# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The term "AI-generated text detection" describes the application of AI technology to recognize and detect text material produced by an AI system that might include inaccurate or deceptive information or content that is in violation of the law. As deep learning technology advances and becomes more widely used, AI-generated text detection will become more crucial in a variety of industries, including news media, network security, and public opinion tracking. Since generative models like GANs (Generative Adversarial Networks) and RNNs (Recurrent Neural Networks) are frequently used in text generation tasks due to the rapid advancement of AI technology, particularly the recent advances in deep learning, the problem of false and fraudulent information has become more serious. A significant amount of information, particularly a significant proportion of text material produced by AI systems, cannot be efficiently managed or monitored because to the Internet's fast expansion in information. For this reason, research on AI-generated text detection is crucial to preserving cyberspace order and safeguarding user rights and interests. As people increasingly rely on social media, news websites, and other online platforms to receive information and share ideas, rumours and incorrect information disseminated by AI-generated text also pose a severe risk to public opinion and have social repercussions.

Deep learning techniques are essential for text detection produced by AI. Because deep learning algorithms have strong pattern recognition and feature extraction capabilities, they perform exceptionally well in AI-generated text detection tasks. By constructing intricate neural network structures, deep learning algorithms are able to identify hidden patterns and features in text data. For instance, deep learning models like the Transformer model have been widely employed in text data processing tasks in the field of natural language processing. Large-scale datasets can be used to train deep learning algorithms, which will enhance model performance. Having authentic and sufficiently diverse labelled data is crucial for guaranteeing model correctness and resilience in AI-generated text identification tasks. Furthermore, word vector representations, attention mechanisms, and other approaches in natural language processing can be paired with deep learning algorithms to enhance comprehension and analyse text data and mine potential false information or offending content from it.

# Chapter 2

# Literature Survey

The development of synthetic text creation has given rise to some very important questions with implications for society and the internet. It is anticipated that applications driven by LLM would supplant a sizable human workforce [17]. Numerous duties in the fields of education, law, advertising, science and creative writing, entertainment, and many more are already being handled by chatbots. Additionally, LLMs provide attackers with more pervasive and hazardous talents [18]. It is more difficult to identify instances of academic dishonesty, bogus news and reviews, spam and phishing, and other abuses [14], [19]. All educational institutions view ChatGPT as a significant challenge because conventional plagiarism detection equipment is unable to distinguish text produced by artificial intelligence. Since the Turing test and its application to chatbot evaluation, the issue of AI-generated text, or more broadly, machine-generated text, detection, has gained a lot of attention [20]. Rather than concentrating on hybrid or human detection techniques, we examined automated strategies in this work. According to Crothers, Japkowicz, and Viktor, there are two broad categories into which autonomous AI-generated text identification techniques can be divided: feature-based and neural language models [14]. However, some studies concentrated on identifying particular areas, like academic environments [21]–[25], scientific settings [26]–[31], and fake news/fake reviews/misinformation [32]–[36], among others.

In the work by **Hao Wang and et.al** [1], they created an effective artificial intelligence (AI)-generated text identification model based on the BERT algorithm, offering fresh concepts and approaches to related issues. In order to assure data quality and accuracy, a number of procedures were performed to process the text during the data preparation stage. These procedures included converting the text to lowercase, word splitting, removing stop words, stemming extraction, removing digits, and removing redundant spaces. They discovered that the model functioned successfully during training by splitting the dataset into a test set and a training set in the ratio of 60% and 40%, and then tracking changes in the accuracy and loss values during the training process. The BERT model was able to detect AI-generated text with high accuracy, and the prediction results are gradually approaching the real classification results. The accuracy increased steadily from the initial 94.78% to 99.72%, while the loss value decreased from 0.261 to 0.021 and converged gradually. Subsequent examination of the training and test set results indicated that the test set's average loss is 0.0917, indicating a marginally greater loss value, compared to the training set's average loss of 0.0565. In terms of accuracy, the test set's average accuracy was 97.71%, and

the training set's average accuracy reached 98.1%. These figures being relatively close to one another, suggested that the model has high generalization capabilities. In conclusion, trials demonstrated good accuracy and stability for the AI-generated text detection model based on the BERT algorithm suggested in this paper, offering a useful solution for related domains.

In the work by **Trung T. Nguyen and et.al** [2], the authors proposed two methods to recognize text generated by AI: a) text similarity based methods and b) machine learning based methods. They gathered two distinct datasets—Wikipedia-based articles and US Election 2024 News items—in order to test the proposed techniques. Next, they implemented extracting four distinct sets of hand-crafted features from the raw text: Topic Modeling Features, Basic NLP Features, Term Frequencies and Ngram Features, and Other features (NER count, grammar error, and readability score). Next, in order to prove that the suggested techniques work, three experiments were conducted. In the first experiment, three machine learning models were trained using RF, SVM, and XGBoost on the categorization of text created by ChatGPT or authored by humans after extracting the handcrafted features. The F1 score for the RF and XGBoost models was 0.9993, indicating good performance. This method was observed to have great detection accuracy and works well in any domain. In order to implement this strategy, the authors gathered the human-written ground truth texts and prompted artificial intelligence (AI) technologies, such ChatGPT, to create an artificial text by constructing relevant questions from the ground truth texts. The second experiment was designed to demonstrate the second approach to detecting AI-generated text. Based on subject modelling and keyword extraction, they created related queries for any text, whether it was authored by a person or produced by AI. Text similarity based on cosine was used as the result. The results of the second experiment showed that there is a clear classification boundary between human-written text and AI-generated text. This approach proved effective and is found to work in general situations, as no ground truth data needs to be collected.

