Project Final Report

Introduction/Background

The launch of ChatGPT on November 30, 2022 incurred proliferation of artificial intelligence (AI). With over 180 million monthly users, ChatGPT, along with other AI tools has made tremendous impacts on academic institutions, workplace, and everyday communication. Large language models (LLMs) represent large, complex deep neural networks, pre-trained on vast datasets sourced from the web. Primarily, LLMs serve as the foundational technology of ChatGPT-like conversational bots. In a recent report by OpenAI, it's acknowledged that their latest AI-text detector isn't entirely dependable. The evaluation revealed that the classifier accurately identifies only 26% of AI-generated text, while also misclassifying 9% of human-authored text (false positives). However, when dealing with shorter text, machine learning models can be trained to improve accuracy. Mitrović et.al (2023) fine-tuned a Transformer-based model for detection of short ChatGPT-generated reviews that achieved an impressive accuracy of 79%.

Problem Definition

Generally, detecting AI-generated text is a challenging problem. Adversarial training has also been used in this field. To tackle paraphrasing attacks, Hu et al. (2023) proposed a novel detection framework called RADAR. This framework utilizes an adversarial learning approach to concurrently train both a detector and a paraphraser.

Traditional models sometimes achieve better prediction results. A. Occhipinti et al(2022) utilized 12 traditional machine learning methods to predict a public spam corpus and achieved a 94% F1 score using random forest and XGBoost in text classification .

Our initiative is to train both supervised and unsupervised models that could distinguish between text generated by AI and that written by humans. The dataset used for this project includes 788,922 unique values and five columns, including the text (AI or human), prompt identification, text length, and word count. Access to the dataset is here.

Methods

1. Data Preprocessing:

In the data preprocessing stage, we enhanced our dataset's readiness for analysis by employing several key techniques. First, we removed stopwords to filter out common but non-informative words, thus streamlining the dataset. Next, stemming and lemmatization techniques were applied to consolidate words to their root forms, ensuring consistency across variations. Finally, we utilized Term Frequency-Inverse Document Frequency (TF-IDF) to quantify the importance of words within the dataset's documents, providing a weighted framework for our machine learning analysis. This comprehensive preprocessing approach was used to refine our dataset, setting a solid foundation for accurate model training and text classification.

2. Feature Extraction

For the feature extraction, we employed the Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer, specifically choosing an n-gram range from 1 to 3 and enabling the sublinear term frequency scaling. The rationale behind using TF-IDF with an n-gram range is to capture not only the importance of individual words but also that of word pairs and triples, which can provide more context and may be indicative of the text's origin (AI or human). The sublinear TF scaling was applied to reduce the bias that might occur due to different lengths of text, making word counts more comparable across documents.

3. Model Selection

We selected a Logistic Regression model for its efficiency and effectiveness in binary classification tasks. Logistic Regression is particularly suitable for this type of problem because of its ability to handle sparse matrices, which is a common trait of TF-IDF vectorized text data. Moreover, it's a probabilistic model which can provide a probability score indicating the likelihood of a text being AI-generated, adding interpretability to our predictions.

Model 2 - K-means Clustering (Unsupervised Learning)

Next, we applied the K-Means clustering algorithm to the TF-IDF vectorized text. K-Means is an unsupervised learning method that organizes data into K distinct clusters based on feature similarity, without requiring predefined labels. We chose K-Means for its computational efficiency and simplicity, making it ideal for handling large datasets and for performing exploratory analysis.

Key Reasons for Choosing K-Means:

1. **Efficiency and Scalability:** K-Means can quickly process large volumes of data, crucial for dealing with extensive text datasets.
2. **Simplicity:** The straightforward approach of K-Means in segmenting data into clusters based on feature variance helps in intuitively identifying natural groupings in the dataset.
3. **Exploratory Analysis:** This method allows us to explore potential natural distinctions between AI-generated and human-generated texts, which are indicated by their differing TF-IDF features.

In our implementation, we configured K-Means to form two clusters, hypothesizing that the texts would naturally separate into AI-generated and human-generated groups. We then evaluated the clustering outcome to see if this method could effectively distinguish between the two types of text.

