

***Dissertation on***

**“Detection of Euphemism using BERT model”**

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

*of*

**Bachelor of Technology**

**in**

**Computer Science & Engineering**

**UE21CS320A – Capstone Project Phase - 1**

***Submitted by:***

|  |  |
| --- | --- |
| **Chethan V**  **Darshan GN**  **Harshini M**  **Janav Shetty** | **PES2UG21CS143**  **PES2UG21CS151**  **PES2UG21CS193**  **PES2UG21CS210** |

*Under the guidance of*

|  |  |
| --- | --- |
| |  | | --- | | Dr. Kamatchi Priya L  Associate Professor  PES University | |

**January - May 2024**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

FACULTY OF ENGINEERING

**PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

Logo, company name

Description automatically generated

**PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

**FACULTY OF ENGINEERING**

**CERTIFICATE**

*This is to certify that the dissertation entitled*

**‘Detection of Euphemism using BERT model’**

*is a bonafide work carried out by*

|  |  |
| --- | --- |
| **Chethan V**  **Darshan GN**  **Harshini M**  **Janav Shetty** | **PES2UG21CS143**  **PES2UG21CS151**  **PES2UG21CS193**  **PES2UG21CS210** |

In partial fulfillment for the completion of sixth-semester Capstone Project Phase - 1 (UE21CS320A) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period Jan. 2024 – May. 2024. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 6th-semester academic requirements in respect of project work.



|  |  |  |
| --- | --- | --- |
| Signature  **Dr Kamatchi Priya L**  Associate Professor | Signature  Dr. Sandesh B J  Chairperson | Signature  Dr. B K Keshavan  Dean of Faculty |

|  |  |
| --- | --- |
|  |  |

**DECLARATION**

We hereby declare that the Capstone Project Phase - 1 entitled **Detection of Euphemism using BERT model** has been carried out by us under the guidance of **Dr Kamatchi Priya L** and submitted in partial fulfillment of the course requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester January – May 2024. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PES2UG21CS143**  **PES2UG21CS151**  **PES2UG21CS193**  **PES2UG21CS210** | **Chethan V**  **Darshan GN**  **Harshini M**  **Janav Shetty** |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**ACKNOWLEDGEMENT**

I would like to express my gratitude to **Dr. Kamatchi Priya L**, Associate Professor Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE21CS390A - Capstone Project Phase – 1.

I am grateful to all Capstone Project Coordinators, for organizing, managing, and helping with the entire process.

I take this opportunity to thank Dr. Sandesh B J, Professor & Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro-Chancellor, PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University, and Prof. Nagarjuna Sadineni, Pro-Vice Chancellor, PES University, for providing me with various opportunities and enlightenment every step of the way. Finally, Phase 1 of the project could not have been completed without the continual support and encouragement I have received from my family and friends.

**ABSTRACT**

Toxic behavior in online conversation is so common in today's world due to an increase in social media users. User also tend to use euphemistic language (using special symbols like \* ! @) to pass the existing toxicity detection model. So there is a need to come up with a different model to solve this problem. Our project “Detection of euphemism using BERT Model”, aims to develop a computation solution for automated detection of euphemistic language.

We focus on using advanced Natural Language Processing (NLP) techniques , particularly BERT(  Bidirectional Encoder Representations from Transformers) .BERT enables us to achieve our goal in this project better than any existing models.

The project scope would be training the model with a wide variety of datasets and developing a real-time detection system. We aim to implement the user-friendly interface and evaluate the performance using different metrics.

We also strive to solve the possible challenges like language biases and continuous adaptation that way occur during the development of the project .

