Supervised Machine Generated Text Detection Using LLM Encoders In Various Data Resource Scenarios

A Major Qualifying Project
Submitted to the Faculty of the
WORCESTER POLYTECHNIC INSTITUTE
In Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science

Authors:
Marc Capobianco
Matthew Reynolds
Charles Phelan
Krish Shah-Nathwani
Duong Luong

Faculty Advisor:
Kyumin Lee

With Guidance From:
Muralidharan Kumaravel

Abstract:

With the recent innovation in Large Language Models, the world has been taken by storm by the vast implications and applications of including these effective new creations into our everyday lives. However, as with most things, these new wonderful leaps forward in technology are being perverted with malicious intent. Models that can effectively replicate human speech have been used to plagiarize texts, spread false information, and displace workers from their careers. In order to use these models in a beneficial way to society it's extremely important to have detection methods in place to detect non-human generated content. However, as these models are becoming ever more complex, simple solutions that have worked for previous models are rapidly becoming obsolete. So in this project, we explore the effectiveness of BERT and RoBERTa-based machine generated text detection in a supervised setting. We created several models for both BERT and RoBERTa, trained with 5%, 10%, 15%, 20%, and 100% of the dataset. We conducted these experiments with frozen and unfrozen parameter variations. We found that frozen variations of BERT outperformed frozen RoBERTa when trained on a more limited dataset, as opposed to when found that the unfrozen variation of RoBERTa outperformed unfrozen BERT when trained on the same limited data. With some final analysis, we found RoBERTa outperformed BERT in both the frozen and unfrozen variations when trained on the entire dataset.

Table of Contents:

1. Introduction:

Over the past year, the prevalence of ChatGPT in our culture has been undeniable. As artificial intelligence continues to advance, large leaps in technology often go unnoticed by the general public. However, the cultural impact of ChatGPT stands out, highlighting the perceived value attributed to new, more powerful LLMs (Large Language Models). A key factor in its prevalence is the LLM's ability to simulate human communication styles. ChatGPT can understand and respond to queries, provide information, and engage in dynamic discussions, which has led many laymen to gain a fascination with the complexity of its design. Highly technical videos about how LLMs are made, videos which would have gotten much less attention in years past, are getting millions of views. As time continues, it seems as though more of our lives have seen influence through LLMs.

With the high effectiveness of these models, many possibilities open up to what their uses could be. There are already many academics and workers who will go to an LLM for inspiration on a line of code or critique on a written piece. However, with any tool, there must be a focus on how to prevent these models from being misused.

Models such as ChatGPT have been misused in a variety of ways, but are most commonly misused for spreading misinformation and plagiarizing texts. Due to their lack of understanding of the concept of truth, these models can inadvertently report inaccurate statements as factual information, leading to significant consequences down the road. Another common misuse of LLMs is the generation of written content by rearranging information sourced from platforms like Wikipedia, which raises many questions about LLMs' relationship to intellectual property rights. What's clear is that plagiarism, intentional or unintentional, undermines the integrity of creative and scholarly works.

In our project, we repurposed the BERT and RoBERTa models to detect if text has been generated by Chat GPT. During the course of our research, we discovered and were inspired by a large variety of techniques used by others. With what we have discovered and produced over the course of this project, we hope that further steps can be taken to keep the use of LLMs as secure as possible.

2.Related Works:

Before embarking on this project, our team read many papers, and 2 of them proved to be imperative to our success. Those were: “GTLR: Statistical Detection and Visualization of Generated Text” [1], and “Can AI-Generated Text be Reliably Detected?”[2] From these papers, we were able to glean essential information that lent itself to our understanding of the project.

An AI text detector is an essential tool in discerning between text generated by artificial intelligence (AI) systems and human-written content. In this section of the paper, we focus on the GTLR (Generative Textual Likelihood Ratio) approach, which aims to create an effective AI text detector. [1] The GTLR approach utilizes likelihood ratios to quantitatively assess the evidence supporting the hypothesis that a given text is generated by AI. In addition to likelihood ratios, the authors introduce the concept of LLRP (Log Likelihood Ratio Plot) as a means of visually representing these likelihood ratios, in an effort to make them easier to interpret. To calculate likelihood ratios for new text samples, the GTLR approach employs two models: a generative model trained on AI-generated text and a discriminative model trained on human-written text. By utilizing these two models, the GTLR approach aims to provide a robust framework for AI text detection.

As mentioned previously, the likelihood ratio-based approach forms a fundamental aspect of AI text detection. By using two models, generative and descriptive, and utilizing tools such as LLRP, this approach allows for the quantification of the likelihood of text being AI-generated. LLRP also allows for deeper analysis and interpretation.

Various methods have been developed to detect AI-generated text, encompassing binary classification models, zero-shot classifiers [3], neural network-based detectors such as RoBERTa from Open AI [4], and soft watermarking [5]. Binary classification models have been specifically designed to distinguish between AI-generated and human-written text by training on labeled datasets containing examples from both categories. However, their effectiveness heavily relies on the quality and representativeness of the training data, and they may struggle with sophisticated AI-generated text that closely mimics human writing. Zero-shot classifiers, on the other hand, are trained on extensive collections of human-written text, allowing them to generalize their understanding of human language and identify linguistic patterns as well as semantic cues indicative of human-written text, Nonetheless, they face challenges in accurately detecting AI-generated text that closely resembles human writing due to their lack of explicit training on AI-generated examples.

Neural network-based detectors utilize advanced architectures to identify features and patterns of a characteristic of AI-generated text. These detectors leverage the power of neural networks to learn complex representations and make accurate predictions. They can capture intricate relationships and nuances in the text, enhancing the detection capabilities compared to traditional approaches. Additionally, to make plagiarism easier to catch, AI-generated text has explored soft watermarking techniques. This involves the embedding of watermark signatures within the AI-written text. These signatures serve as subtle indicators of AI generation and can be used to identify the origin of the text. However, like any watermarking technique, they may be susceptible to removal or alteration attempts, and their effectiveness depends on the robustness of the watermarking scheme.

Current AI text detectors are vulnerable to paraphrasing attacks, where AI-generated text is rephrased or rewritten to alter its linguistic characteristics. Paraphrasing attacks pose a significant challenge to the effectiveness of detection methods since they aim to evade detection by altering the text's surface features while preserving its underlying meaning. Such attacks can undermine the accuracy of AI text detectors as they exploit the limitations in capturing nuanced changes introduced by paraphrasing techniques. The ability to deceive detectors through paraphrasing attacks highlights the need for more robust and nuanced detection mechanisms that can accurately identify AI-generated text regardless of its phrasing or superficial alterations.

In the aforementioned paper "Can AI-Generated Text be Reliably Detected?" [2], a hypothesis gaining prominence in the field is that AI text and human text will eventually become indistinguishable. As AI models continue to advance and generate more sophisticated text, there is a growing concern that distinguishing between AI-generated and human-written content will become increasingly challenging. If this hypothesis proves true, it poses significant implications for the development of AI text detectors. It implies that detectors will need to adapt and evolve to discern minute differences in linguistic patterns, style, and context that may still exist between AI-generated and human-written text. This challenges researchers and developers to continuously improve the accuracy and robustness of AI text detectors in the face of an ever-narrowing gap between AI and human writing.