Model 3 - Recurrent Neural Network - LSTM (Supervised Learning)

We also implemented the RNN model, given its capability of processing sequential data that makes it particularly useful for this task. We reckon that the sequences of words may encapsulate important information that distinguishes AI and humans, since the words may interrelate and convey complex semantic information. Through these patterns, we are hopeful that the RNN would capture these subtle nuances. We believe that RNN is advantageous in the following aspects:

1. Ability to store past information: the hidden layer of RNN can store and use previous inputs for future predictions through its recurrent workflow. This can be particularly useful to capture the styles of texts generated by AI or written by humans.
2. Prevention of gradient vanishing problem: the integration of gate mechanism addresses the gradient vanishing problem by controlling the entry of new input. The gates allow for retention or deletion of existing information so that the gradients for backpropagation will not shrink excessively.
3. Flexibility in data handling: LSTM can handle input with varying length, which is often the case with text data. This flexibility makes it more efficient at recognizing the variations in sentence structures.

We set the input dimension of the embedding layer to be 20,000, allowing the model to capture a wide range of vocabulary, and the input length is set to 200 given the average length of our texts. We added a bidirectional layer so that the LSTM could process the sequence both forward and backward, followed by a fully connected layer that uses ReLU as activation function. The final output layer uses a sigmoid activation function since our task is binary classification.

Model 4 - Generative Adversarial Model (Unsupervised Learning)

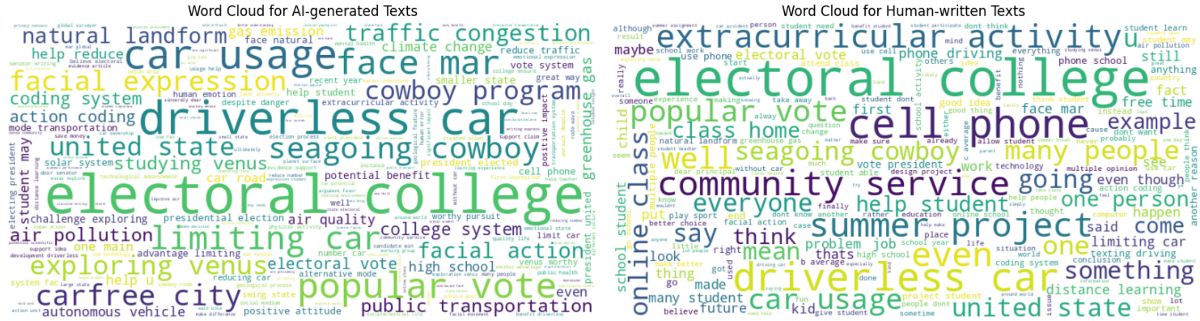
After preprocessing the text data, we first applied the generative adversarial model to the TF-IDF vectorized text. GAN is an artificial intelligence learning method that can generate new data instances that resemble the training data, so the model consists of two parts. The first one is the generator, which learns to generate plausible data, and the initial input is random noise. GAN actually has several advantages in NLP classification problems:

1. **Data Augmentation:** One of the biggest challenges in NLP problems is lack of sufficient labeled data. Applying the GAN method can efficiently solve this issue.
2. **Feature Learning:** GAN encourages the model to capture complex patterns in the original data, potentially leading to better performance in downstream classification tasks.
3. **Improving Robustness:** By being exposed to adversarially generated data, the model can handle a variety of inputs and random noises.

Results and Discussion

1. Visualization

We generated word clouds for two types of text separately. This visualization shown comprises two word clouds, representing the most frequently occurring words in two datasets of texts: one generated by artificial intelligence (AI) and the other written by humans. The size of each word in the cloud indicates its frequency, with larger words being used more often within the texts.


Word Cloud for AI-generated Texts

Word Cloud for Human-written Texts

In the word cloud for AI-generated texts, prominent words include "car," "electoral," "facial expression," "climate change," "driver-less," "college," and "seagoing cowboy." This suggests that the AI-generated texts cover a diverse range of topics. The word cloud for human-written texts has "extracurricular," "electoral," "cell phone," "community service," "college," and "summer project" as some of its most prominent words. The human-written texts seem to discuss academic and societal activities, political themes, and technology's role in everyday life.

These visualizations are useful in quickly identifying the themes or topics that are most prevalent in large sets of texts, allowing us to compare and contrast the focuses of AI-generated versus human-written content. Notably, the term "electoral" features prominently in both clouds, indicating it's a common subject for both AI and human writers in this dataset. The presence of unique terms in each cloud also suggests differences in the content or style between the two types of text.

