.cuda.is_available() else "cpu")
PATIENCE = 2
MIN_DELTA = 0.001

train_path = "train.json"
val_path = "val.json"

MODEL_NAME = "roberta-base"
```

Figure 5.3.4.2: Model Configuration



```python
# ================ TOKENIZER & DATA ================
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
special_tokens = {"additional_special_tokens": ["[CAPTION]", "[PARENT]", "[REPLY]"]}
tokenizer.add_special_tokens(special_tokens)

train_dataset = CyberbullyingDataset(train_path, tokenizer, debug=True)
val_dataset = CyberbullyingDataset(val_path, tokenizer, debug=True)

train_loader = DataLoader(train_dataset, batch_size=BATCH_SIZE, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=BATCH_SIZE, shuffle=False)

tokenizer.save_pretrained("./special_token")
```

Figure 5.3.4.3: Tokenizer Configuration

```python
# ================ CLASS WEIGHTS ================
def get_class_weights(dataset, attr, num_classes):
    labels = [int(sample[attr]) for sample in dataset]
    labels = torch.tensor(labels, dtype=torch.long)
    counts = torch.bincount(labels, minlength=num_classes).float()
    counts[counts == 0] = 1e-6
    weights = 1.0 / counts
    weights = weights / weights.sum() * num_classes
    return weights

sentiment_weights = get_class_weights(train_dataset, "sentiment", len(SENTIMENT_LABELS)).to(DEVICE)
role_weights = get_class_weights(train_dataset, "role", len(ROLE_LABELS)).to(DEVICE)
type_weights = get_class_weights(train_dataset, "type", len(TYPE_LABELS)).to(DEVICE)
```

Figure 5.3.4.4: Class Weighting Function for imbalance class

```python
# ================ MODEL ================
model = MultiTaskClassifier(
    model_name= MODEL_NAME,
    num_sentiment=len(SENTIMENT_LABELS),
    num_role=len(ROLE_LABELS),
    num_type=len(TYPE_LABELS)
).to(DEVICE)

model.encoder.resize_token_embeddings(len(tokenizer))
```

Figure 5.3.4.5: Load the model

```python
# ================ LOSS FUNCTIONS ================
criterion_sentiment = nn.CrossEntropyLoss(weight=sentiment_weights)
criterion_role = nn.CrossEntropyLoss(weight=role_weights)
criterion_type = nn.CrossEntropyLoss(weight=type_weights)
```

Figure 5.3.4.6: Loss Function declare

```python
# ================ OPTIMIZER & LR SCHEDULER ================
for param in model.encoder.parameters():
    param.requires_grad = False

optimizer = AdamW(filter(lambda p: p.requires_grad, model.parameters()), lr=LR_HEAD)
num_training_steps = EPOCHS * len(train_loader)
lr_scheduler = get_scheduler("linear", optimizer=optimizer, num_warmup_steps=0, num_training_steps=num_training_steps)
```

Figure 5.3.4.7: Flattened data for annotation

```python
# ================ TRAIN LOOP ================
for epoch in range(start_epoch, EPOCHS):
    # Unfreeze BERT after 3 epochs
    if epoch == 3:
        for param in model.encoder.parameters():
            param.requires_grad = True
        optimizer = AdamW(model.parameters(), lr=LR_ENCODER)
        lr_scheduler = get_scheduler(
            "linear", optimizer=optimizer, num_warmup_steps=0,
            num_training_steps=(EPOCHS-epoch)*len(train_loader)
        )

    # ============ Training ============
    model.train()
    train_loss = 0
    for batch in tqdm(train_loader, desc=f"Epoch {epoch+1} Training"):
        input_ids = batch["input_ids"].to(DEVICE)
        attention_mask = batch["attention_mask"].to(DEVICE)
        sentiment_labels = batch["sentiment"].to(DEVICE)
        role_labels = batch["role"].to(DEVICE)
        type_labels = batch["type"].to(DEVICE)

        outputs = model(input_ids, attention_mask)
        loss_sentiment = criterion_sentiment(outputs["sentiment"], sentiment_labels)
        loss_role = criterion_role(outputs["role"], role_labels)
        loss_type = criterion_type(outputs["type"], type_labels)
        loss = loss_sentiment + loss_role + loss_type