This study by **Eriksen and et.al** [3], shed light on potentially significant disparities between human-written and AI-generated academic abstracts. The study found distinct discriminatory patterns through a thorough examination of numerous textual characteristics, including type token ratios, grammar, n-gram distributions, and perplexity. When compared to their human-written counterparts, the AI-generated texts demonstrated superior token-level predictability and grammatical precision. The methods in the study, particularly the feature-based one, showed impressive accuracy and confidence in differentiating between information created by humans and artificial intelligence. With perplexity standing out as a particularly relevant metric in addressing the difficulty of AI authorship attribution, such precision highlights the significance of the selected traits. They framed two methods of machine learning techniques for attribution of AI authorship: one based on features, the other on text. Whereas the latter made use of the precomputed attributes detailed in the analysis, the former depended only on the language of the abstracts. Because it was difficult to derive a comprehensive view from a textual examination of abstracts alone, the text-based technique was used as a standard to evaluate the effectiveness of the feature-based approach. Three machine learning algorithms were used to compare the two methods: Multinomial Naïve Bayes, Random Forest, and Logistic Regression. For contrast, a zero-rule baseline technique for predicting the majority class—the most-frequent label classifier baseline—was presented. The precomputed features—perplexity, grammar, type-token ratio for 1-, 2-, and

3-grams, frequency of function words, and average token length—were used in the feature-based technique to estimate authorship attribution. These features proved to have a strong discriminating value based on our previous investigations. Grid search with 5-fold cross-validation was used to optimize for precision scores during hyperparameter tweaking. From the best-performing model, the most significant features were taken after training. They also used a text-based method as a comparative. They employed the TF-IDF vectorizer to numerically transform the text, emphasizing n-grams to capture token sequence contexts. Like the primary approach, hyperparameters were optimized through grid search with 5-fold cross-validation, and based model selection on precision scores. The Logistic Regression model proved superior, exhibiting a test precision score of 0.988.

The work by **Giulia Socolof and et.al** [4] used style embeddings as a key component in order to: (1) further pretrain an existing style embedding model on texts authored by humans and artificial intelligence, in order to determine if the author of a text was a human or an AI, and (2) to investigate hostile environments by figuring out if a model can recognize and categorize hybrid human-AI material with accuracy. The authors used three basic classification heads on top of the embedding model: a Multi-Layer Perceptron,K-Nearest Neighbors and K-Means Clustering.The results showed to develop a robust representation of natural language, embedding space, and human and mixed text, surpass baselines on classification, and do so with a more comprehensible and effective model than existing detection techniques. They framed the AI text detection problem as a generalization of the authorship verification problem, in which the task is to identify the author of a text among a group of recognized authors. The primary focus was on content-independent style embeddings, which were first described by **Wegmann et al. in 2022** [5]. The training process learns a representation of authorship in embedding space by contrasting and comparing text authors while accounting for the text's content. On top of the embedding model, they trained the above mentioned classification heads.

In order to further identify an author in a multi-author manuscript, writing style change detection seeks to pinpoint the text location at which the author shifts. The paper by **Haoyang Chen and et.al** [6] presents the experiment on PAN 23's shared task. In order to improve the encoder's ability to process sentences with similar styles and increase the distance between the embedding representations of paragraphs with different styles, the authors applied the comparative learning method to the analysis of writing style. Specifically, they optimized the sentence segment embedding output produced by the pretraining model's encoder. Using the improved encoder, they analyzed he tag data along with sample pairs of paragraphs and categorize them using a full connection layer to create sentence embeddings.Through experiments, they obtained F1-scores of 0.9145, 0.8203, and 0.6755 on Task 1, Task 2, and Task 3 of the official test set, respectively.

# Chapter 3

# Methodology

Google's pre-trained language model, known as BERT (Bidirectional Encoder Representations from Transformers), is based on the Transformer architecture. The BERT model's two basic steps are pre-training and fine-tuning. It can perform exceptionally well in a variety of natural language processing tasks by learning text data representations using large-scale unsupervised textual data representation. Figure 3.1 depicts the BERT model's structure. First, using the Transformer design, BERT creates a bidirectional encoder that, in contrast to the conventional unidirectional language model, can better capture the link between words by simultaneously accounting for contextual information. When working on natural language tasks, this helps BERT comprehend the semantics and sentence structure in a sentence more effectively. Large-scale unlabelled text data is used by the BERT model for pre-training, where the textual representation is learned through two tasks: Next Sentence Prediction (NSP) and Masked Language Model (MLM). Figure 3.2 depicts the BERT model's training procedure. BERT can be optimized for a range of natural language processing tasks, including question-and-answer systems, named entity recognition, text categorization, and more, following pre-training. To complete a task using supervised learning, one need only add an output layer to the pre-trained model during the fine-tuning phase and merge it with labelled data. Since BERT has learnt a common language representation, it just needs to fine-tune a small number of parameters to produce better results when encountering new jobs. BERT, being a groundbreaking model for natural language processing, has set new standards and produced state-of-the-art performance in multiple tests. Its strength is evident in both its outstanding performance and its great degree of adaptability and versatility.

The various objectives identified for the project are:

- To AI generated text detection using an encoder based transformer architecture.
- To fine-tune the transformer to do classification of HWT and MGT.
- To study the distribution of the embeddings given by the transformer for the HWT and MGT.
- To fine-tune the transformer on custom dataset.

The proposed methodology for the work is as shown in Figure 3.3.