Our project aims to create an impact to create a respectful online conversation and provide a base for advanced online toxicity detection.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page No.** |
|  | **INTRODUCTION** | **01** |
|  | **PROBLEM DEFINITION** | **02** |
|  | **LITERATURE SURVEY**  3.1 Paper-1  3.2 Paper-2  3.3 Paper-3  3.4 Paper-4 | **04**  04  05  06  08 |
|  | **DATA** | **10** |
|  | **SYSTEM REQUIREMENTS SPECIFICATION**  5.1 Software Requirements  5.2 Functional Requirements  5.3 External Interface Requirement  5.4 Non Functional Requirements  5.5 Safety Requirements | **12**  12  13  14  15  17 |
|  | **SYSTEM DESIGN (detailed)**  6.1 Architecture Diagrams  6.1.1 High Level System Design  6.1.2 Master Class Diagram  6.1.3 Architecture Diagram  6.1.4 External Interfaces  6.2 Design Considerations  6.2.1 Design goals  6.2.2 Architecture Choice  6.2.3 General Constraints, Assumptions and Dependencies  6.3 Design Details  6.3.1 Novelty  6.3.2 Interoperability  6.3.3 Performance  6.3.4 Maintainability  6.3.5 Security  6.3.6 Portability  6.3.7 Reusability | **18**  18  18  19  20  21  25  25  25  26  26  26  27  27  27  27  27  28 |
|  | **CONCLUSION OF CAPSTONE PROJECT PHASE - 1** | **29** |
|  | **PLAN OF WORK FOR CAPSTONE PROJECT PHASE - 2** | **30** |
| **REFERENCES/BIBLIOGRAPHY** | | **31** |
| **APPENDIX A DEFINITIONS, ACRONYMS, AND ABBREVIATIONS** | | **32** |
|  | |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
|  |  |  |
| **6.1.1** | High Level System Design | **18** |
| **6.1.2** | Master Class Diagram | **19** |
| **6.1.3** | Architecture Diagram | **20** |
| **6.1.4.1** | Main page view | **21** |
| **6.1.4.2** | Number of user | **22** |
| **6.1.4.3** | User view | **23** |
| **6.1.4.4** | Test report view | **24** |
| **8** | Plan of Work for Capstone Phase-2 | **30** |

**CHAPTER 1**

**INTRODUCTION**

In the vast landscape of online communication, the issue of toxicity looms large, necessitating effective strategies for detection and mitigation. The negative impact of this hateful content in online platforms on an individual’s mental health and the overall decorum of an online discussion is concerning. Content moderators might not be able to effectively moderate the toxic content on social media due to the sheer volume of the data available. The evolving nature of the languages and slang words also pose a huge problem. Toxic content can be of any form, including explicit language, insults, cultural nuances and subtle sarcasm, making it a complex problem to identify and classify accurately.

The detection process is further challenging due to the use of Euphemistic Language to mask the toxic content. Euphemism is the use of subtle words or visually similar symbols to represent negative words. This is done to bypass the existing toxicity detection systems. Euphemistic terms are usually coded or cryptic words and indirect language. It disguises toxic meanings behind seemingly innocuous phrases.

Another factor to be considered when determining the toxicity of a sentence is the context of the message. Context of a sentence deals with the understanding of the entire conversation, user history, cultural nuances and the real-time dynamics. The understanding of the context of the entire conversation makes the identification of foul language easier.

In our project, we aim to use the current Natural Language Processing techniques, like BERT and LSTM to address the aforementioned issues. We attempt to automate the process of euphemism detection. We strive to improve the ability of identification and classification of euphemisms using the power of the BERT model, to foster a safer environment in social media platforms.

**CHAPTER 2**

**PROBLEM DEFINITION**

This research project aims to identify and analyze the euphemistic toxic language in online communication. It uses techniques of natural language processing focusing primarily on BERT model. Our project aims to find the euphemisms that hide toxic or harmful content using which we provide automated ways to flag such language for further review.

Euphemistic language poses a challenge in its detection .This is because it often includes subtle expressions, alternative words, or symbols that are used to convey a toxic language indirectly. The traditional methods have been struggling to accurately identify and flag problematic language. Thus our research aims to leverage the advanced technologies to improve the detection of euphemisms and enhance the moderation of online communication.

The BERT model is chosen for its strong performance in analyzing natural language text. The model must recieve additional contextual information about conversations in order to enhance its ability to detect euphemistic language. By focusing on the broader context, the model can better understand the intent behind a user's words.

Our project will also take audio data in addition to the text data for a more comprehensive analysis and improved detection of toxic language. By including both data types, the model can provide a more nuanced and thorough analysis.

Technically we plan to use Python-based frameworks like PyTorch, scikit learn, and Hugging Face Transformers for training the model, its evaluation and deployment. Streamlit will be used to create

an interactive user interface which will allow our users to easily input text and audio data and receive toxicity classification results.