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        lr_scheduler.step()

        train_loss += loss.item()

    avg_train_loss = train_loss / len(train_loader)
    print(f"Epoch {epoch+1} Train Loss: {avg_train_loss:.4f}")
```

Figure 5.3.4.8: Model Training Logic

5.4 System Operation

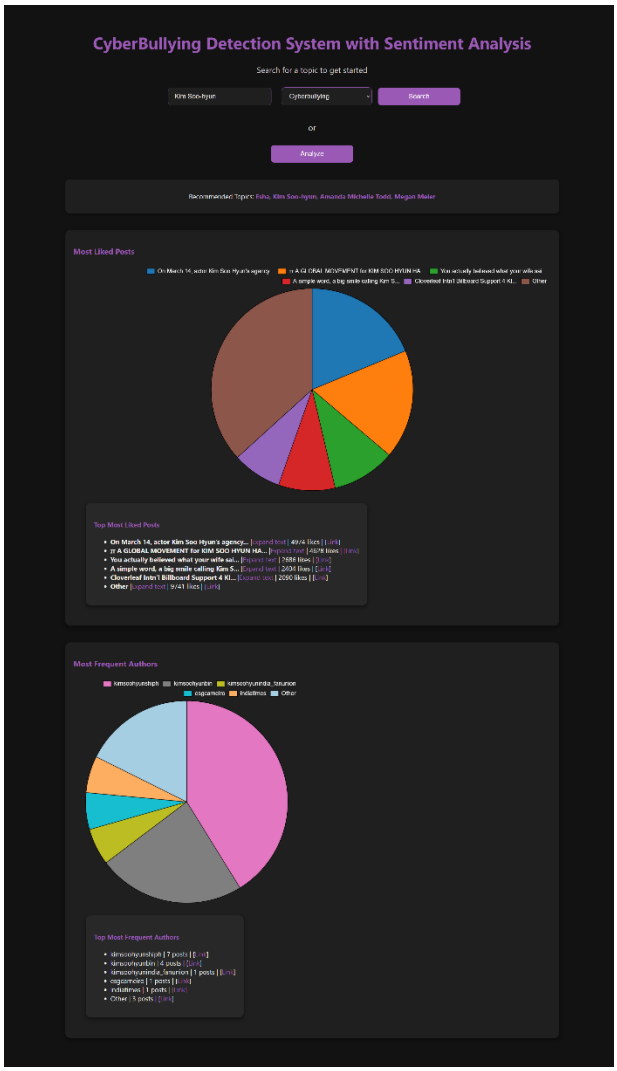


Figure 5.4.1: Main page

First of all, based on the figure 5.4.1. The user first can search any related or hot topics that potentially has cyberbully involved. The topic can be search along side with keyword for example, “cyberbullying”, “Harassment”, “Abuse”, “Hate Speech” and “Discrimination”. After that, the user can press search and the system backend will scrape through google search and collect all the Instagram post that might related to the topic. After the process is finished, all the related data is organized and display in the pie chart for example, group by most liked post and group by most frequent authors. Lastly, the user can proceed to click the ‘Analyse’ button and it will bring user to the page shows at Figure 5.4.1 to analyse the Instagram post comment by link.

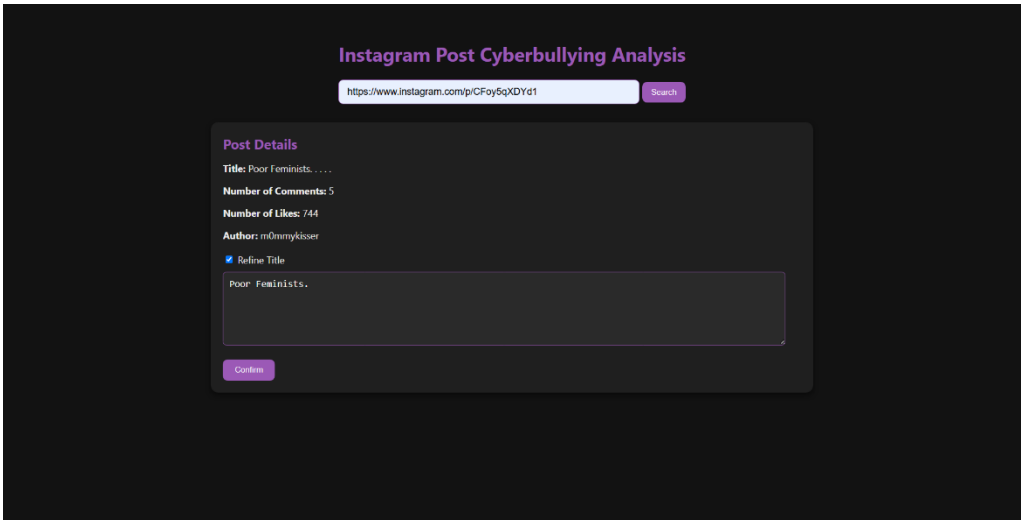


Figure 5.4.2: Analyse Comment by Post Page

Based on figure 5.4.2, this page allows user to enter a URL link of an Instagram post. When the 'search' button is pressed, the system backend will scrape and collect the metadata of the Instagram post and all the comments in the post. The scrapping method used instaloader library and http simulation of graphql request to get the metadata and comments. After the process is done, the metadata and the comments will be saved in a session storage for later use. Before proceed to next page, the user can refine the title to further make sure the title is clean and concise before going through the model for prediction.

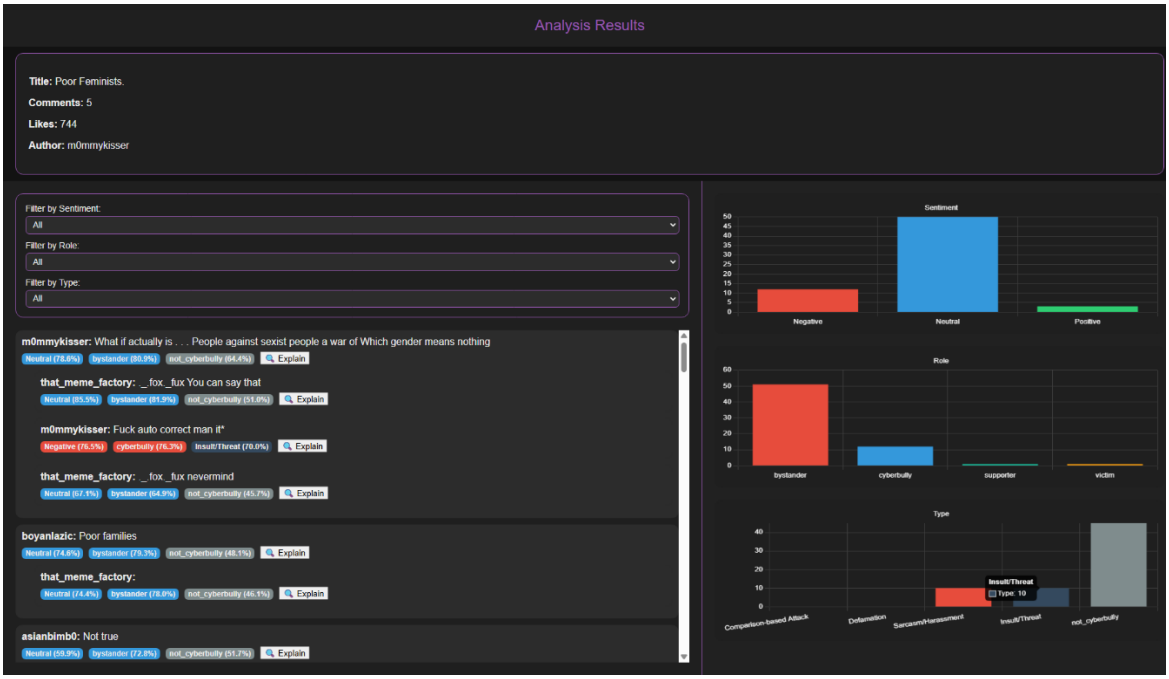


Figure 5.4.3: Analyses Results Page

Based on figure 5.4.3, this page is the analyse result page and is the continued from the figure 5.4.2. This page reads through the data stored in the session storage and display the post information. The comments will then be gone through the model to make prediction. The prediction result is then pass back for display. The comments and result are displayed in threaded form to give the user clear context about the conversation. On the right-hand side, bar charts are displayed to visualized the frequency of each label in this whole post. If user wants to learn more on how the model predict that specific comment, there is a button that allows user to perform Explainable Artificial Intelligence to visualize how the model comes out with the prediction.

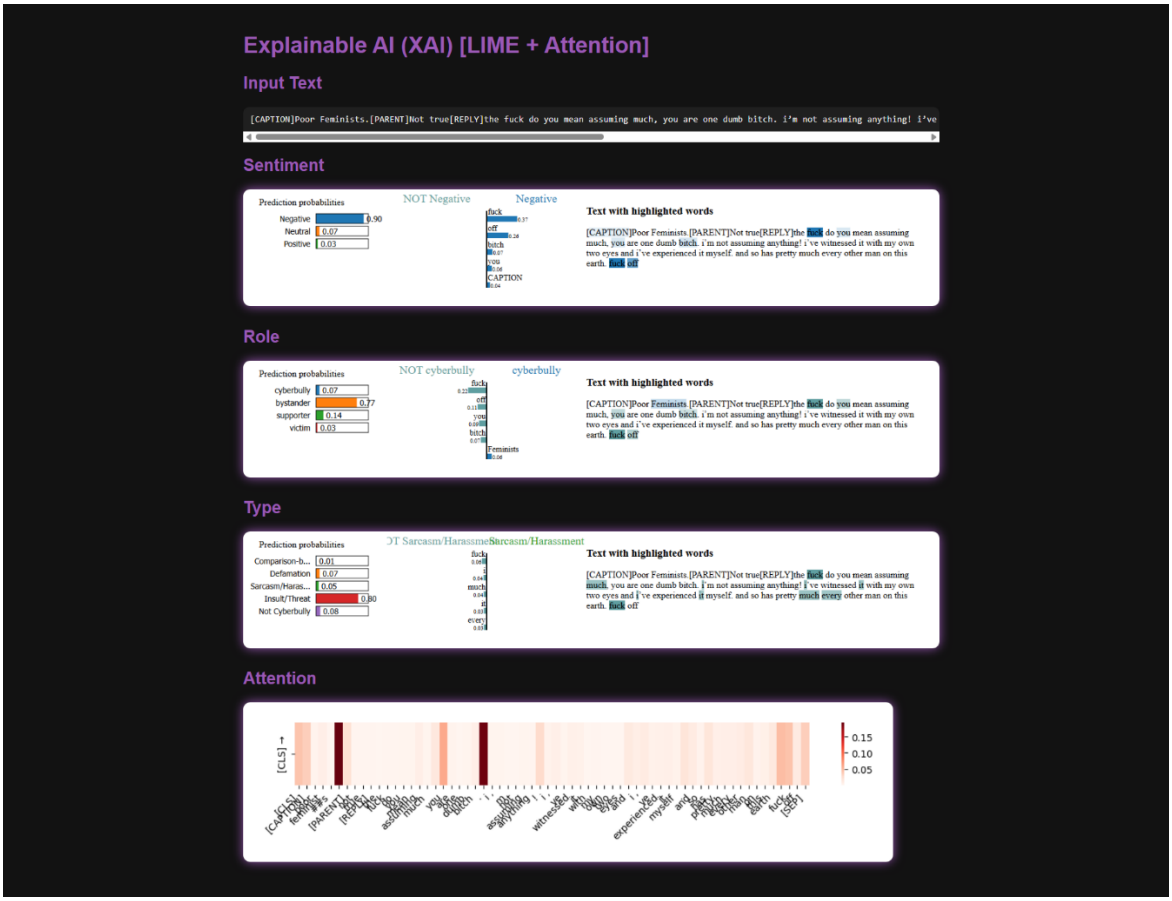


Figure 5.4.4: Explainable AI Result Page

According to Figure 5.4.4, the Explainable AI Result Page shows the result of the cyberbullying detection model in an easy-to-use format. The predicted sentiment, conversational role, and type of cyberbullying of the text under analysis are shown on the page, as well as the model attention for each word. In order to make it easier to interpret the findings, the system points out key words or phrases that have had the most significant input in the prediction to enable the user to know why a certain message was highlighted. Such transparency assists in gaining confidence in the system and produces valuable follow-up action information.

5.5 Implementation Issues and Challenges

During the development of the system that detects and predicting cyberbullying content. There are several issues faces during the implementation of the projects. To begin with, the web scraping operation was constrained because the number of comments that could be collected by the Instaloader was 1,000 comments per post on Instagram, and anything beyond this was bound to fail. The solution to this was to modify the method and use HTTP to emulate the GraphQL requests and be able to scrape posts of posts with over 1,000 comments.