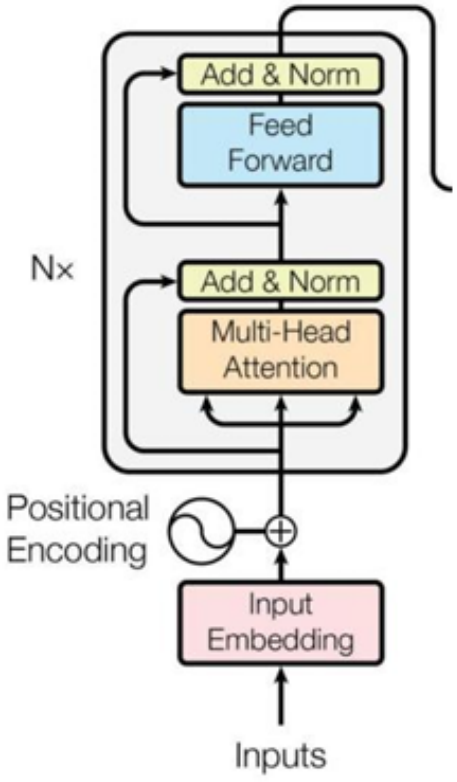


Figure 3.1: The structure of BERT model

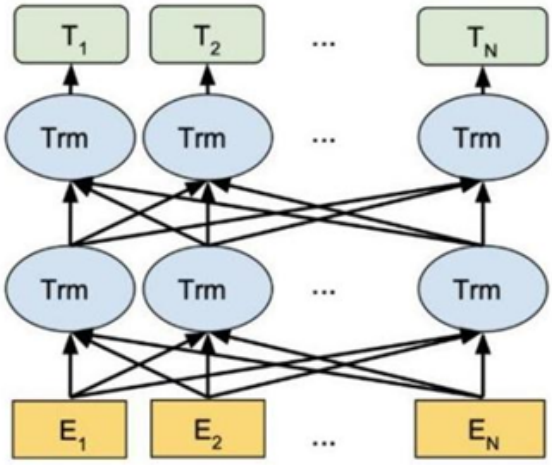


Figure 3.2: The training process of the BERT model

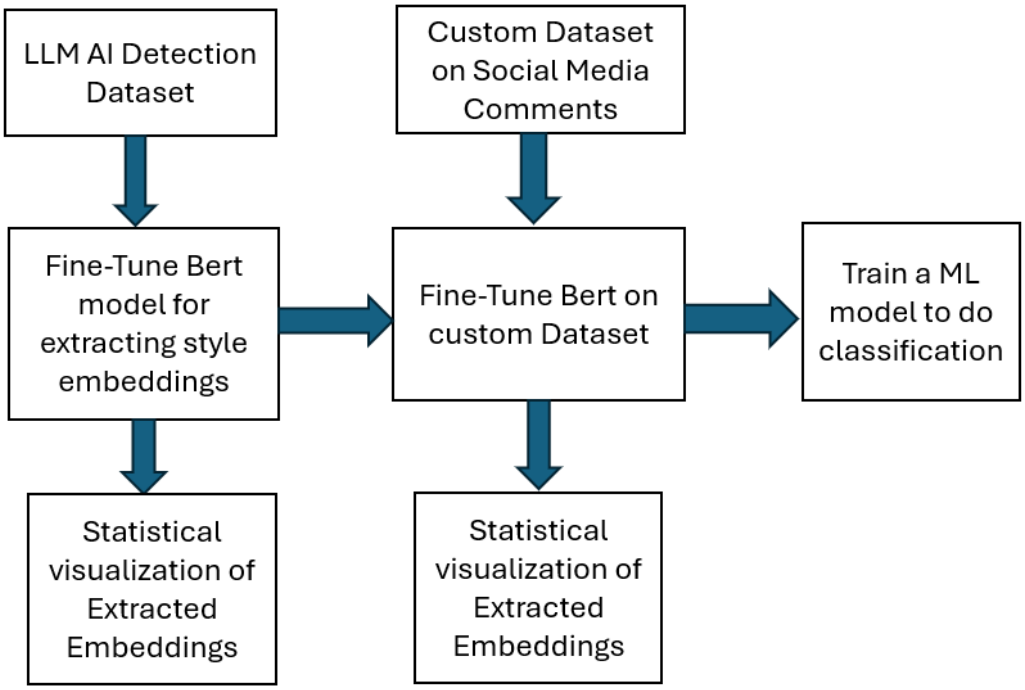


Figure 3.3: Proposed Methodology

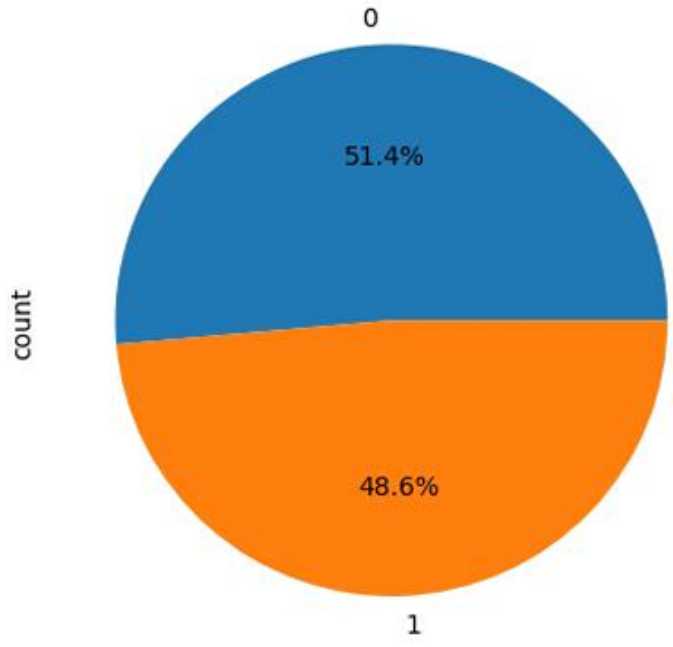


Figure 3.4: The proportional distribution of the two classes

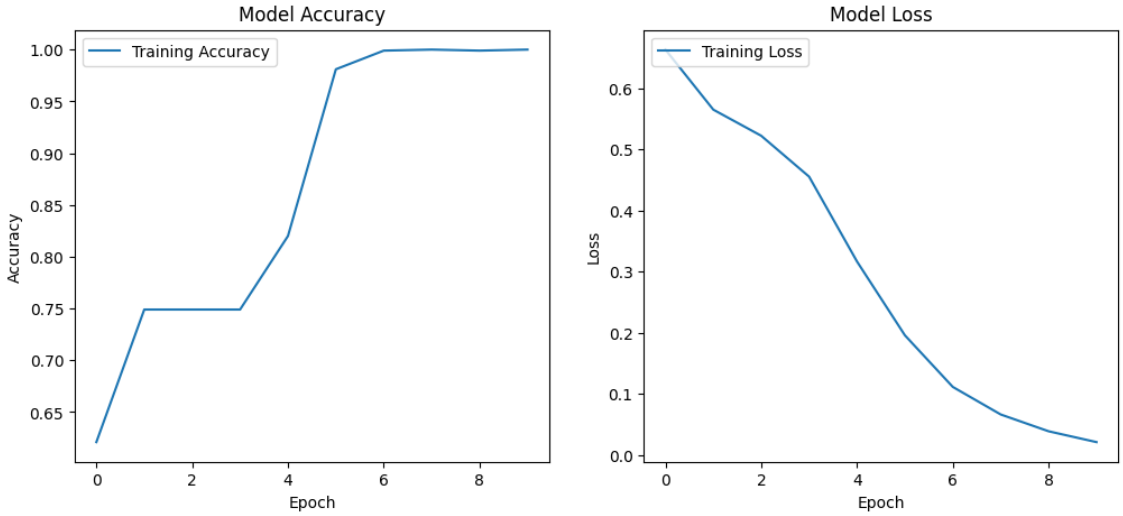


Figure 3.5: Training Accuracy and Training Loss plot

## 3.1 LLM AI Detection Dataset

This work uses a public dataset to conduct experiments that include a significant number of texts generated by AI and texts that are not, labelled as 0 and 1, respectively. Dataset contains more than 28,000 essay written by student and AI generated. The proportional distribution of the two classes is shown in Figure 3.4

## 3.2 Fine-Tune Bert model to extract style embeddings

A BertSequenceClassifier with a binary classification head was fine-tuned on the identified LLM AI detection dataset to classify an input text as AI generated or no. The training configuration was:

- Batch size : 128
- Number of epochs : 10
- Model used : bert long uncased
- Tokenizer used : bert long uncased tokenizer

The plot of training accuracy and training loss with epoch is shown in Figure 3.5

## 3.3 Statistical visualization of extracted embeddings

The texts from the validation set were provided as input to the fine-tuned BertSequence-Classifier and the embeddings were extracted from the last hidden layer of the model.The embedding was a vector of dimension 768 which was then decomposed to 2 principal components using PCA. The decomposed vectors were then standardised and a distribution plot

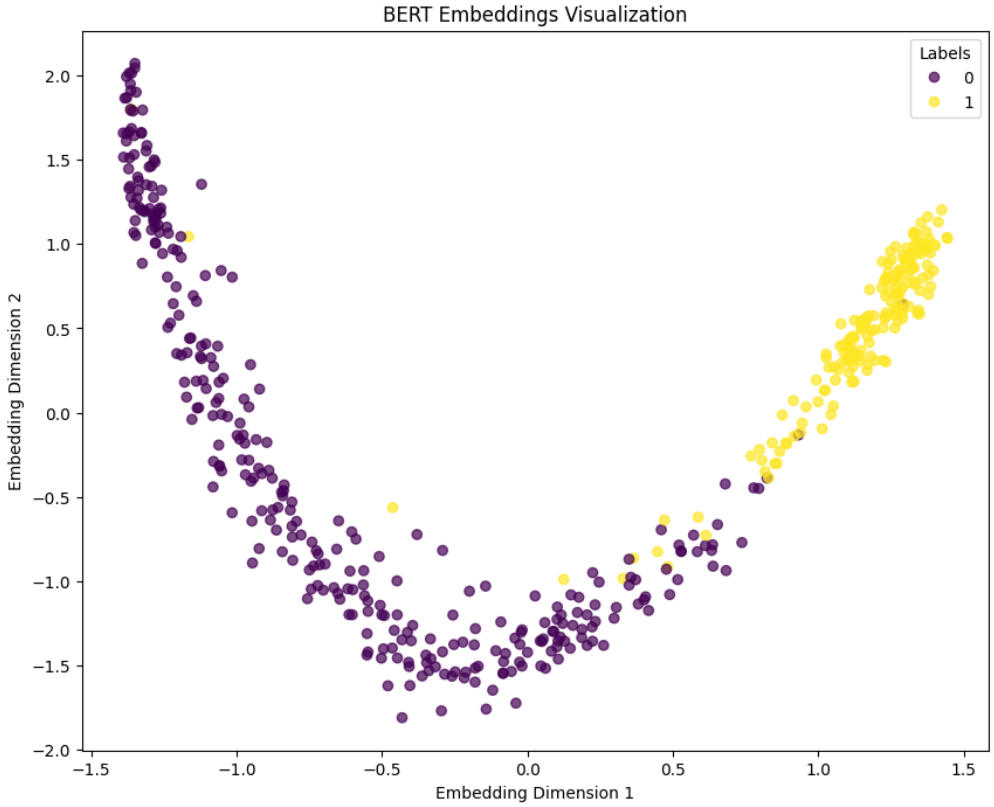


Figure 3.6: Distribution embeddings of validation dataset

was made with the class labels as the legend. The distribution the standardised 2 component PCA is shown Figure 3.6, which shows a clear decision boundary between the embeddings of the two classes.