The project emphasizes on accuracy and efficiency in detecting euphemistic language, aiming to create a system that is effective in flagging harmful content and user-friendly. By leveraging advanced NLP models and innovative data handling, this project aims to contribute to the responsible moderation of online communication and create a safer digital space.

**CHAPTER 3**

**LITERATURE SURVEY**

**3.1 PAPER 1**

Kurita, Keita, Anna Belova, and Antonios Anastasopoulos. "Towards robust toxic content classification." arXiv preprint arXiv:1912.06872 (2019).

Link : <https://arxiv.org/abs/1912.06872>

The objective of this paper is to understand the impact of adversarial attack on existing neural toxicity detectores solve the issues in such cases. By understanding the problems in existing toxicity classifiers, the research aims to enhance the robustness of the model for more accurate toxicity detection .

The methodology used creates a lexicon of the toxic token to generate realistic adversarial attacks .These attacks involve character-level perturbation and injections and non-toxic distractors  token to classify the toxicity . The models used are ELMo and BERT toxicity detection . It also uses

contextual denoising autoencoder (CADE) as an approach to learn robustness in character level and contextual information to improve the resilience of toxicity classifiers against adversarial attacks.

Two approaches are used to solve the problem , first involves training on synthetically noised data to enhance the model ability , second is using CADE to learn robust representation that can classifier denoise the token and improve the classification.

Results show that adversarial attacks can significantly reduce the detection in the present model , with some models experiencing a drop of 50% in recall performance . The metrics used to evaluate are AUC , F1 score and recall .The study focuses on developing defenses that can withstand sophisticated adversarial attacks and improve the overall reliability of a toxicity detection system.

**3.2 PAPER 2**

Zhu, Wanzheng, and Suma Bhat. "Euphemistic phrase detection by masked language model." arXiv preprint arXiv:2109.04666 (2021)

Link: <https://www.researchgate.net/publication/354542691_Euphemistic_Phrase_Detection_by_Masked_Language_Model>

The paper introduces a new method to automatically identify multiword euphemisms that are used by fringe groups and organizations in online forums. Euphemisms are ordinary words with hidden meanings that make content moderation challenging on social media.

 The researchers of this paper have divided their work into three main stages. First the quality phrases are extracted using Autophrase which is a data driven phrase mining tool. This helps in creating a list of potential candidates which are high-quality phrases that can be used for further analysis. Then the euphemistic phrase candidates are preselected by calculating the cosine similarities of the word embeddings using word2vec.This helps in filtering out the noisy candidates and identify phrases that are semantically related to the target keywords, such as drugs with context to this paper. The last step is to rank the preselected euphemistic phrase candidates using SpanBERT.

SpanBERT was designed to predict the token spans in text which helps the researchers to analyze the long textual data more effectively. Thus by combining these three stages, we achieve the automated detection of euphemistic phrases.

In addition to the above features, this study also allows the detection of both known and potentially new euphemisms used online. This enhances the efficiency and accuracy of identifying euphemisms online. By using the paper's methods such as phrase mining and SpanBERT , we plan to improve the detection of toxic language phrases in online interactions. Overall, by referencing this paper we hope to improve the content moderation efforts on digital platforms.

**3.3 PAPER 3**

Pavlopoulos, J., Sorensen, J., Dixon, L., Thain, N., & Androutsopoulos, I. (2020). “Toxicity detection: Does context really matter?”. arXiv preprint arXiv:2006.00998.

Link :<https://arxiv.org/abs/2006.00998>

This paper talks about “Does Context Really Matter?” as the heading of the paper itself states it. So mainly in this paper the authors have focussed on two questions (a) Does context affect human judgement, and (b) Does conditioning on context improves performance of toxicity detection systems?.

To investigate or see whether considering context improves the performance of toxicity detection systems. Two data sets were created in this paper : (i) CAT-SMALL  and (ii) CAT-LARGE.

CAT-SMALL consisted of around 250 comments from Wikipedia Talk Pages, it was divided into two groups. One group had context, the other group did not have context. Each comment was analysed for toxicity by three annotators, and scores were rounded to toxic or non toxic.

CAT-LARGE consisted of around 20,000 comments, half of them were annotated with context and the rest half of them without context. This larger dataset allowed the authors for a more detailed analysis of the impact of context on the toxicity detection.