Secondly, the sentiment, conversational role, and type of cyberbullying labels were annotated using the Label Studio, which was very tedious and time-consuming.

Thirdly, the quality of the dataset was a problem since very extreme or extreme comments on bullying were uncommon in the scraped data resulting in a bias in the classes with most comments being neutral or mild. This imbalance had to be addressed in the model training with extra techniques like weighted loss functions so that the model does not become one-sided to majority classes.

Fourthly, training a transformer-based model like BERT required substantial computational resources, and because of hardware constraints, training was slow and liable to memory errors, which had to be alleviated by smaller batch sizes and stage training with frozen and unfrozen layers.

Fifth, the implementation of the trained model in the Flask dashboard posed a difficulty in terms of consistency in the preprocessing pipeline applied during training with the one applied during deployment, since any inconsistency would lead to prediction errors.

Lastly, the addition of the Explainable AI module was technically intricate because the creation of a clear and easy-to-understand visual representation of the attention weights and the highlighted terms demanded more development, testing processes and is computational expensive.

5.6 Concluding Remark

To sum up, the entire process of implementing the proposed cyberbullying detection system has been outlined in this chapter, including the technical development of the system and integration of its significant elements. This was implemented through the data collection and preprocessing stage, and then the design of the detection model which is a combination of sentiment analysis and contextual understanding to extract subtle online bullying patterns. The implementation of the model with the help of a Flask-based dashboard also evidenced the way the system could be used to deliver real-time analysis and at the same time have a user-friendly interface.

Besides, the chapter pointed out the main issues in implementation like scraping restrictions, the problem of annotating data, imbalanced classes, and the incorporation of explainability modules. Nevertheless, the system has been implemented and fully ready to be evaluated in the next chapter despite these challenges. In general, the implementation phase made this conceptual design come to life and also supplied important information on the technical and practical aspects that should be taken into consideration to be applied in the real world.

Chapter 6

System Evaluation and Discussion

6.1. System Testing and Performance Metric

A systematic testing of the proposed cyberbullying detection system was done with generally accepted behaviour metrics to test the efficacy of the system. Because the system combines several subtasks (sentiment classification, role classification, and cyberbullying type classification), the evaluation framework had to indicate the overall accuracy and the robustness per-class. The accuracy was applied to give a general picture of the right prediction across all the tasks. The accuracy, weighted F1 and AUC were used to reflect the trustworthiness of the predictions on the class level. The use of Macro F1 score was made with the purpose of giving minority and majority classes the same consideration which is vital considering the imbalance of classes. It also had weighted F1 score to reflect on the class distribution and present a balanced approach in the case where the majority classes prevail. In order to ensure stability in training, early stopping was introduced along the principles of validation loss, to prevent overfitting. Furthermore, class weights were added to the loss function to address the issue of imbalance of the dataset and provide demonstration that is more open to underrepresented classes. Lastly, Explainable AI (XAI) was applied using two complementary methods of interpretability. The influential words and phrases were highlighted through attention visualization and this offers intrinsic information about the way the model pays attention to contextual information in the making of predictions. Moreover, the LIME was also implemented as a post-hoc technique to confirm and cross-verify the attention outcomes and make sure that the model prediction could be attributed in a consistent way in various viewpoints.