## 3.4 Custom Dataset

A custom dataset of Youtube comments was made. Relevant political comments relating to India Elections 2024 were gathered from comments of various videos in Youtube and labelled as Human written. In order to make sure that the comment was human written, the comment was passed as input to the fine-tuned Bert model for checking. The class to which the text belongs was predicted and also the embeddings were extracted. The comment was used to prompt Google Gemini to generate three similar comments which were then passed to the Bert model to extract its embeddings. The Cosine Similarity score of the human written comment with each individual AI generated comment was found to be more than a threshold angle signifying significant decision gap between the two classes and also the authenticity of the Human written comment. The complete process of custom dataset generation is shown in Figure 3.7.

An example of the embedddings distribution of a HW comment and its three AI counterparts are shown in Figure 3.8.

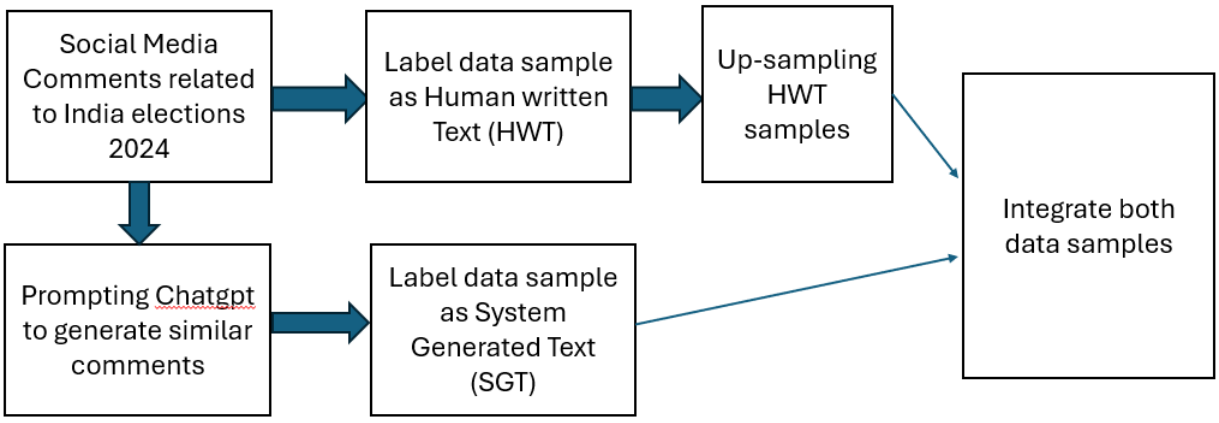


Figure 3.7: Custom dataset generation process

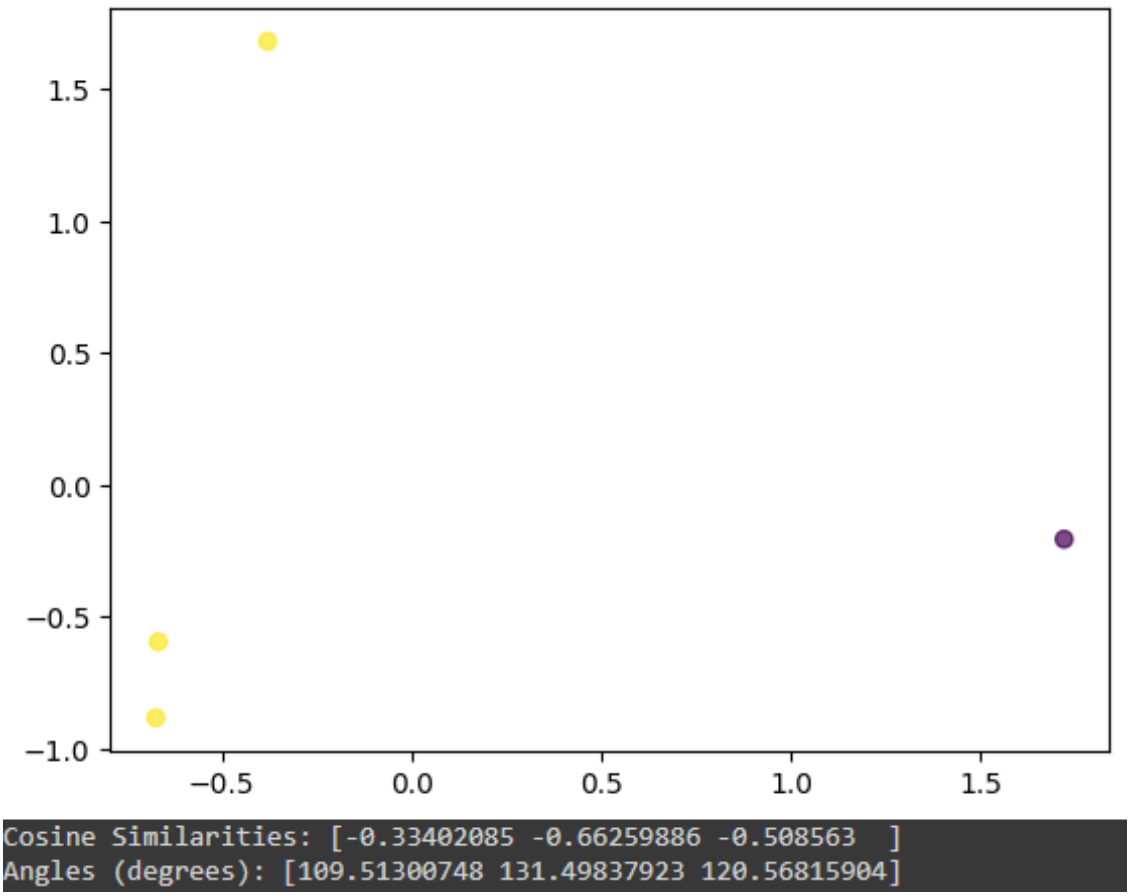


Figure 3.8: Distribution of embeddings of Human written comment and AI generated comments