2. Quantitative Metrics

Our project is a classification machine learning problem because the response of the model is a binary variable. So we decided to use accuracy, precision, recall, F1 score, and ROC Curve.

2.1 Accuracy: It is simply a ratio of correctly predicted observations to the total observations. The formula for calculating accuracy is:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

However, accuracy alone can be misleading if the class distribution is imbalanced so we also put other metrics like F1 score into consideration.

2.2 Precision: It is the number of correct positive results divided by the number of all positive results, including those not identified correctly. The formula for precision is:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

2.3 Recall: It is the number of correct positive results divided by the number of all relevant samples. The formula for recall is:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

2.4 F1 score: The F1 score is the harmonic mean of precision and recall, taking both false positives and false negatives into account. It is a good way to show that a classifier has a good value for both recall and precision. The formula for the F1 score is:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

2.5 ROC Curve: The Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It is created by plotting the true positive rate (TPR, or recall) against the false positive rate (FPR, or 1-specificity) at various threshold settings

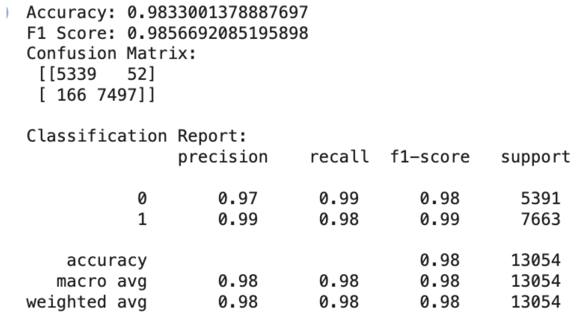
3. Analysis of Logistic Regression

The model has an accuracy of approximately 98.33%, indicating that it correctly predicted the outcome for that percentage of the test set.

For the negative class (0), precision is 0.97, and for the positive class (1), it's 0.99.

For the negative class, recall is 0.99, and for the positive class, it's 0.98. This means the model is capturing 99% of the actual negative class and 98% of the actual positive class.

The F1 score is about 98.57%, which implies a very strong balance between precision and recall.



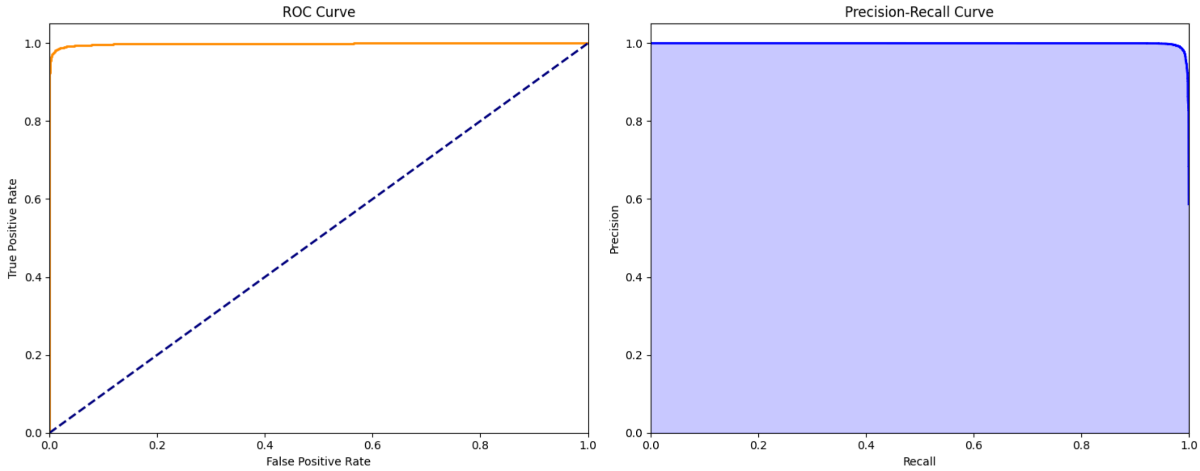
```
Accuracy: 0.9833001378887697
F1 Score: 0.9856692085195898
Confusion Matrix:
 [[5339   52]
 [ 166 7497]]

Classification Report:
               precision    recall  f1-score   support

           0       0.97      0.99      0.98      5391
           1       0.99      0.98      0.99      7663

    accuracy                           0.98     13054
   macro avg       0.98      0.98      0.98     13054
weighted avg       0.98      0.98      0.98     13054
```

The Precision-Recall Curve is close to the top right corner, indicating both high precision and high recall across different threshold settings. The area under this curve (AUC for Precision-Recall Curve) also suggests a high measure of separability performed by the model.