Spoofing attacks present a unique challenge in AI text detection, involving instances where humans intentionally write text that imitates the style and characteristics of AI-generated content. In these attacks, humans intentionally mimic the writing patterns, vocabulary, and syntax commonly associated with AI models. The goal is to confuse or mislead AI text detectors, making it difficult to differentiate between AI-generated and human-written text. Detecting spoofing attacks requires detectors to have a deep understanding of the nuanced differences in writing styles, semantic coherence, and contextual cues that exist between AI-generated and human-written text. The challenges associated with spoofing attacks necessitate the development of more sophisticated detection techniques that can effectively discern these subtle differences and mitigate the risk of false positives or negatives.

The use of PEGASUS-based paraphrasers [6] adds another layer of complexity to AI text detection. PEGASUS models are known for their ability to generate high-quality paraphrases, which can be utilized to remove the watermark signature or other telltale signs of AI-generated text. These paraphrasers can modify the AI-generated text in a way that it retains the same underlying meaning but eliminates the distinctive features used by detectors for identification. This poses a challenge for detection methods that rely on watermarking or specific markers to identify AI-generated text. The effectiveness of detection methods may be compromised as PEGASUS-based paraphrasers continue to improve, emphasizing the need for more comprehensive and resilient detection techniques that can withstand the removal or alteration of AI-generated text's distinctive attributes.

Zero-shot detectors, while promising in their ability to generalize knowledge from human-written text, are not without limitations and vulnerabilities. One key concern lies in their reliance on training solely on human-written data, making them inherently biased towards human language characteristics. This bias can hinder their ability to accurately identify AI-generated text that closely mimics human writing patterns, syntax, and vocabulary. Additionally, zero-shot detectors may

struggle to adapt to evolving AI models and novel AI generation techniques, potentially leading to reduced detection performance. Developing zero-shot detectors that can effectively handle the intricacies of AI-generated text and remain adaptable to emerging AI advancements is crucial to enhance the reliability and robustness of AI text detection approaches.

In summary, vulnerabilities and challenges in AI text detection include susceptibility to paraphrasing attacks and the potential impact on detection methods. The indistinguishability of AI and human text raises concerns about the future effectiveness of AI text detectors, emphasizing the need for continuous improvement and innovation in detection techniques. Spoofing attacks further complicate the detection landscape by introducing intentional human deception that mimics AI-generated content. Detecting such attacks required detectors to be equipped with advanced capabilities to discern subtle differences in writing styles and contextual cues.

The emergence of PEGASUS-based paraphrasers presents a new challenge as they can effectively remove the watermark signature or other identifiable features of AI-generated text. This necessitates the development of more comprehensive and resilient detection techniques that can adapt to evolving paraphrasing strategies. Furthermore, concerns regarding zero-shot detectors stem from their inherent biases toward human language, limiting their ability to accurately identify AI-generated text that closely resembles human writing. Overcoming these challenges requires the development of detection methods that can effectively handle the nuances of AI-generated text and remain adaptable to the evolving landscape of AI advancements.

In conclusion, addressing the vulnerabilities and challenges in AI text detection requires continuous research and development. Future detection methods should consider the susceptibility to paraphrasing attacks, the potential indistinguishability of AI and human text, the need to detect spoofing attacks, the impact of PEGASUS-based paraphrasers on detection effectiveness, and the limitation of zero-shot detectors. By addressing these challenges, we can enhance the reliability and robustness of AI text detectors, enabling us to effectively distinguish between AI-generated and human-written content in an evolving landscape of AI advancements.

To summarize everything discussed in this section, the AI text detection field still needs lots of research and development into the current numerous vulnerabilities and challenges it faces. End users paraphrasing AI generated text still quashes the effectiveness of all current detection methods because paraphrasing alters the original text while still preserving its overall meaning. The hypothesis that the differences between AI and human generated text becoming indistinguishable from one another raises concerns about the effectiveness of detectors in the future. The development of the generators mandates the development of the detectors, they grow hand in hand with one another. Spoofing attacks as well complicate the detectors, meaning the models that are being developed need to be made ever more advanced to discern the very subtle differences in the writing styles and the use of contextual cues. The emergence of PEGASUS-based paraphrases adds complexity as it removes AI generated identifying features from the text. Additionally, there are inherent biases in zero-shot detectors that limit their overall ability to identify AI-generated text accurately. Which emphasizes the need for new and improved detection methods. By addressing and hopefully remedying these challenges we can enhance the overall reliability and robustness of the AI generated text detectors. This would enable effective distinction between AI and human written text in a developing AI landscape.

3.Methodology:

3.1 The Dataset:

We collected our data from a public dataset known as “HC3 Corpus” [7]. This is a corpus of questions with accompanying expert human answers and answers from ChatGPT. The group asked 24322 questions and recorded the answers to those questions. The distribution of answers between ChatGPT and humans is visualized in figure 1. The questions and responses were in two different languages, English and Chinese, so we decided to discard the Chinese and keep the English for our analysis. The answers generated by experts came through two main sources, question-answering datasets that are open for anyone to access and also from Wiki text. The dataset has several subcategories of questions asked. In total, there were 3933 questions with a financial focus, with 3933 human answers and 4503 AI answers. There were 1248 with a medical focus, with 1248 human answers and 1337 AI answers. There were 1187 Open QA questions, with 1187 human answers and 3561 AI answers. There were 17112 questions from Reddit’s “Explain Like I’m Five” subreddit, with 51336 human answers and 16660 AI answers. And there were 842 questions from Wikis, with 842 Human and 842 AI answers. The full distribution of questions and answers can be viewed in Table 1. The answers generated by ChatGPT were manually imputed, with each question asked on a new, blank thread to avoid influence from prior chat history. We chose this dataset so that our effort would be focused on the experiments rather than the data collection process itself.

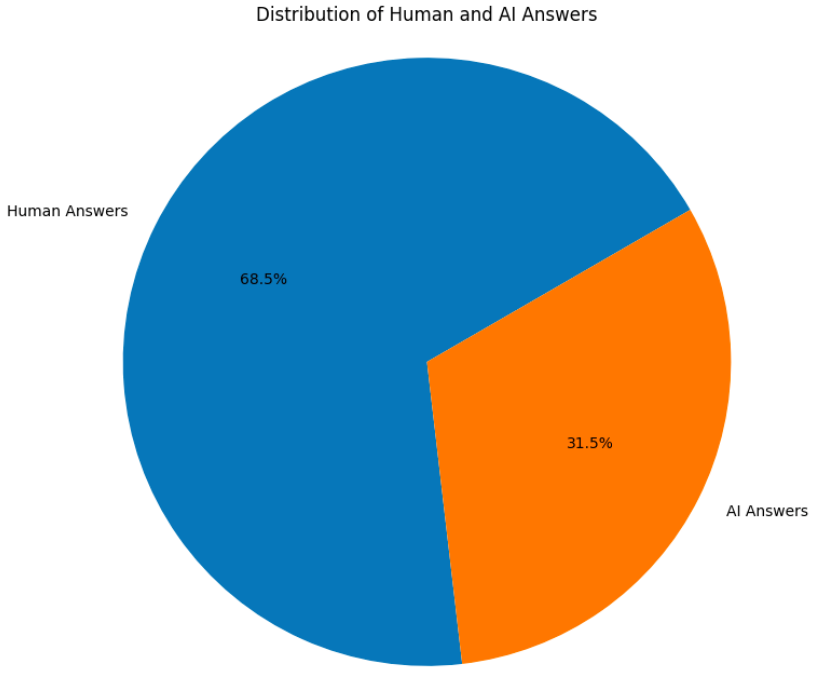

Figure I: Visualization of Distribution of Total Human and AI Answers

Category	# of Questions	Human Answers	AI Answers
Financial Focus	3933	3933	4503
Medical Focus	1248	1248	1337
Open QA	1187	1187	3561
Reddit's "Explain Like I'm Five" Subreddit	17112	51336	16660
Questions from Wikis	842	842	842
Total	24322	58546	26903