If we look at the models and approaches used in this paper, the researchers have explored various techniques. They used RNN language models and combined the consecutive comments before using RNN. They also included contextual features for sentiment classification. These methods were used to see how context influenced the toxic comments and how well the toxicity classifiers work.

The results of the analysis revealed that context had a statistically significant effect on the perceived toxicity of the comments. Approximately 5.2% of comments showed changes in toxicity labels when the context was considered. However the research found no evidence that context actually improved the performance of toxicity classifiers.

The study concluded that context can make comments seem more or less toxic, but it did not improve any system performance. So the researchers said we need bigger datasets to understand the context’s impact on toxicity analysis better.

In summary, the research paper provides important details about how context affects toxicity detection. It also shows the complexity or challenges in accurately judging the toxicity of online comments and using context to improve automated toxicity detection systems.

**3.4 PAPER 4**

Zhu, Wanzheng, et al. "Self-supervised euphemism detection and identification for content moderation." *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021.

Link: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9519422>

This paper talks about using Masked Language Model (MLM) for Euphemism Detection. It introduces the concept of self-supervision into the task of detection and classification of Euphemism. Self supervision of this task helps reduce the manual efforts required to define the euphemisms and also deals with problem of subjectivity that is introduced due to human judgement.

The authors have divided their research to focus mainly on two tasks- euphemism detection and euphemism identification. Euphemism detection is the algorithm that takes a set of forbidden words as input and tries to generate the possible variations of the words, basically euphemisms.These set of words generated may signify the same meaning as the words in the forbidden set, and is called Candidate set. The candidate set of terms is used by the model to find new euphemisms and detect forbidden words. The second task is to take a single word that is considered to be euphemistic and find out its meaning.

Both the tasks are applied in succession, like a pipeline, by the moderators to discover new euphemisms and their meanings. The authors use Masked Language Models and other self-supervised techniques for accomplishing these tasks. In this paper, the approach used is to consider an input sentence and mask the term that is likely to be an euphemism. Based on the context of its occurrence and the rest of the sentence, the meaning of the masked term is predicted.

The classification of a euphemistic term is highly dependent on the context of the sentence as the word considered may or may not be used in a veiled sense. The paper uses Masked Language Model twice- once to filter out the masked sentences and then to detect hidden meanings in the masked terms. For euphemism identification task, it uses two classifiers- coarse classifier, to filter out the sentences and multi-class classifier, to identify the meaning of the masked terms.

The model proposed in this paper can detect previously unknown euphemisms, which makes it more effective for content moderation without manual efforts. It proposes a model with self-supervision but the model needs domain-specific training. The paper considers precision at k as the evaluation metric.

**CHAPTER 4**

**DATA**

**LARGE DATASET**

Context information contains the parent comment, discussion topics . The large dataset is included in the data folder in the form of two CSV files. In this there is a gn.csv file which comprises the output of context annotations and gc.csv comprises the in-content annotation.

SOURCE : <https://github.com/ipavlopoulos/context_toxicity/tree/master/data>

**SMALL DATASET**

Content – text , parent\_text,labels . CC was originally annotated by ten annotators per post, but the parent post (the previous post in the thread) was not shown to the annotators. We call this new dataset Civil Comments in Context (CCC). Each CCC post was rated either as NON-TOXIC, UNSURE, TOXIC, or VERY TOXIC, as in the original CC dataset. We unified the latter two labels in both CC and CCC annotations to simplify the problem.

In only 71 posts (0.07%) an annotator said UNSURE, meaning annotators were confident in their decisions most of the time. We excluded these 71 posts from our study, as there are too few to generalize about.

CITE: <https://github.com/ipavlopoulos/context_toxicity/tree/master/data#how-to-cite-this-dataset>

SOURCE : <https://github.com/ipavlopoulos/context_toxicity/tree/master/data>

**SPOKEN UTTERANCE DATASET**

Files included – metadata.csv, test.csv, train.csv, trigger\_text.csv , valid.csv

It contains text data from the CMU-MOSEI dataset, with columns like FileName, text, and label2a indicating toxicity. It could be used for training and testing a model to detect social media toxicity using BERT (Bidirectional Encoder Representations from Transformers).