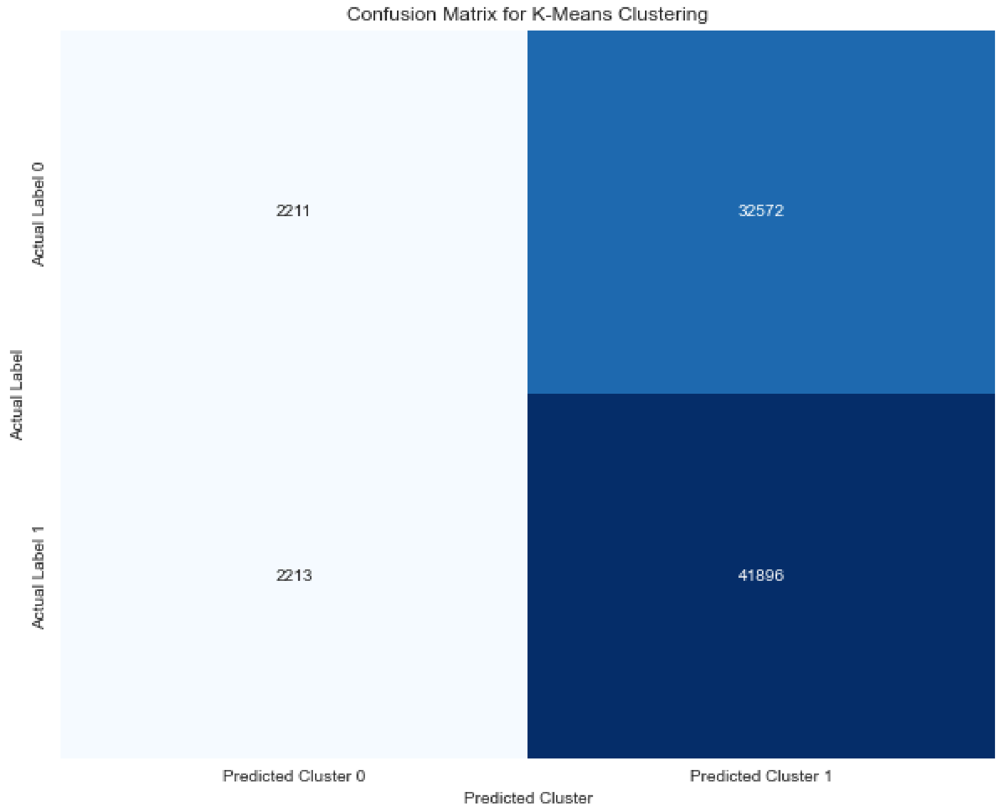


Overall, the model exhibits excellent performance metrics across the board, indicating it is well-fitted to the test data with a high predictive ability. However, it is crucial to also validate these results against an independent dataset or through cross-validation to ensure that the model is not overfitted to the test set.

Model 2 - K-means Clustering (Unsupervised Learning)

Visualization & Analysis

The application of the K-Means algorithm to differentiate between two types of text generated a distinct partition, as indicated by the confusion matrix. The visualization of the confusion matrix provides a clear and immediate understanding of the clustering performance. We note that there are 2,211 instances where texts from Cluster 0 correctly match Label 0, and 41,896 instances where texts from Cluster 1 correctly correspond to Label 1. However, there is a significant number of texts (32,572) from Label 0 that were misclassified into Cluster 1.



The confusion matrix illustrates the K-Means algorithm's high propensity for placing texts into Cluster 1. This skew may be indicative of an imbalance in the distinctiveness of the linguistic features between the two types of text. Cluster 1 could potentially be capturing features more prevalent across the dataset, which may include common linguistic patterns shared by both human and AI-generated texts.