Table I: Summary of "HC3 Corpus" [7] Sources and Responses

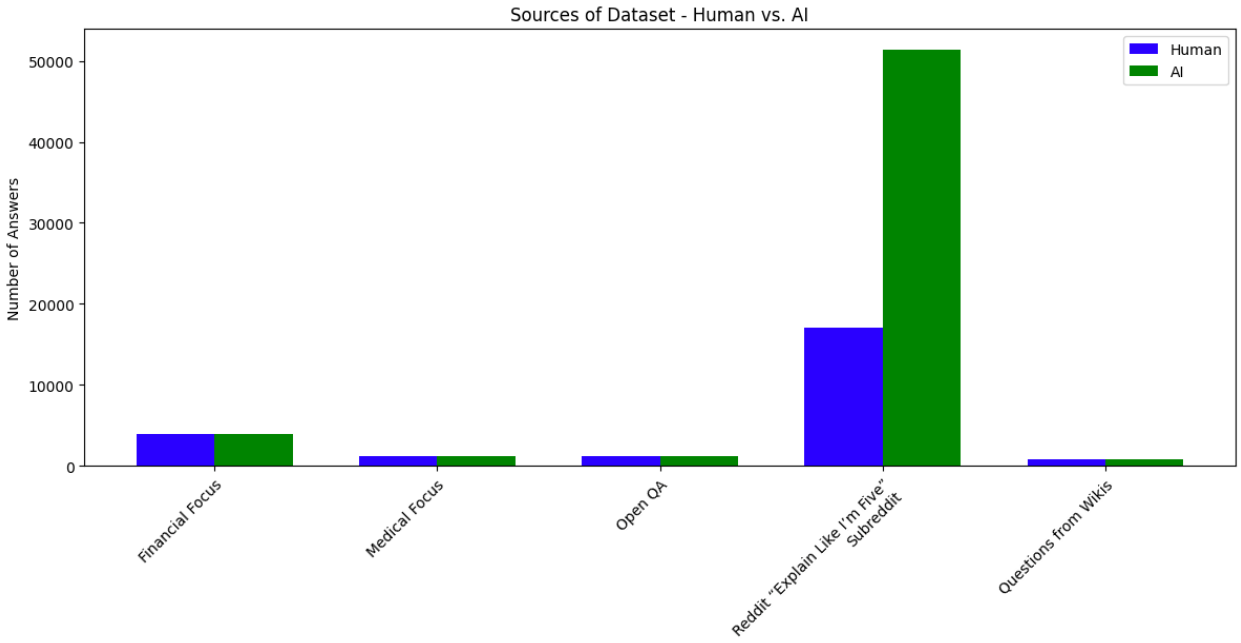

Figure II: Visualization of the Sources of Human and AI Answers

Figure II shows the large disparity in the sources of the content. The content derived from Reddit's "Explain Like I'm Five" subreddit comprises 87.7% of the total Human generated answers. And 61.9% of the total AI generated answers. Reddit is an online forum for users to post and be responded to in the form of comments by other users. Anyone with a registered email address can create an account and begin posting and commenting on other users' posts. There is no authentication or verification beyond registration, meaning there most likely are a significant

amount of uninformed users who are commenting on things as if they were experts. Reddit's "Explain Like I'm Five" subreddit is a subcategory, referred to as a subreddit, from the overall platform. In it, users post questions and other users explain to the original poster the concept in simple terms. This means that most of the content derived from reddit in our data is a user explaining a concept in simple terms to another user, as if they were five years old.

3.2 The Models Used and Their Architecture:

BERT, short for Bidirectional Encoder Representations from Transformers, is the improvement upon traditional NLP models. Instead of the recurrent neural networks and convolutional neural networks which struggle with understanding contextual information due to its sequential processing, BERT is based on a Transformer architecture which means it's able to circumnavigate these issues with bidirectional pre-training. BERT's most significant improvement is its ability to understand contextualized word representations based on considering both its left and right context in a sentence. Another thing that separates BERT from its predecessors is that it pre-trains on a corpus of text using two unsupervised tasks, masked language modeling and next sentence prediction.

In its masked language modeling task, random words in a sentence are masked, and the BERT model is tasked with having to predict what the original word is based on its context in the sentence. The next sentence prediction task evaluates BERT's ability to comprehend how two consecutive sentences are related. This means the model predicts whether or not two sentences appear consecutively in the training data or not. Due to these tasks the BERT model develops a deep understanding of the nuances of language, specifically it is capable of understanding intricate sentence structure, it can resolve ambiguities in words, and can contextualize the meanings of words more accurately.

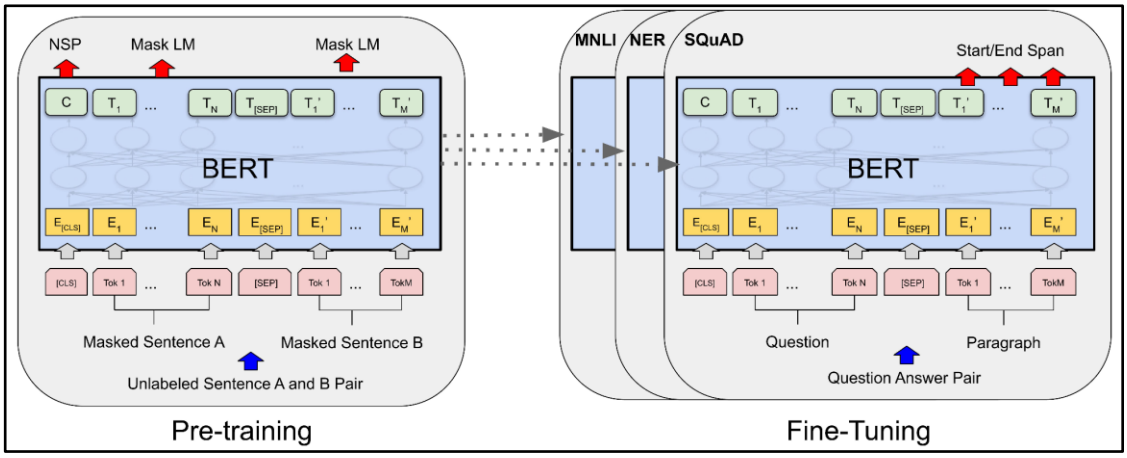

Figure III: The Model Architecture of BERT [9]

Freezing the weights in BERT refers to when the parameters of a pre-trained BERT model are fixed during fine-tuning. This method is preferable when there's a smaller dataset due to it helping to prevent overfitting the dataset. During the fine-tuning process, the pre-trained BERT

model's parameters are kept fixed throughout. Instead of updating all the parameters during fine tuning, it only updates the weights in the specific task layer that is added on top of BERT are modified. Freezing the majority of the weights of BERT's parameters makes the fine-tuning process significantly more stable, less data consuming, and less computationally intensive.