SOURCE : <https://github.com/Sreyan88/Toxicity-Detection-in-Spoken-Utterances/tree/main/data>

DATA CURATION SOURCES : <https://github.com/Sreyan88/Toxicity-Detection-in-Spoken-Utterances/blob/main/README.md#download-the-data>

**CHAPTER 5**

**SYSTEM REQUIREMENT SPECIFICATION**

**5.1 SOFTWARE REQUIREMENTS**

**5.1.1 Deployment**

* **Name and description**:  **STREAMLIT**, a open source python library that allow users to create interactive web applications for data science and machine learning projects with simple python script.
* **Version**: 1.32.0
* **Operating system** : Compatible with major operating system such as windows , mac and linux .
* **Source** : Official documentation [**https://docs.streamlit.io/**](https://docs.streamlit.io/)

**5.1.2 Large Language Model**

* **Name and description : BERT ,** a pre-trained natural language processing model developed by Google , capable of understanding context and semantics in text through bidirectional training on large amounts of unlabeled data .
* **Version** : BERT Large(LARGE , L-24,H-1024,16-LAYER,340M parameters)
* **Operating system** : compatible with major operating system such as windows , mac and linux .
* **Tools and libraries** : Hugging face transformer , transformers , scikit-learn , NLTK (Natural language toolkit) , Pandas , Matplotlib .
* **Source 1 :** google official documentation <https://blog.research.google/2018/11/open-sourcing-bert-state-of-art-pre.html>
* **Source 2**: hugging face documentation <https://huggingface.co/docs/transformers/model_doc/bert>

**5.1.3 Version Control**

* **Name and description : Git and Github** is a web-based platform for hosting and collaboration on software development project using Git version control system.
* **Version :** 2.44.0(latest version)
* **Operating system** : Compatible with major operating system such as windows , mac and linux .
* **Tools :** Git bash , Git GUI
* **Source** : Git official documentation  <https://docs.github.com/en>
* **Repository** : <https://github.com/chethanv-20/toxicity_detection>

**5.2 FUNCTIONAL  REQUIREMENTS**   
  
**5.2.1 Validity test on input**

* **Validity check** : Ensure the format can be only text or audio.
* **Error handling** : provide information message for incorrect input .
* **User guidance** : offer suggestion to correct input issues , enhancing user experience .

**5.2.2 Sequence of operation**

* Processing data from frontend to backend to process.
* **Large Language Model processing** : implement a stepwise approach for a large language model to analyse toxicity and capture the context .
* **Classification algorithm** : define a sequence of steps for the classification algorithm to classify text or audio accordingly.

**5.2.3 Error Handling and Recovery**

* **Detection** : implementation mechanism to to detect and handle errors related to incorrect or malformed genomic substring inputs .
* **Information Message**: Provide clear and informative error messages guiding users on how to rectify  input issues .
* **Detection**: establish error detection mechanism in case the large language the model encounters difficulties in processing specific genomic substring .
* **Recovery Strategies** : Define recovery strategies , such as reprocessing or alternative model usage , to mitigate the impact of interference errors.

**5.3 EXTERNAL INTERFACE REQUIREMENTS**

**5.3.1 User Interface**

* **Input submission** : provide users an interface to enter audio and video options .
* **Result** : Show results of toxicity in detail and show resultant flags for further validation .
* **Error Message** : Clear and descriptive error messages for invalid inputs or system errors . Suggest for users to correct input mistakes .

**5.3.2 Hardware Requirements**

The hardware of the project is important to support the processing needs of large genomic data and machine learning studies . This section provides an overview of the logical and physical characteristics of the connections between software and hardware.

* **Device Type** :We thrive to build applications that can be accessed from various devices , including desktop , laptop and smartphone when deployed to the internet and optimal display across different screen sizes and resolutions.
* **Cloud service provider** : open-source cloud provider by AWS/Google Cloud and native cloud given by streamlit for deployment will be used on the scalability , flexibility and wide range of computing resources .
* **Hardware Dependencies** : The processing and analysis tasks are done on cloud or server , so that the hardware demands are not very high . This enables you to just interact with the browser .

**5.4 NON FUNCTIONAL  REQUIREMENTS**

**5.4.1 Performance Requirement (function specific performance):**

* Large language model interface .
* Large language model interaction with BERT Model.
* with a good average response time per query .
* **Quality attribute** : speed and efficiency.