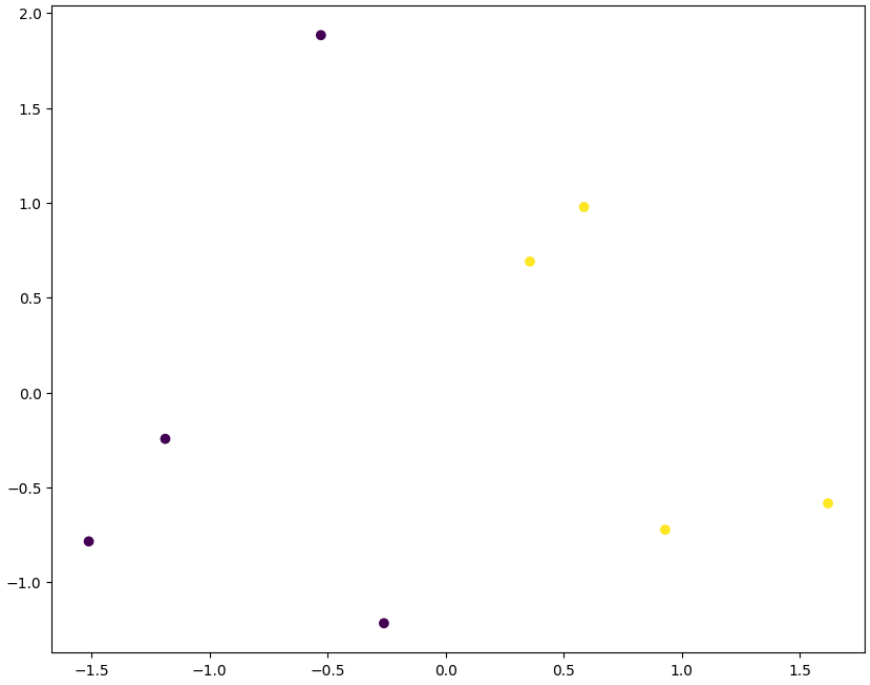


Figure 3.9: Experiment 1

## 3.5 Experimenting with the model to find out its efficacy

In order to visualize the efficacy of the model in extracting embeddings of human written text and AI generated text and distribute the embeddings with a considerable decision distance, the following validation experiments were done:

- Four human written comments and their AI generated counterparts were provided as input to the Bert model to extract it s embeddings an plotted as shown in Figure 3.9. The plot showed a significant decision gap between the embeddings of human written text and AI written text.

- The plot of the extracted embeddings was done with pair-wise legend as shown in Figure 3.10. Each color represent a single Human written and corresponding AI generated comment pair. The plot showed that there is a significant decision gap between each human written comment and their corresponding AI counterpart.

- For one comment from the four set of comments, six AI generated counterparts were prompted. The corresponding embeddings distribution can be seen in the Figure 3.11. This plot shows that the embeddings of all six AI generated counterparts of the same human written comment are clustered far away from the embedding of the original comment.