Quantitative Metrics - Silhouette Score

In the quantitative analysis of our K-Means clustering, we have employed the silhouette score as a metric to evaluate the quality of the clusters formed. The silhouette score is a measure of how similar an object is to its own cluster compared to other clusters. The values range from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

For our dataset, the silhouette score is approximately 0.0020. This score is close to zero, indicating that no substantial structure has been found. This low score indicates that the cluster overlap is considerable, making it difficult to distinguish between them confidently.

Model 4 - GAN models

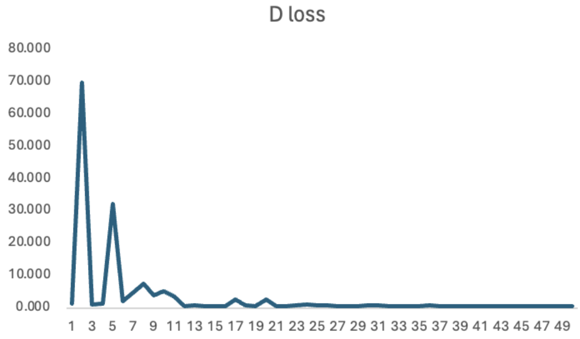
Visualization & Metric Analysis

We choose three metrics for the GAN model, and they are Generator loss, Discriminator loss, and accuracy separately.

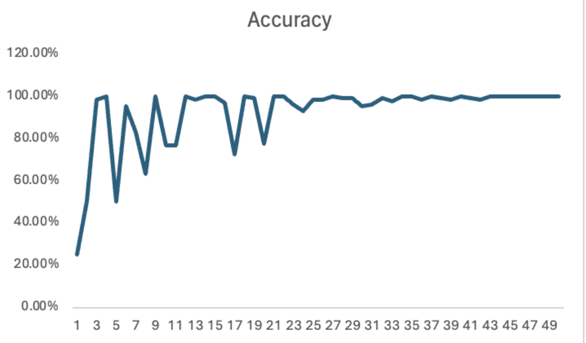
1. Generator Loss: Generative loss quantifies how well it can trick the discriminator into believing that the generative data is human-generated. We used the cross-entropy loss as the loss function. The generator loss is often expressed as:

$$L_G = -\frac{1}{m}\sum_{i=1}^{m}\log(D(G(z^{(i)})))$$

Where G is the generator network, D is the discriminator network, z is a batch of random network vectors, and m is the batch size.



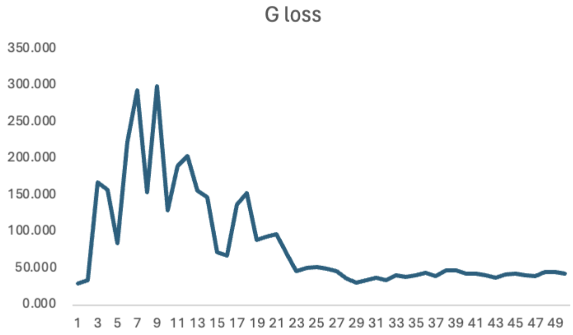
2. Accuracy: It measures how well the discriminator perform on distinguishing between human-generated and AI-generated data, and how it can classify human-generated data into the correct categories.



3. Discriminator Loss: It measures how well the GAN model classifies real data as human-generated and AI-generated data as fake. It often uses binary cross-entropy loss as evaluation metrics. The discriminator loss can be expressed as:

$$L_D = -\frac{1}{m}\left[\sum_{i=1}^{m}\log(D(x^{(i)})) + \sum_{i=1}^{m}\log(1 - D(G(z^{(i)})))\right]$$

Where x is the batch of real data.



With the number of epochs growing, both D loss and G loss is increasing rapidly and become steady after 30 epochs. Also, the accuracy increases to nearly 100%, indicating that the GAN model can accurately classify between human-generated text and AI-generated text.

4. Next Steps

Next, our project will try to apply more models to see the performance, including more sophisticated machine learning models such as Transformer-based architectures, to do the prediction, and compare the performance of these methods. Also, we will conduct extensive hyperparameter optimization for each model to identify the best settings that maximize performance on validation datasets. In addition, we will try to acquire more data to enhance the model's ability for identification. Finally, we can collaborate with domain experts to enhance the models' capabilities and address ethical considerations around AI-generated text detection.