**5.4.2 Scalability** :

* Data Volume Scalability : The system must scale to accommodate growing data and ensure optimal performance as the dataset expands .
* **Quality Attribute** : Scalability , performance

 **5.4.3 Reliability , Availability and Fault tolerance :**

* The system should gracefully handle faults , ensuring continued functionality in the presence of unforeseen  errors or disruption .
* **Quality Attribute** : Robustness , reliability

**5.4.4 Resource Utilization :**

* **Objective** : The system should maintain optimal CPU and memory utilization to prevent resource saturation and ensure stability .
* **Quality Attribute** : Resource efficiency , Stability , database query response time , database queries should be executed within a time limit to maintain efficient data retrieval and processing , Database performance , efficiency .

**5.4.5 Safety requirement**

* Regulatory compliance
* the system should comply with relevant industry and regulatory standards.
* **Quality attribute** : compliance , security

**5.5 SAFETY REQUIREMENTS**

**5.5.1 Data security , privacy and confidentiality**

* all data submitted by the user must be treated confidentiality .
* **Security Measure** : Implement t encryption protocols for data  and to  prevent unauthorised access .

**5.5.2 Privacy compliance**

* use relevant data protection regulations and privacy standards.
* **Compliance Measure** : regularly check t data handling practices to ensure alignment with applicable privacy law.

**5.5.3 User interaction , communication , and clear user guidance**

* provide clear instruction and guidance to users regarding data submission and interpretation of results .
* **User interface design** : implement tooltips , help section and user-friendly documentations to assist users.

**5.5.4 Natural Language interaction safety**

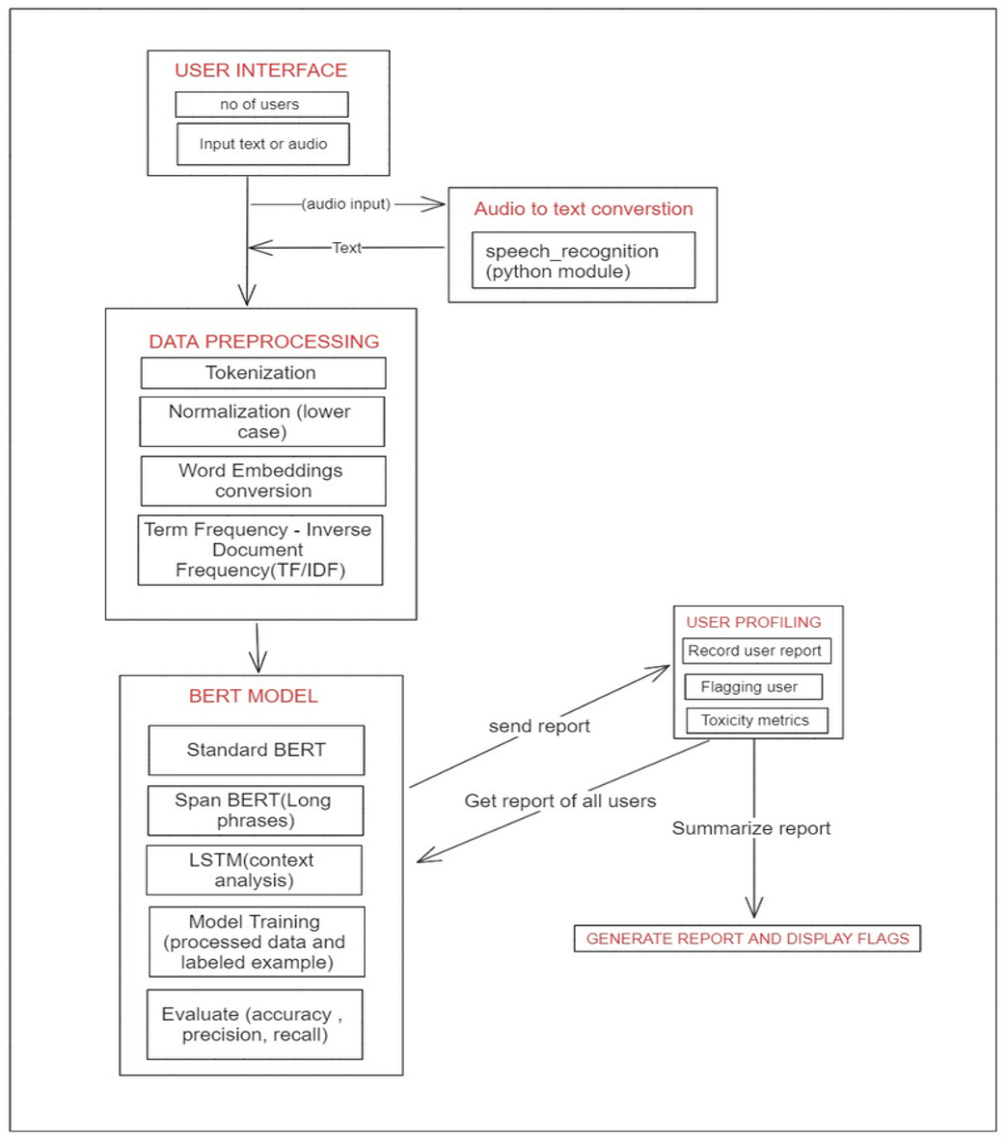
* safeguard against potentially sensitive or inappropriate language in natural language interactions .
* **Filtering mechanism** : integrate filters to identity  and handle inappropriate language , maintaining a respectful and secure environment  .