In comparison, RoBERTa, an acronym for "A Robustly Optimized BERT Approach," represents a significant advancement in the field of natural language processing (NLP). Introduced by researchers at Facebook AI in 2019, RoBERTa builds upon the foundation laid by the highly influential BERT model. While BERT demonstrated remarkable success in various NLP tasks, RoBERTa addresses some of its limitations and achieves even higher performance through a series of improvements.

At its core, RoBERTa leverages the Transformer architecture, which has proven to be highly effective in modeling and understanding sequential data, such as language. Similar to BERT, RoBERTa follows a pre-training and fine-tuning paradigm. During pre-training, the model is exposed to massive amounts of unlabeled text data, learning to predict masked words within a given context. This process allows RoBERTa to capture the syntactic and semantic properties of language.

What distinguishes RoBERTa from its predecessor is the approach taken in training. The researchers behind RoBERTa recognized that BERT's training methodology had some room for improvement. They conducted a series of experiments to identify the impact of various hyperparameters and training strategies on the model's performance. The result was a set of modifications that led to substantial improvements in the model's effectiveness.

One key enhancement in RoBERTa is the increase in the amount of training data. BERT was trained on a combination of English Wikipedia and the BooksCorpus dataset. In contrast, RoBERTa leverages an even larger corpus of text, incorporating additional sources such as Common Crawl, a vast web crawl containing diverse and extensive linguistic data. By training on a broader range of text, RoBERTa gains a deeper understanding of language patterns and nuances.

Furthermore, RoBERTa employs a modified training schedule that eliminates the Next Sentence Prediction (NSP) task. In BERT, the NSP task involves predicting whether two sentences appear consecutively in a document. The exclusion of this task in RoBERTa allows for more dynamic and focused training, as it encourages the model to learn better representations for individual sentences. Consequently, RoBERTa demonstrates improved performance on a range of downstream tasks.

The training of RoBERTa also involves adjusting hyperparameters such as the batch size, sequence length, and learning rate schedule. By meticulously tuning these parameters, the researchers fine-tuned the model's ability to capture context and generalize across various NLP tasks. These optimizations contributed to the model's robustness and significantly enhanced its performance on benchmarks.

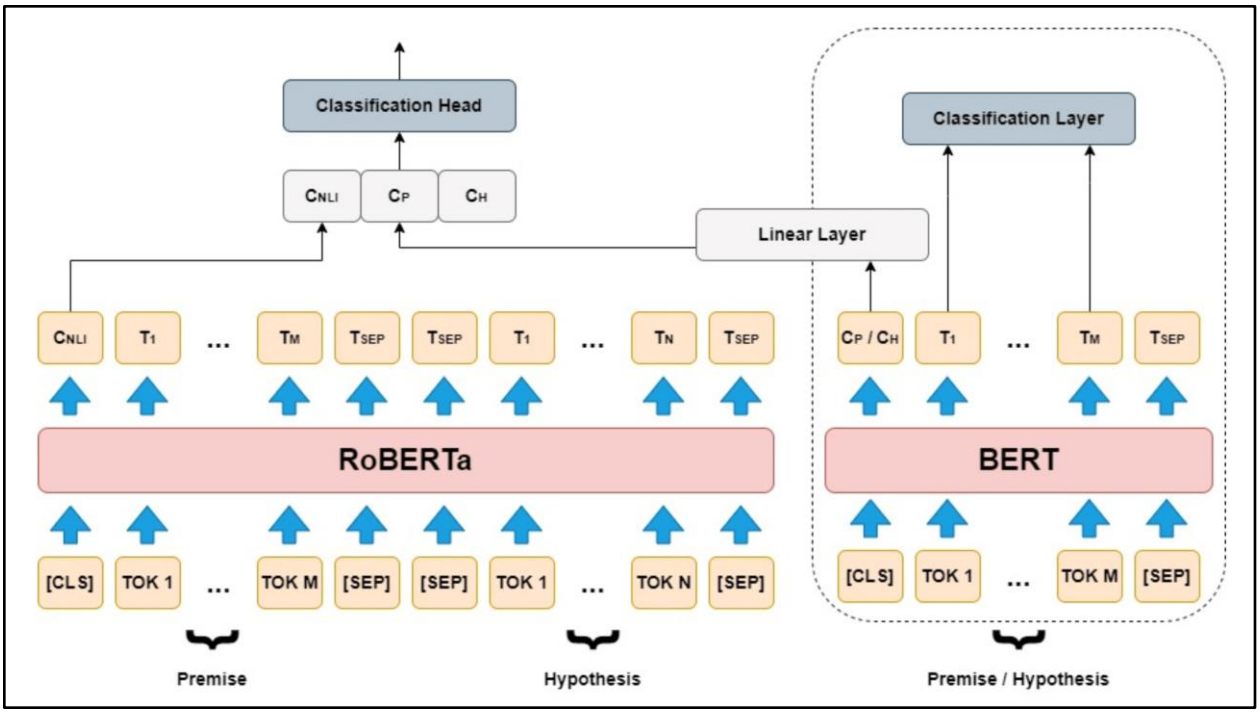

Figure IV: The Model Architecture of RoBERTa [8]

In summary, RoBERTa is an advanced language model that builds upon the success of BERT. Through modifications in training methodology, including an increase in training data, the exclusion of the NSP task, and fine-tuning of hyperparameters, RoBERTa achieves state-of-the-art performance in a wide range of NLP tasks. Its ability to understand context, capture intricate language patterns, and generate accurate predictions makes it a valuable tool in natural language understanding and generation applications.

3.3 Other Necessary Knowledge:

3.3.1 Perplexity:

In the context of NLPs, perplexity is a measure of how predictable any given word is, if you know all of the words that have come beforehand. For example, think of the phrase “Once upon a time…” If the language model knows that a sentence starts with “Once”, that model is somewhat likely to predict that “upon” would be the next word, but it could also predict “there” as the next word. In that case, both of those words would have a high probability of being next, and that section of text (“Once upon…” or “Once there…”) would have a low perplexity. In contrast, there would be a lower probability of “fox” being directly after “Once”. Therefore, “Once fox…” would have a higher perplexity. The probability of each word given the previous words could then be multiplied to have a perplexity score, which could then be normalized using some geometric mean, depending on the analysis.

3.3.2 Fine Tuning:

The last important concept to understand before we explain our model is the general process of fine tuning a pre-trained language classification model. First, it is important to organize your data into a few sets: a training set, a validation set, and a testing set. Then you should choose whether or not to add layers of new classifiers to your model based upon any factors that you can extract. These added layers would then be used to fine tune the pre-trained model by training the added layers with your training data set. If there will not be layers added, then you may fine tune the model end-to-end by training the entire model with your training data set. Next, you could fine tune your hyperparameters which would be different for any given model. However, hyperparameters such as learning rate and batch size are very common. Next, you may train and partially test your model on a secondary data set or a validation set. The training here should be done after the previous steps in order to test your models strength, so that you may modify any problems with the hyperparameters, extracted features, ect. before your model is properly evaluated using the test set. The final step is to finally run a test set through your model to view the model's effectiveness.