6.2. Testing Setup and Result

Experiment on gathered Instagram data with about 3000 comments as the dataset was performed at the testing stage. Preprocessing of the data took place to eliminate duplication, non-English text and redundant symbols. The dataset was preprocessed, and then divided into the train and the test sets with ratio 80-20. The tasks of fine-tuning two pretrain llm models were to classifies:

1) Sentiment (Negative, Neutral, Positive)
2) Role Classification (Bystander, Cyberbully, Supporter, Victim)
3) Cyberbullying Type Classification (Comparison-based Attack, Defamation, Sarcasm/ Harassment, Insult/ Threat, Not Cyberbully)

The evaluation of the performance was conducted with Accuracy, Weighted F1, Macro F1, and AUC (Macro) across all tasks.

Table 6.2 Performance Comparison between BERT-base and RoBERTa-base on Sentiment Classification

Task	**Metric**	**BERT-base**	**RoBERTa-base**
Sentiment Classification	**Accuracy**	0.7205	0.7609
	Weighted F1	0.7272	0.7674
	Macro F1	0.6753	0.7002
	AUC (Macro)	0.8676	0.8981
Role Classification	**Accuracy**	0.7174	0.7578
	Weighted F1	0.7204	0.7616
	Macro F1	0.6998	0.7463
	AUC (Macro)	0.9077	0.9073

Type Classification	**Accuracy**	0.7267	0.7453
	Weighted F1	0.7309	0.7464
	Macro F1	0.7295	0.7519
	AUC (Macro)	0.9179	0.9256

Based on Table 6.2.1, RoBERTa-base achieved a higher performance as compared to BERT -base in all the evaluated measures, which are Accuracy, Weighted F1, Macro F1, and Area Under the Curve, and so, the model exhibits its better ability to extract sentiment details that exist in the context of cyberbullying.

6.2.1 Sentiment Classification Confusion Matrix

Table 6.2.1.1: Confusion Matrix of Sentiment Classification using BERT-base

BERT-base-uncased			
True \ Pred	**Negative**	**Neutral**	**Positive**
Negative	140	39	12
Neutral	24	73	8
Positive	1	6	19

Table 6.2.1.2: Confusion Matrix of Sentiment Classification using RoBERTa-base

RoBERTa-base			
True \ Pred	**Negative**	**Neutral**	**Positive**
Negative	148	34	9
Neutral	17	80	8
Positive	1	8	17

Based on Table 6.2.1.1. and 6.2.1.2, the confusion matrices show how the predictions fall in each of the sentiment classes. The two models have strong performance of negative and neutral sentiments. RoBERTa-base has better differentiation of negative and neutral than BERT. These two models tend to over tag positive sentiment, indicating difficulties faced in handling instances of the minority classes.

6.2.2 Role Classification Confusion Matrix

Table 6.2.2.1: Confusion Matrix of Role Classification using BERT-base

BERT-base-uncased				
True \ Pred	**Bystander**	**Cyberbully**	**Supporter**	**Victim**
Bystander	**79**	**18**	**10**	**0**
Cyberbully	**30**	**104**	**12**	**2**
Supporter	**5**	**6**	**18**	**5**
Victim	**2**	**0**	**1**	**30**

Table 6.2.2.2: Confusion Matrix of Role Classification using RoBERTa-base

RoBERTa-base				
True \ Pred	**Bystander**	**Cyberbully**	**Supporter**	**Victim**
Bystander	**84**	**17**	**6**	**0**
Cyberbully	**21**	**107**	**19**	**1**
Supporter	**5**	**4**	**21**	**4**
Victim	**1**	**0**	**0**	**32**

Based on the table 6.2.2.1 and 6.2.2.2, RoBERTa-base has a high accuracy, especially regarding the identification of the Cyberbully and Victim categories, and it has a lower misclassification rate than the baseline. However, the two models also share some level of overlap of the Supporter and Bystander roles and this highlights the role of role ambiguity that exists in some conversational situations.