**CHAPTER 6**

**SYSTEM DESIGN**

**6.1 Architecture Diagrams**

**6.1.1 High Level System Design**



**Fig 6.1.1** **High Level System Design**

**6.1.2 Master Class Diagram**

A diagram of a computer

Description automatically generated

**Fig 6.1.2 Master Class Diagram**

**6.1.3  Architecture Diagram**

A diagram of a program

Description automatically generated

**Fig 6.1.3 Architecture Diagram**

**6.1.4 External Interfaces**

**Main Page:**

**A screenshot of a computer

Description automatically generated**

**Fig 6.1.4.1**

**Selecting number of users:**

**A screenshot of a computer

Description automatically generated**

**Fig 6.1.4.2**

**User Page:**

**A screenshot of a chat

Description automatically generated**

**Fig 6.1.4.3**

**Test Report:**

**A white background with text

Description automatically generated**

**Fig 6.1.4.4**

**6.2  Design Considerations**

**6.2.1 Design Goals**

* **Speed -** Try to generate the output of all phases in a reasonable amount of time.
* **Scalability -** should be scalable for any further additional updates of features in future.
* **Accuracy -** models which we are using should reach high accuracy.
* **Portability -** the system which we are trying to build will be platform-independent, that will work on different systems.

**6.2.2 Architecture Choice**

Our project architecture choice is a client-server architecture. In which the client will be a web browser or mobile app, that allows users to give input messages for toxicity analysis.

Then the user input messages are sent to the server, where the analysis for the toxicity detection is done or performed using the models which we have considered such as BERT and LSTM models. After the analysis is done the server will send back the toxicity analysis results to the client, which will be checked by the client or user.

**Pros -**

* The architecture that is chosen will allow the server to handle multiple client requests, so it will be easier to scale.
* It also provides centralized processing, which can be beneficial for complex hate language detection algorithms.
* Gives security where server can implement security measures such as encryption and authentication, so that the sensitive data is protected.
* Maintenance is easy, as updates and maintenance tasks can be performed centrally on the server, making it easier to manage the system.

**Cons -**

* Latency may be involved as the request has to be sent to server and processed and returned, so there may be latency in response time.
* Dependency on Server, client depends or relies on the server, so if the server is down or has any issues then the analysis might not take place.
* Complexity may get added as we need to develop both on client side as well as server side.

**6.2.3 General Constraints, Assumptions and Dependencies**  
Computational Resources

* Data Availability
* External Libraries
* Evaluation Metrics

**6.3  Design Details**

**6.3.1  Novelty :-**

Input format - We have included audio data in addition to textual data.

Handling Euphemism - Our model focuses on detecting euphemistic language,which is a new and challenging aspect of toxicity detection. This includes symbols  \*, #, !, alternate words, and other similar expressions that users may employ to convey toxicity indirectly.

Context of the conversation - Understanding the context of the conversation for accurately analyzing language and detecting toxicity.

**6.3.2 Interoperability :-**

Ensures that our system can integrate with current tools or content management systems which are used across different online platforms, like social media networks or forms.