3.4 Direct Methodology Implemented:

3.4.1 BERT:

We used the BERT machine learning model on a classification problem to detect human and machine-generated text. Compared to previous language models based on unidirectional architecture, BERT (Bidirectional Encoder Representations from Transformers) utilizes bidirectional architecture, thus allowing for tokens to incorporate context from both the previous tokens and consequent tokens. The pre-training methodology employs "masked-language model" and "next sentence prediction" tasks to process language in the pre-training dataset.

In the Masked Language Model input words are converted to tokens and a percentage of those tokens are randomly masked. The vectors of the maxed tokens are input to a SoftMax function to determine the probability distribution for the predicted tokens. For this task, the token with the highest probability is taken as the most suitable token.

The Next Sentence Prediction task uses sets of sentences to understand the relationships between sentences in a monolingual corpus. The task is carried out by having sets of sentences consisting of sentences A and B. B consists of labeled sentences that either follow sentence A or are a random sentence. The sentence sets are analyzed by a neural network to improve natural language inference and question and answering tasks capabilities.

The pre-training data used for our Bert model was trained on BooksCorpus and English Wikipedia. The pre-trained parameters of BERT are fine-tuned using the human-chatGPT comparison corpus HC3 dataset. The dataset consists of three labels: Question, Human Answer, ChatGPT Answer. The training, validation and testing sets a split from the data at a ratio of 40%,

50%, 10% respectively. A tokenizer is prepared from Hugging Face to map strings to token space and is used on the Bert-base-uncased model. Training, validation, and test tokens are given a max length of 128 and padded to reach the limit of tokens for BERT input before performance decrease (512 tokens). The tokens are then converted to tensor multi-dimensional arrays. We used a batch size of 6 and fine-tuned for 3 epochs over the HC3 corpus. The BERT model from Hugging Face uses 12 layers of transformers block with a hidden size of 768, representing the number of features used to embed a word. The best fine-tuning learning rate used was 1e-5.

3.4.2 RoBERTa:

This section explains RoBERTa's pre-training methodology and steps that we have applied to fine-tune this language model for our classification task.

Introduced by Facebook AI in 2019, the RoBERTa model represents a major improvement over its predecessor - BERT with a series of modifications. While both language models share the same Transformer architecture and a similar pre-training/fine-tuning paradigm, RoBERTa is more robust and achieves greater performance thanks to its larger training corpus and BPE vocabulary size, the removal of the Next Sentence Prediction (NSP) task, the implementation of dynamic masking, and training with large mini-batches.

Compared to BERT, RoBERTa was pre-trained on a significantly larger corpus, incorporating BOOKCORPUS plus English WIKIPEDIA (16GB), CC-NEWS (76GB), OPENWEBTEXT (38GB), and STORIES (31GB) [4]. Being pre-trained on a more extensive corpus for an extended period helps RoBERTa gain much deeper insights into human language and its pattern.

One key enhancement that helps RoBERTa outperform BERT is the elimination of the Next Sentence Prediction (NSP) task from the training objectives. To keep our paper from being redundant, an explanation of why this modification increases RoBERTa's end-task performance can be found in the model architecture section.

Another key enhancement of RoBERTa is the implementation of dynamic masking. In static masking, as seen in BERT, a fixed set of tokens are randomly masked during pre-training [4] and the model's task is to predict these masked tokens. However, in dynamic masking, a set of tokens are randomly masked in each iteration of pre-training, meaning that there will be a different set of masked tokens in each pre-training epoch. This strategy improves RoBERTa's generalization by preventing the model from concentrating on specific patterns that may develop as a result of static masking.

Furthermore, the RoBERTa model was pre–trained with large mini-batches to achieve better end-task performance. Researchers who created the RoBERTa model has found during their experiments that "training with large batches improves perplexity for the masked language modeling objective, as well as end-task accuracy."[4] Mini-batch training is a common approach in deep learning where the training weights are updated several times during one single epoch.

The process of fine-tuning the RoBERTa model begins with converting the downloaded HC3 dataset (JSONL file) to the Pandas DataFrame. The newly created DataFrame contains four columns, including "Question", "Text", "LabelName", and "Label". The "Question" column is where we can find all the question prompts, while the "Text" and "LabelName" columns consist of all the input texts and their corresponding labels (Human Answer or ChatGPT Answer). The last column "Label" displays

“0” or “1” depending on whether the input text is human-written or machine-generated. Next, we splitted the dataset into the training, validation, and testing sets with a ratio of 40%, 50%, 10% respectively. We then tokenized the input text data by using a RoBERTa tokenizer from Hugging Face. The model that we used was Roberta-base. The tokenized sequences were assigned a max length of 128 and they were truncated or padded to match the RoBERTa model’s input requirements. The tokens were then converted to tensors. Similar to BERT, RoBERTa has 12 layers of transformers block with a hidden size of 768. For the optimizer and loss function, we used the Adam optimizer and Cross-Entropy loss function respectively. During the training process, we tuned hyperparameters such as the learning rate, the number of batch size, and the number of training epochs to achieve the best performance. After numerous trials, we observed that the optimal setting to achieve best performance is to have a batch size of 6, learning rate at 1e-5, and fine-tuned for 5 epochs over the HC3 corpus dataset.

4. Experiments and Results:

4.1 Direct Explanation:

We trained 10 different variations of our BERT model. We split them into two sections, 5 with frozen parameters and 5 with unfrozen parameters. We then split and trained one of each with 5%, 10%, 15%, 20%, and 100% of the data and recorded the results in our table.

To explain the results of our experiments, we used several key metrics. We used Precision, Recall, F1 score, and Support to determine the success as these are commonly used metrics to evaluate classification models. Precision is a metric that is the number of positive predictions that belong in the positive class. Recall is a metric that is the number of positive class predictions made of all positive examples in the overall dataset. The F1 score is a balance of Precision and Recall. The F1 score is a sigmoid, meaning it falls between 0 and 1. The closer to 1 the value is the greater the overall accuracy of the model is.

Model	Weight Type	Testing Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)
BERT	Frozen	0.90	0.88	0.91	0.89
	Unfrozen	0.99	0.99	1.00	0.99
RoBERTa	Frozen	0.95	0.93	0.96	0.94
	Unfrozen	1.00	1.00	1.00	1.00

Table II: BERT and RoBERTa Models Trained with 100% of the Data.