6.2.3 Type Classification Confusion Matrix

Table 6.2.3.1: Confusion Matrix of Type Classification using BERT-base

BERT-base-uncased					
True \ Pred	**Comparison-based Attack**	**Defamation**	**Sarcasm/Harassment**	**Insult/Threat**	**Not Cyberbully**
Comparison-based Attack	**28**	**4**	**1**	**3**	**2**
Defamation	**0**	**25**	**0**	**1**	**1**
Sarcasm/Harassment	**0**	**3**	**46**	**10**	**5**
Insult/Threat	**1**	**9**	**3**	**74**	**10**
Not Cyberbully	**1**	**11**	**8**	**15**	**61**

Table 6.2.3.2: Confusion Matrix of Type Classification using RoBERTa-base

RoBERTa-base					
True \ Pred	**Comparison-based Attack**	**Defamation**	**Sarcasm/Harassment**	**Insult/Threat**	**Not Cyberbully**
Comparison-based Attack	**32**	**2**	**1**	**3**	**0**
Defamation	**0**	**20**	**1**	**5**	**1**
Sarcasm/Harassment	**1**	**0**	**43**	**9**	**11**
Insult/Threat	**1**	**4**	**8**	**73**	**11**
Not Cyberbully	**0**	**5**	**7**	**12**	**72**

Based on table 6.2.3.1 and 6.2.3.2, the confusion table of the type classification shows that the process of fine-grained categorization is challenging, and it includes the categories such as Comparison-based Attack, Defamation, Sarcasm/Harassment, Insult/Threat, and Not Cyberbully. The two models proved to have some difficulties as seen through poor scores in overall accuracy. However, RoBERTa-base had slightly better discrimination ability between Insult/Threat and Not Cyberbully, as opposed to BERT who made even more misclassifications than RoBERTa-base. These results highlight the fact that type classification is a more complicated issue than sentiment or role-based classification tasks.

6.3. Project Challenges

The project had wider problems besides technical implementation issues. To begin with, privacy issues and sensitivity of cyberbullying content limited the data availability since ethical principles were strict, and constrained the range of available datasets.

Secondly, the data used was obtained mostly via Instagram which casts doubt on the generalizability of the statements, because the model might not exactly work equally on other communication platforms like Twitter, Tik Tok or Reddit that vary in communication patterns.

Thirdly, it was challenging to assess the cases of extreme bullying because it was not common in the data and thus it was not possible to test how robust the model was to work with extreme cases.

Fourthly, the scope of work and time limitations were one of the greatest challenges since data annotation, model training, and the integration of explainability were all time-consuming activities that had to be dealt with in the project schedule.

Fifth, the model has shown reasonable accuracy in controlled experiments but cyberbullying in real-world conversations is very subtle as it contains sarcasm, coded language, and cultural references that are hard to be deciphered by the model.

Finally, though it was demonstrated that the explainability module identified terms that influenced the final results, it was still difficult to make sure that such visualizations were meaningful and understandable by non-technical users as the question of how to make the technical in outputs and human understandable is a current difficulty.

6.4. Objectives Evaluation

Objective 1: Design and deploy a cyberbullying detection system within scheduled project time which outperforms keyword-based systems through the use of advanced sentiment analysis combined with emotion detection by LLMs.

This objective is met because the system was created with the help of transformer-based language models (BERT-base-uncased and RoBERTa-base) that enabled contextual knowledge as opposed to the simple matching of the key words. The results of sentiment classification proved to be reasonably accurate and F1, which showed that contextual modeling is effective.