**6.3.3  Performance :-**

Response time - For the generation of system output within a feasible time frame.

Scalability - Scaling effectively to satisfy increasing user demand without sacrificing performance or responsiveness.

**6.3.4  Maintainability :-**

**Modularity** - Designing a model with modular components that can be easily modified and replaced without affecting the other parts of the system.

**Documentation** -Provides a detailed documentation that explains the project architecture, functionality and usage instructions.

**Code quality -** Ensuring that our code is well-organised and documented so it is easier for us to understand and modify the code when needed.

**6.3.5  Security :-**

Prioritizing data security and privacy, taking strong measures to protect user inputs, result analysis and any sensitive information.

**6.3.6  Portability :-**

Our project is a platform-independent application that is able to work together with different operating systems and web browsers.

**6.3.7  Reusability :-**

Components and protocols in the architecture are designed to be reused, making it flexible for deployment in different network environments and scenarios.

The architecture can be adapted and scaled for use in various organizational settings.

**CHAPTER 7**

**CONCLUSION OF CAPSTONE PHASE-1**

In this phase of Capstone, we have explored various models and techniques available for detection of euphemism and toxic content. Our literature survey focused on models like BERT, SpanBERT, LSTM and others.

* We have only considered  the models which we think might be best suited to our research. This helps us focus on enhancing the model to increase the efficiency and accuracy.
* We have defined all the diagrams required for the project in review 3 of phase 1. We have also described architecture that includes a detailed view of bert architecture and system architecture .
* We came up with the prototype for user interface using streamlit , it includes selecting numbers , adding text and audio and a sample output with flagged users with a threshold percentage .
* We conclude our phase 1 here and our challenge is to curate a suitable dataset(that has euphemistic language in it)  that is most suitable for our projects from already existing dataset .

**CHAPTER 8**

**PLAN OF WORK FOR CAPSTONE PHASE-2**

**A white calendar with blue lines

Description automatically generated with medium confidence**

**Fig 8 Plan of Work for Capstone Phase-2**

* Above, is a visual overview of the Project Phase-2 schedule has been presented using a Gantt Chart. A Gantt chart is a bar chart that shows the planned timeline for the project's phases and key deliverables.
* Vertical Axis: The vertical axis lists the major planned milestones of phase 2 of the project. They include Technology Research, Implementation, Testing, Validation and Documentation.
* Horizontal Axis: The horizontal axis represents the project timeline with the time frame ranging from August’24 to December’24.
* Task Durations: The shaded sections within the cells represent the planned duration for each project phase.

**REFERENCES/ BIBLIOGRAPHY**

[1] Kurita, Keita, Anna Belova, and Antonios Anastasopoulos. "Towards robust toxic content classification." arXiv preprint arXiv:1912.06872 (2019).

Link : <https://arxiv.org/abs/1912.06872>

[2] Zhu, Wanzheng, and Suma Bhat. "Euphemistic phrase detection by masked language model." arXiv preprint arXiv:2109.04666 (2021)

Link: <https://www.researchgate.net/publication/354542691_Euphemistic_Phrase_Detection_by_Masked_Language_Model>

[3] Pavlopoulos, J., Sorensen, J., Dixon, L., Thain, N., & Androutsopoulos, I. (2020). “Toxicity detection: Does context really matter?”. arXiv preprint arXiv:2006.00998.

Link :<https://arxiv.org/abs/2006.00998>

[4] Zhu, Wanzheng, et al. "Self-supervised euphemism detection and identification for content moderation." *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021.

Link: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9519422>

**APPENDIX A DEFINITIONS, ACRONYMS, AND**

**ABBREVIATIONS**

**BERT**:  Bidirectional Encoder Representations from Transformers, a pre-trained natural language processing model developed by Google.

**Euphemism**: A mild or indirect word or expression used in place of a more direct or harsh one, often to avoid offense or discomfort.

**Context Analysis** : The process of examining the surrounding circumstances, environment, or background information to understand the meaning and implications of a situation or communication.

**LSTM :** Long Short-Term Memory, a type of recurrent neural network (RNN) architecture designed to model sequential data and maintain long-term dependencies.

**Toxicity** : The quality of being harmful, poisonous, or detrimental, especially in the context of social media, referring to content that is offensive, abusive, or harmful to others.