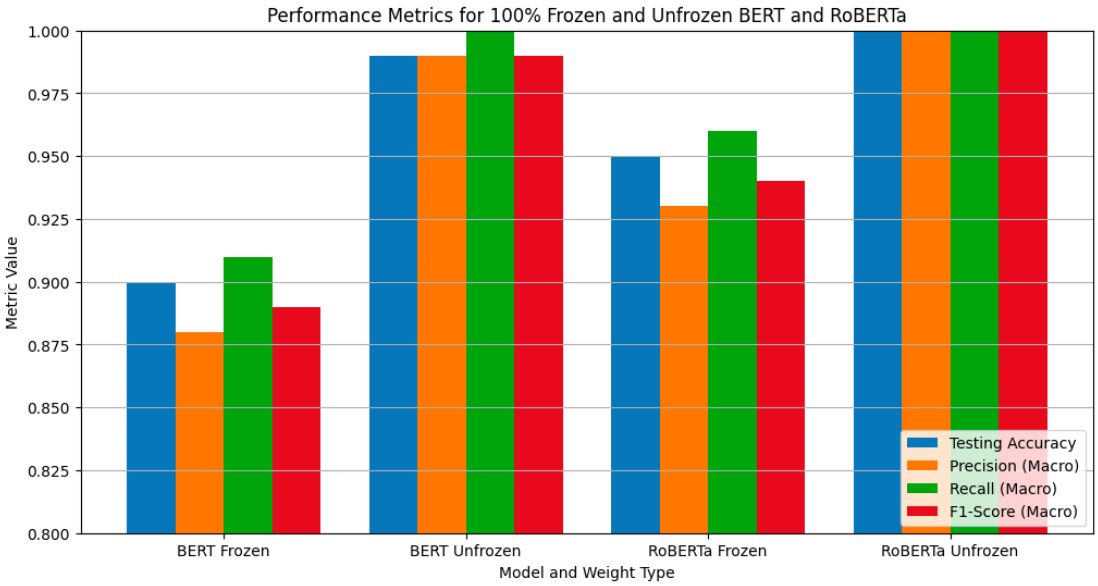

Figure V: Performance Metrics for Frozen and Unfrozen BERT and RoBERTa Trained with 100% of the Data.

Figure V shows the differences between each of the BERT and RoBERTa models across all metrics. It shows that the unfrozen models marginally outperformed their frozen counterparts. Interestingly, both of the frozen models show the same pattern in their results.

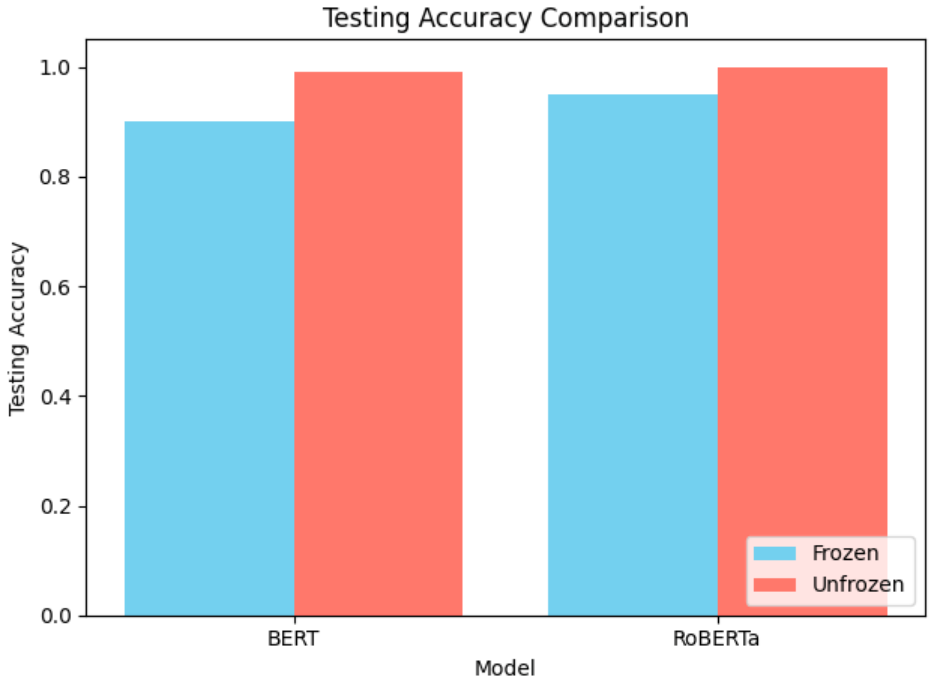

Figure VI: Comparison of Testing Accuracy for Both the BERT and RoBERTa Models in Frozen and Unfrozen Variations

Figure VI compares the Testing Accuracy metric between the four models. The frozen models are marginally different, with the RoBERTa being slightly more accurate than the BERT model. And the unfrozen graphs are nearly identical, with RoBERTa being 1% more accurate.

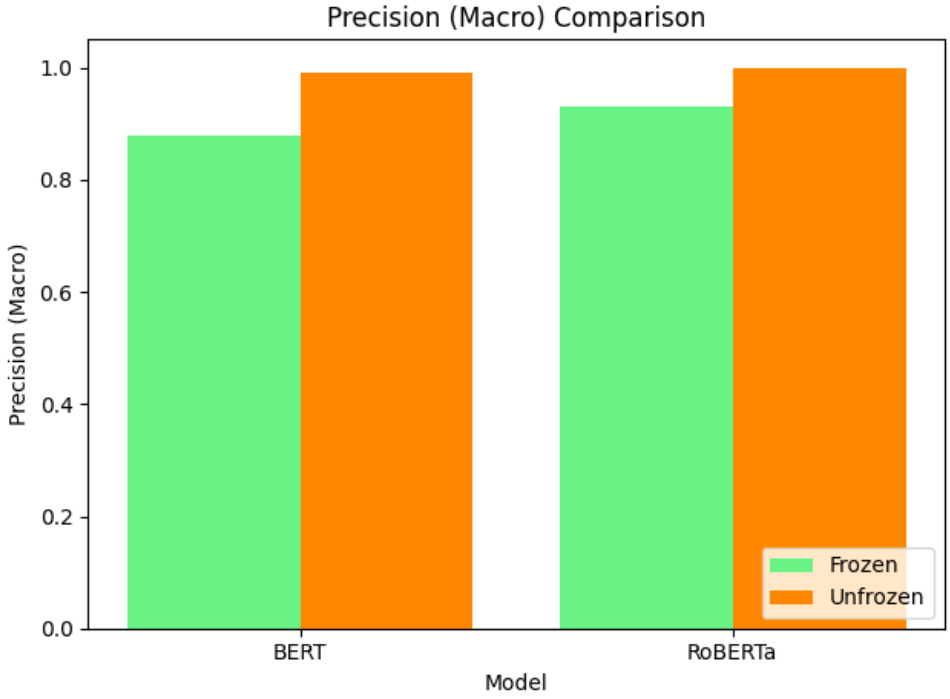

Figure VII: Comparison of Precision for Both the BERT and RoBERTa Models in Frozen and Unfrozen Variations

Figure VII compares the Precision metric for the four models. It again shows that the unfrozen variation of these models outperforms the frozen models and that the RoBERTa scores the highest of the two. This figure, however, shows the largest disparity between the frozen and unfrozen variations. The delta of the two BERT variations is 11%.