Objective 2: Develop a classification system that divides the different forms of disparate manifestations of cyberbullying, which are insults, threats, and defamation and more using textual characteristics.

This objective is partially fulfilled because even though the system could distinguish between the different types of cyberbullying, such as insult, threat, and defamation. However, the accuracy of classification of finer distinctions was lost and some rare cases are struggle to classifies, which can be explained by a lack of data on some specific categories.

Objective 3: Design and implementing a system that not only includes isolated message but also the overall conversation.

The aim of this was met with the introduction of special tokens [CAPTION] [PARENT] [REPLY] to maintain a conversational context during model training. It could learn through interaction of messages as opposed to message analysis because it fed the model contextualized sequences.

6.5. Concluding Remark

The comparison between BERT-base-uncased and RoBERTa-base in solving cyberbullying detection tasks proves that RoBERTa is always better than BERT in all the evaluated measures.

RoBERTa has a better accuracy, macro-F1 and area-under-the-curve (AUC) than BERT in sentiment classification showing better distinction between negative, neutral and positive sentiment.

Regarding the role classification, RoBERTa demonstrated better results in recognizing essential roles, including cyberbully and victim since the macro-F1 score was higher, and misclassifications in minority classes were lower.

When it comes to type classification, RoBERTa also performed better than BERT especially when it comes to separating subtle forms of cyberbullying like sarcasm / harassment and insult/threat.

The influence of the confusion matrices, it can be seen that both models are effective with dominant classes, but RoBERTa has the least mistakes even on less common classes, which only emphasizes its more developed understanding of the context.

This indicates that the detection of cyberbullying behaviors can be enhanced by fine-tuning a powerful language model such as RoBERTa using domain-specific data.

In spite of the overall improvements, both models demonstrate worse performance on minority classes, thus necessitating future effort on balancing of classes, data augmentation or ensemble.

Therefore, RoBERTa-base can be the model of choice in terms of providing the optimal performance that would make this model appropriate to deploy on the dashboard and real-time cyberbullying monitoring system.

Chapter 7
Conclusion and Recommendation

7.1 Conclusion

The objective of the current work was to create and apply a cyberbullying detection system that surpasses traditional methods of key words detection by incorporating sentiment analysis, role, and type classification. The system could process textual content through the prism of a more contextualized system and evaluate dialogue excerpts at a more subtle level using transformer-based language models namely BERT base and RoBERTa base.

Empirical results have shown that RoBERTa has outperformed BERT by a significant margin in all three tasks, achieving higher accuracy, F1- scores, and the area-under-the-curve (AUC) values. In terms of affective categorization, RoBERTa had more discrimination between negative, neutral, and positive affective states. RoBERTa was more effective in the role classification domain by determining victim and cyberbully roles, and support roles remained a problem. To categorize types, both frameworks had challenges which could be explained by the fine-grained category structure; however, RoBERTa had a slightly higher overall performance.