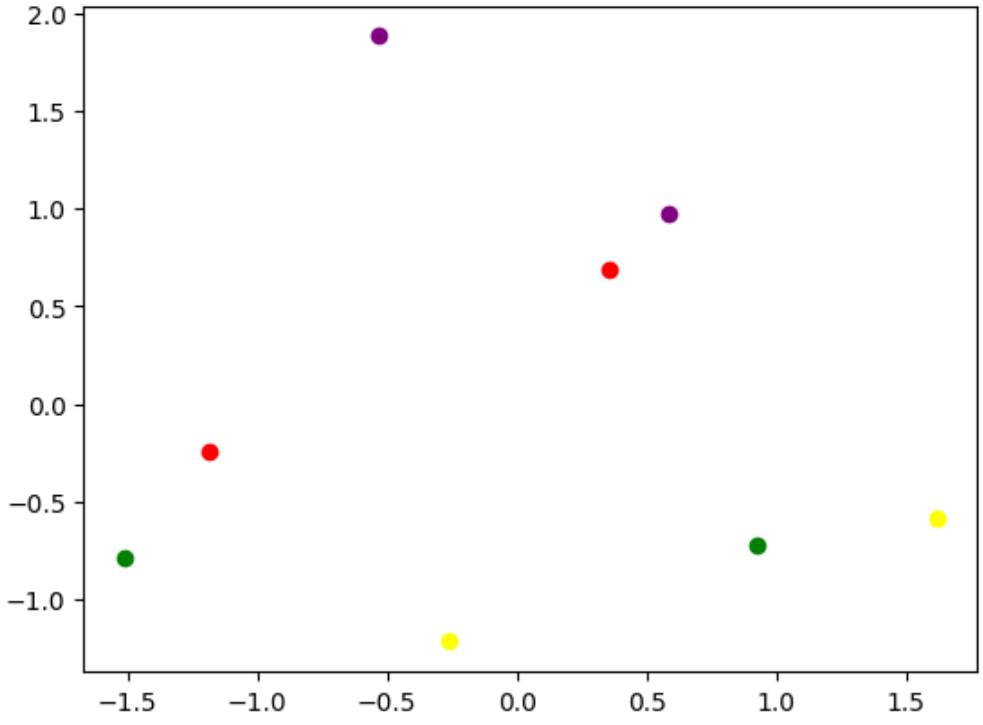


Figure 3.10: Experiment 2

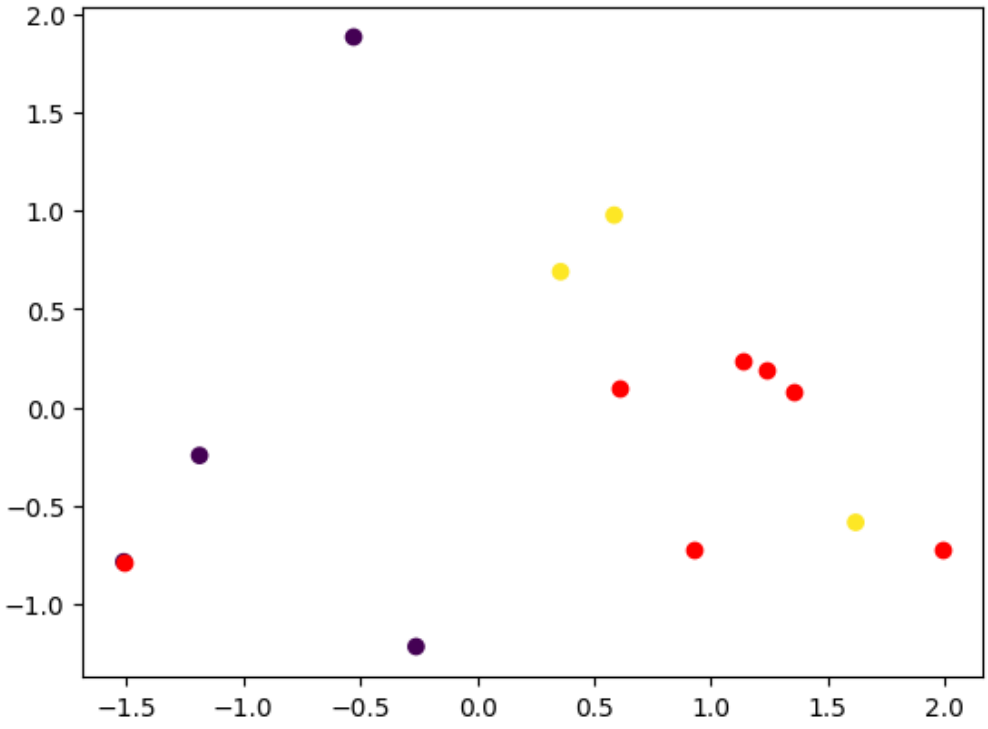


Figure 3.11: Experiment 3

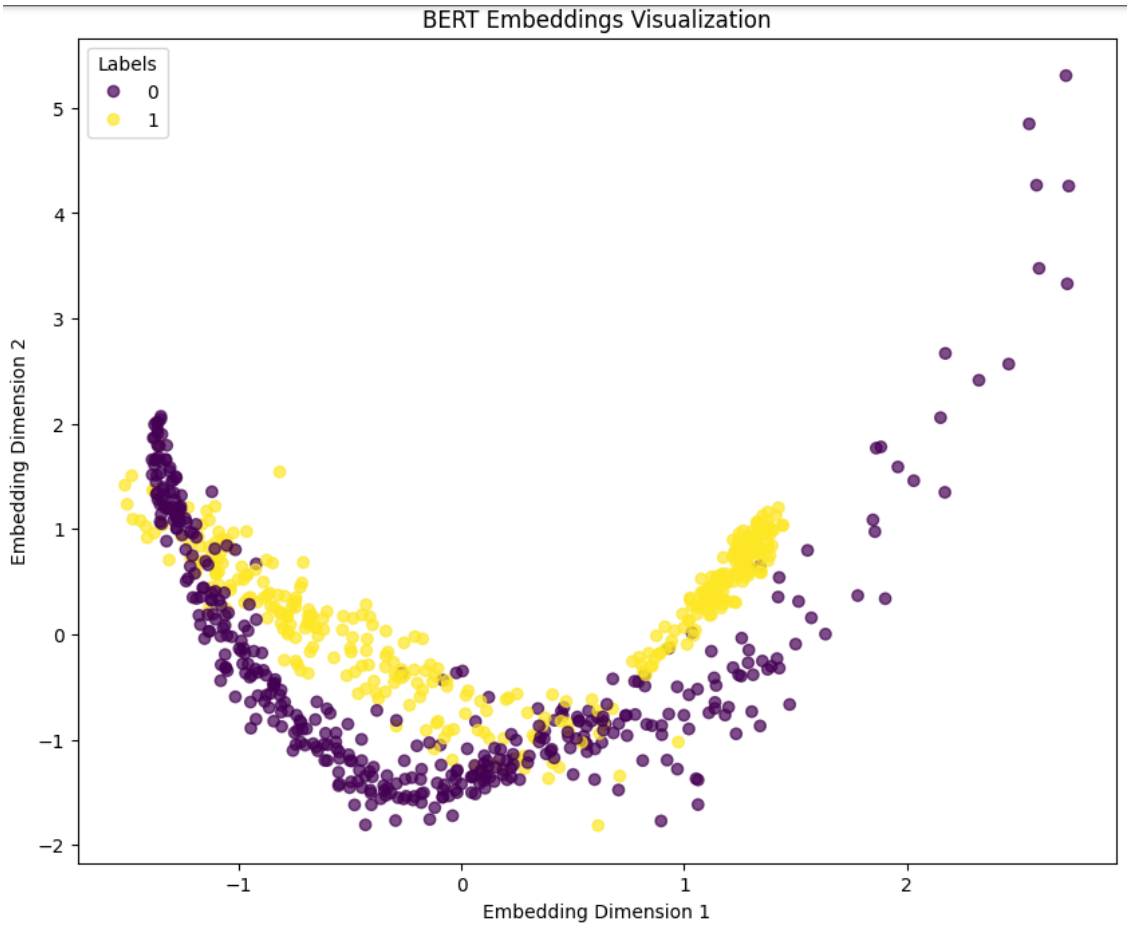


Figure 3.12: Validation distribution of embeddings extracted from fine-tuned model