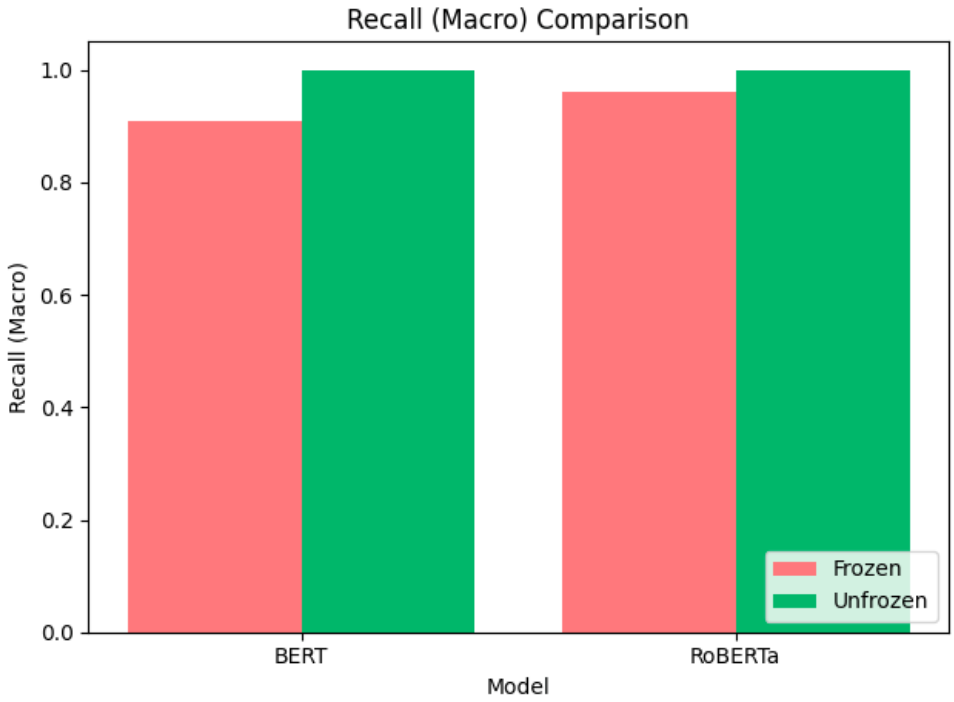

Figure VIII: Comparison of Recall for Both the BERT and RoBERTa Models in Frozen and Unfrozen Variations

Figure VIII compares the Recall metric for the four variations. This shows as well that the RoBERTa model outperforms the BERT models across the frozen and unfrozen variations.

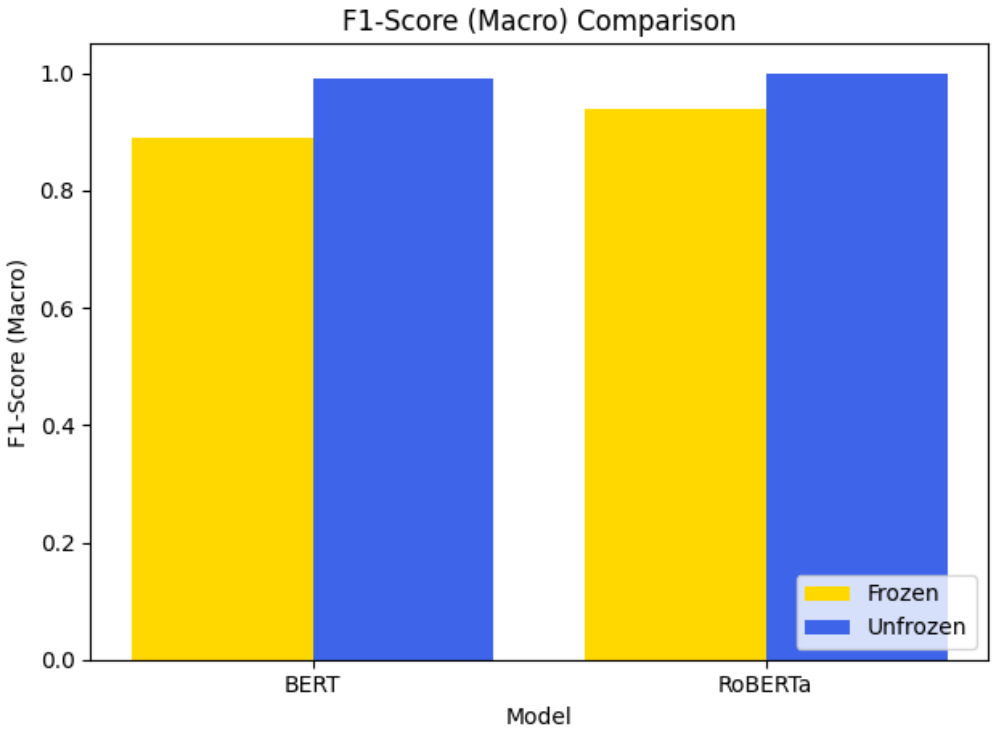

Figure IX: Comparison of F1-Score for Both the BERT and RoBERTa Models in Frozen and Unfrozen Variations

Figure IX compares the F1-Score metric for the four variations. Again, this figure shows that the RoBERTa model outperforms the BERT models by a marginal amount in the frozen and unfrozen variations. This figure also shows the second largest difference between the frozen and unfrozen variations. The delta of the BERT models is 10%.

Model Variation	Data Percentage	Testing Accuracy	Precision (macro)	Recall (macro)	F1-Score (m
BERT Frozen	5%	0.81	0.80	0.75	0.76
BERT Frozen	10%	0.84	0.81	0.82	0.81
BERT Frozen	15%	0.84	0.82	0.85	0.83
BERT Frozen	20%	0.86	0.83	0.86	0.84
BERT Unfrozen	5%	0.95	0.93	0.96	0.94
BERT Unfrozen	10%	0.96	0.94	0.97	0.95
BERT Unfrozen	15%	0.94	0.92	0.96	0.93
BERT Unfrozen	20%	0.96	0.94	0.97	0.95
RoBERTa Frozen	5%	0.69	0.34	0.50	0.41
RoBERTa Frozen	10%	0.69	0.34	0.50	0.41
RoBERTa Frozen	15%	0.69	0.34	0.50	0.41
RoBERTa Frozen	20%	0.70	0.79	0.52	0.45
RoBERTa Unfrozen	5%	0.98	0.97	0.99	0.98
RoBERTa Unfrozen	10%	0.99	0.98	0.99	0.99
RoBERTa Unfrozen	15%	0.99	0.99	1.00	0.99
RoBERTa Unfrozen	20%	0.95	0.93	0.96	0.94

Table III: BERT and RoBERTa Models' Performance Metrics Trained with Varying Amounts of Data