Overall, the current project highlights the efficiency of context-sensitive deep-learning-based technologies in detecting cyberbullying. Even though the system showed strong results in terms of sentiment evaluation and the identification of the user-role, fine-grained type classification remains a significant challenge that should be further improved. Therefore, this study is an important step towards the creation of more robust and interpretive cyberbullying detection models that can potentially be deployed in the real-world online context.

7.2 Recommendation

For recommendations in the future studies, recommendation to considering to expand and diversify the dataset sample to ensure the model generalize well to different cultural, language and online sites. Increasing the size of the dataset by adding more annotations would reduce the issue of underrepresentation in certain categorical subgroups to improve the overall predictive performance.

Secondly, the system can be developed into a real-time deployment mechanism, this allows proactive detection and timely intervention in the social media or messaging ecosystems. This type of implementation would provide direct assistance to victims and enable moderators to react promptly to harmful behavior.

Third, a recommendation to conduct a concerted study on multi-modal cyberbullying detection that goes beyond text in terms of the types of content that takes in to make prediction and classification, including images, videos and emojis. Since cyberbullying often occurs in a combination of all of these media, a combination of multimodal inputs would be more thorough and genuine in the detection framework.

Lastly, a recommendation for future models can incorporate with ongoing learning and adaptive strategies to be more up to date with the changing phenomena of cyberbullying. With the help of reinforcement learning or constant learning paradigm, the system will be able to be tuned to new slang, emerging online behaviors, and changing fashions of communication without requiring extensive retraining.

REFERENCES

[1] C. Joshua, "Key Cyberbullying Statistics for 2024," *Key Cyberbullying Statistics for 2024*, Aug. 27, 2024. https://www.avast.com/c-cyberbullying-statistics

[2] D. C. Sunitharam, P. Sai Nandini, and Rakshita K, "Detection of Cyber-Bullying Through Sentimental Analysis," *International journal of soft computing and engineering*, vol. 13, no. 1, pp. 16–20, Mar. 2023, doi: https://doi.org/10.35940/ijsce.a3594.0313123.

[3] Cynthia Van Hee *et al.*, "Automatic detection of cyberbullying in social media text," *PLOS ONE*, vol. 13, no. 10, p. e0203794, 2018, doi: https://doi.org/10.1371/journal.pone.0203794.

[4] R. Zhao, A. Zhou, and K. Mao, "Automatic detection of cyberbullying on social networks based on bullying features," *Proceedings of the 17th International Conference on Distributed Computing and Networking - ICDCN '16*, 2016, doi: https://doi.org/10.1145/2833312.2849567.

[5] M. Ptaszynski, P. Dybala, Tatsuaki Matsuba, F. Masui, and K. Araki, "Machine Learning and Affect Analysis Against Cyber-Bullying," *ResearchGate*, Mar. 29, 2010. https://www.researchgate.net/publication/228791020_Machine_Learning_and_Affect_Analysis_Against_Cyber-Bullying

[6] S. Salawu, Y. He, and J. Lumsden, "Approaches to Automated Detection of Cyberbullying: A Survey," *IEEE Transactions on Affective Computing*, pp. 1–1, 2017, doi: https://doi.org/10.1109/taffc.2017.2761757.

[7] J. O. Atoum, "Cyberbullying Detection Through Sentiment Analysis," *IEEE Xplore*, Dec. 01, 2020. https://ieeexplore.ieee.org/document/9458024

[8] M. Z. Naf'an, A. A. Bimantara, A. Larasati, E. M. Risondang, and N. A. S. Nugraha, "Sentiment Analysis of Cyberbullying on Instagram User Comments," *Journal of Data Science and Its Applications*, vol. 2, no. 1, pp. 88–98, Apr. 2019, doi: https://doi.org/10.21108/jdsa.2019.2.20.

[9] B. A. Talpur and D. O'Sullivan, "Cyberbullying severity detection: A machine learning approach," *PLOS ONE*, vol. 15, no. 10, p. e0240924, Oct. 2020, doi: https://doi.org/10.1371/journal.pone.0240924.

[10] S. A. Sai and P. Pujari, "Sentiment Analysis of Cyberbullying Data in Social Media," *arXiv.org*, 2024. https://arxiv.org/abs/2411.05958v1