## 3.6 Upsampling Human Written class

The human written comments were upasampled to meet the class imbalace issue. For this, Embedding augmenter and WordNet augmenter were used.

## 3.7 Fine-tuning on custom dataset

The model fine-tuned on LLM AI detection dataset was fine-tuned on the custom dataset. The distribution of the extracted embeddings from the validation set is as given in Figure 3.12.

# Chapter 4

# Conclusion

The project focused on leveraging BERT (Bidirectional Encoder Representations from Transformers) for detecting AI-generated text, demonstrating its effectiveness in distinguishing between human-written text (HWT) and machine-generated text (MGT). The methodology involved fine-tuning a BERT-based model on a public dataset of texts, followed by experimentation with a custom dataset to test the model's adaptability and performance. The project showcased the potential of transformer architectures in natural language processing tasks, particularly in identifying and classifying AI-generated content.

The project's methodology was rooted in the transformer architecture, specifically utilizing BERT's bidirectional encoder capabilities to understand contextual relationships within text. The BERT model was fine-tuned on a large dataset comprising over 28,000 text samples, with equal representation of AI-generated and human-written texts. This fine-tuning process enabled the model to learn nuanced differences between the two types of text, allowing for more accurate classification. Key experiments were conducted to visualize and validate the model's performance. After fine-tuning, the model was tested on a validation set, where embeddings were extracted from the final hidden layer. These embeddings, initially 768-dimensional vectors, were reduced to two principal components using Principal Component Analysis (PCA). The subsequent distribution plots revealed clear decision boundaries between human-written and AI-generated texts, underscoring the model's ability to differentiate between the two classes.

A series of experiments were performed to assess the model's efficacy in various scenarios. The first experiment involved providing the model with a set of human-written comments and their AI-generated counterparts. The resulting embeddings were plotted, revealing significant decision gaps between the two types of text. This gap indicated the model's strong capacity to distinguish between subtle differences in textual content that are indicative of either human or AI authorship.

In another experiment, the embeddings of human-written comments and their AI-generated counterparts were plotted with pair-wise legends. This visualization further confirmed that the model could consistently separate human and AI-generated text, even when the AI-generated text was designed to closely mimic human writing. The embeddings of AI-generated text clustered together and were clearly distinguishable from those of human-written text.

To test the model's robustness, a third experiment was conducted in which a single human-written comment was used to prompt an AI model to generate multiple counterparts. The embeddings of these counterparts, when plotted, clustered far away from the original human-written comment, reinforcing the model's capability to maintain high accuracy even when faced with multiple variations of AI-generated text.

The project also explored the creation of a custom dataset, focusing on politically relevant YouTube comments related to the India Elections 2024. The model's performance on this custom dataset was critical in testing its adaptability to real-world data, where AI-generated content may not strictly adhere to patterns seen in training data.

To address class imbalance, the project employed upsampling techniques using Embedding Augmenter and WordNet Augmenter. These techniques helped balance the dataset, ensuring that the model received an equal representation of both classes during training. This approach not only improved the model's accuracy but also its generalizability, making it more reliable in diverse applications.

In conclusion, the project successfully demonstrated the effectiveness of a fine-tuned BERT model for detecting AI-generated text. The model's ability to create a significant decision boundary between human-written and AI-generated content highlights the potential of transformer-based architectures in natural language processing tasks. By applying advanced techniques like PCA for dimensionality reduction and upsampling for class balancing, the project achieved robust results that underscore the model's applicability in various real-world scenarios. This work paves the way for further exploration into AI-generated content detection, offering a promising solution for maintaining authenticity and integrity in digital communication.

# References


[1] Wang, Hao, Jianwei Li, and Zhengyu Li. "AI-Generated Text Detection and Classification Based on BERT Deep Learning Algorithm." arXiv preprint arXiv:2405.16422 (2024).

[2] Nguyen, Trung T., Amartya Hatua, and Andrew H. Sung. "How to Detect AI-Generated Texts?." In 2023 IEEE 14th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), pp. 0464-0471. IEEE, 2023.

[3] Eriksen, Helene FL, Christopher MJ André, Emil J. Jakobsen, Luca CB Mingolla, and Nicolai B. Thomsen. "Detecting AI Authorship: Analyzing Descriptive Features for AI Detection." In NL4AI@ AI* IA. 2023.

[4] Socolof, Giulia, and Ritika Kacholia. "Fast, Interpretable AI-Generated Text Detection Using Style Embeddings." Custom project for CS224N, Department of Computer Science, Stanford University, n.d.

[5] Wegmann, Anna, Marijn Schraagen, and Dong Nguyen. "Same author or just same topic? towards content-independent style representations." arXiv preprint arXiv:2204.04907 (2022).

[6] Chen, Haoyang, Zhongyuan Han, Zengyao Li, and Yong Han. "A Writing Style Embedding Based on Contrastive Learning for Multi-Author Writing Style Analysis." In CLEF (Working Notes), pp. 2562-2567. 2023.