[11] D. Ottosson, "Cyberbullying Detection on social platforms using LargeLanguage Models," *DIVA*, 2023. https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1786271&dswid=-8569

[12] Y. Kumar *et al.*, "Bias and Cyberbullying Detection and Data Generation Using Transformer Artificial Intelligence Models and Top Large Language Models," *Electronics*, vol. 13, no. 17, p. 3431, 2024, doi: https://doi.org/10.3390/electronics13173431.

[13] Dr.R.JAYAPRAKASH, "JPJA2312-Rule-Based Cyberbullying Detection on Social Media," *JP INFOTECH*, Feb. 10, 2025. https://jpinfotech.org/project/jpja2312-rule-based-cyberbullying-detection-on-social-media/ (accessed May 01, 2025).

[14] sundaresan0502, "GitHub - sundaresan0502/Cyber-bullying-detection-system: Using a machine learning algorithm extracting and detect offensive key word from the text.," *GitHub*, Dec. 12, 2020. https://github.com/Sundaresan0502/Cyber-bullying-detection-system (accessed May 01, 2025).

[15] Lightspeed Systems, "Solutions to Prevent Cyberbullying in Schools | Lightspeed Systems," *Lightspeedsystems.com*, 2020. https://www.lightspeedsystems.com/solutions/safety-wellness/cyberbullying/

[16] ahmad-ndmz, "GitHub - ahmad-ndmz/Cyberbullying-Detection-System," *GitHub*, 2022. https://github.com/ahmad-ndmz/Cyberbullying-Detection-System (accessed May 01, 2025).

[17] shabrozkamboj, “GitHub - shabrozkamboj/Cyber-Bullying-Detection-Bot: Focused on identifying and preventing cyberbullying activities in real time. Creating a ChatBot that will work on text and audio. Languages - English, Hindi ,Hinglish.,” *GitHub*, 2023. https://github.com/shabrozkamboj/Cyber-Bullying-Detection-Bot (accessed May 01, 2025).

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” May 2019. Available: https://arxiv.org/pdf/1810.04805

[19] B. Ogunleye and Babitha Dharmaraj, “The Use of a Large Language Model for Cyberbullying Detection,” Analytics, vol. 2, no. 3, pp. 694–707, Sep. 2023, doi: https://doi.org/10.3390/analytics2030038.

[20] Y. Liu et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” arXiv.org, Jul. 26, 2019. https://arxiv.org/abs/1907.11692

[21] G. Ratnayaka, T. Atapattu, M. Herath, G. Zhang, and K. Falkner, “Enhancing the Identification of Cyberbullying through Participant Roles,” arXiv.org, 2020. https://arxiv.org/abs/2010.06640 (accessed Sep. 18, 2025).

[22] Advancing natural language processing (NLP) applications of morphologically rich languages with bidirectional encoder representations from transformers (BERT): an empirical case study for Turkish - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/BERT-input-representation-Sum-of-segment-position-and-token-embeddings-is-the-input_fig2_351386823 [accessed 18 Sept 2025]

[23] A. Vaswani et al., “Attention Is All You Need,” Cornell University, Jun. 12, 2017. https://arxiv.org/abs/1706.03762

[24] A. Adadi and M. Berrada, “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI),” *IEEE Access*, vol. 6, pp. 52138–52160, Oct. 2018, doi: https://doi.org/10.1109/access.2018.2870052.

[25] H.-Y. Chen and C.-T. Li, “HENIN: Learning Heterogeneous Neural Interaction Networks for Explainable Cyberbullying Detection on Social Media,” *arXiv.org*, 2020. https://arxiv.org/abs/2010.04576 (accessed Sep. 19, 2025).

[26] H. Mehta and K. Passi, “Social Media Hate Speech Detection Using Explainable Artificial Intelligence (XAI),” *Algorithms*, vol. 15, no. 8, p. 291, Aug. 2022, doi: https://doi.org/10.3390/a15080291.

[27] Prama, Tabia Tanzin, Amrin, Jannatul Ferdaws, M. M. Anwar, and I. H. Sarker, “AI Enabled User-Specific Cyberbullying Severity Detection with Explainability,” *arXiv.org*, 2025. https://arxiv.org/abs/2503.10650

[28] M. Sandoval, M. Abuhamad, P. Furman, M. Nazari, D. L. Hall, and Y. N. Silva, “Identifying Cyberbullying Roles in Social Media,” *arXiv.org*, 2024. https://arxiv.org/abs/2412.16417

APPENDIX

Poster