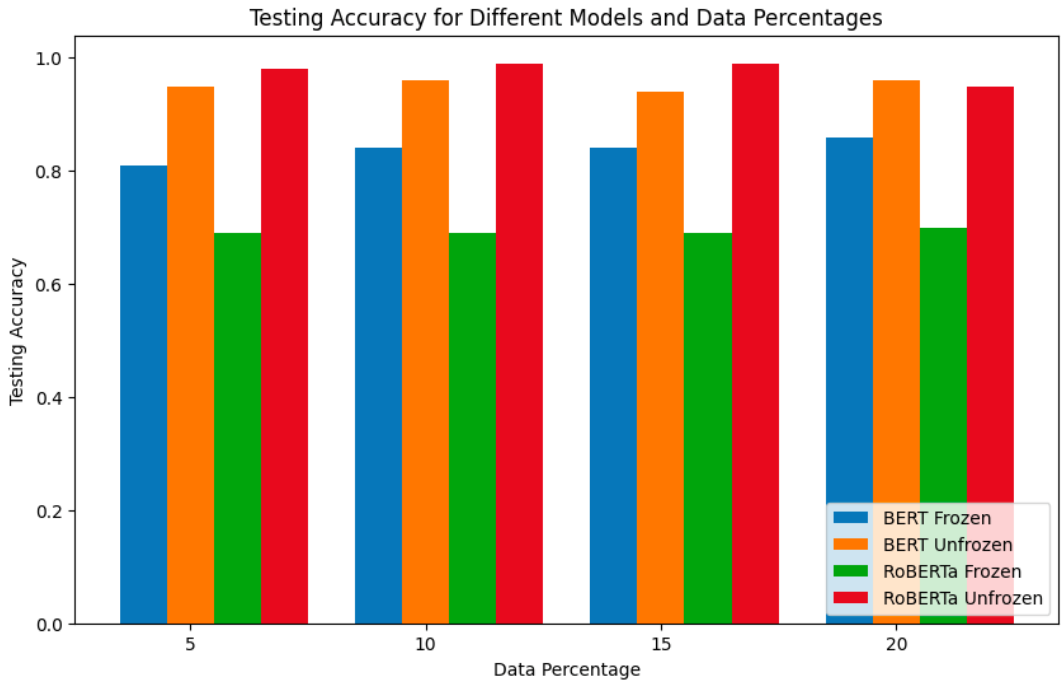

Figure X: Comparison of Testing Accuracy for Both the BERT and RoBERTa Models in Frozen and Unfrozen Increasing at Various Data Percentage Levels

Figure X compares the Testing Accuracy Metric across all percentages of data the four variations were trained on. Across all four variations of data the models were trained on there's very little difference between all of the models. In all the percentages there's only two instances of the models underperforming their prior data percentage; BERT unfrozen at 15% performed 2% worse than its prior, and RoBERTa unfrozen at 20% performed 4% worse than its predecessor.

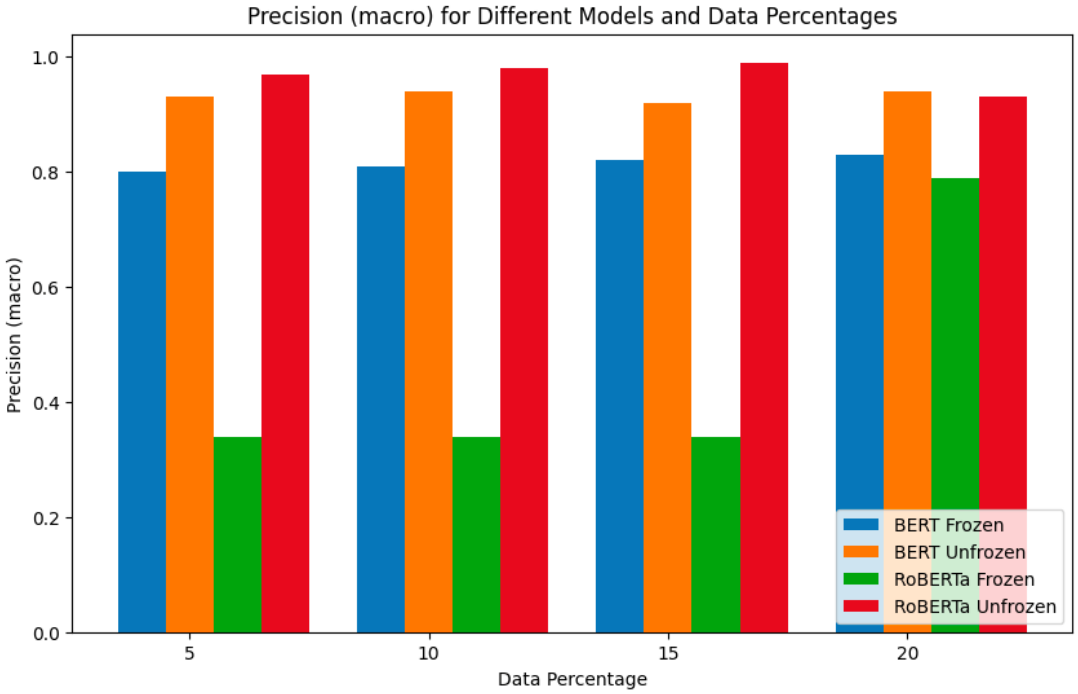

Figure XI: Comparison of Precision for Both the BERT and RoBERTa Models in Frozen and Unfrozen Increasing at Various Data Percentage Levels

Figure XI compares the Precision metric for all the data percentage variations across the four models. Similar to Figure X, the four data percentages look remarkably similar across all four of the percentages, except for RoBERTa unfrozen which has a dramatic spike at 20% data, over doubling its prior score.

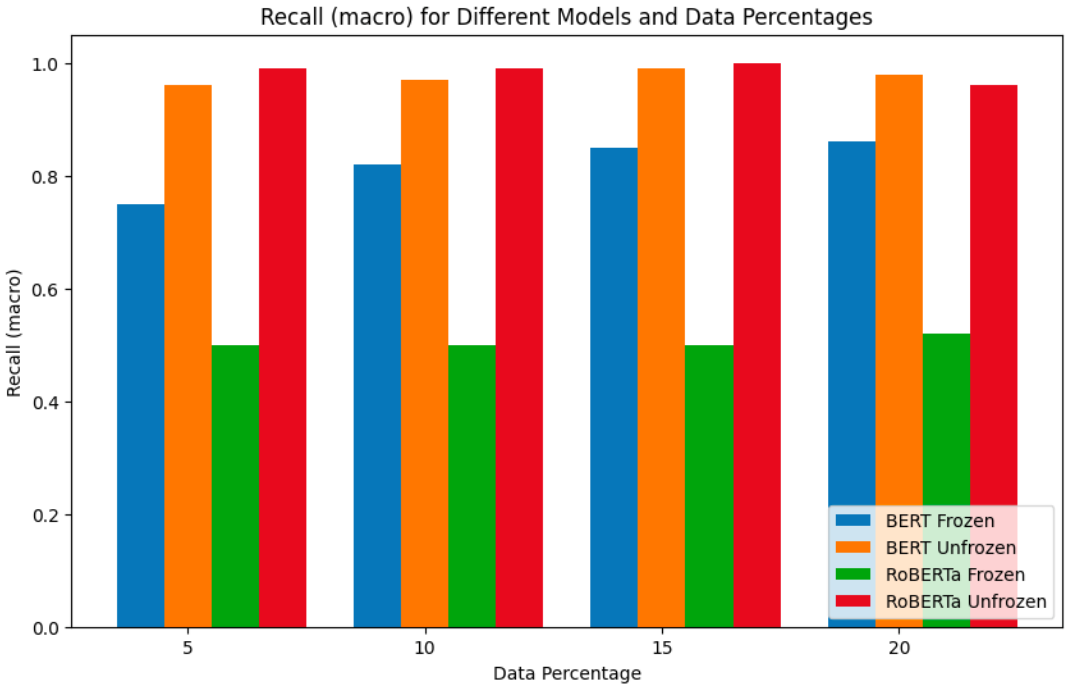

Figure XII: Comparison of Recall for Both the BERT and RoBERTa Models in Frozen and Unfrozen Increasing at Various Data Percentage Levels

Figure XII compares the Recall metric across the data percentage variations for the four models. Figure XII similarly to Figure XI and Figure X shows very little differences across the four data percentages. BERT frozen continues to improve as more data is fed into it; however, all other metrics show marginal or no growth. RoBERTa unfrozen shows an interesting loss at 20% compared to its priors, losing 4% accuracy.

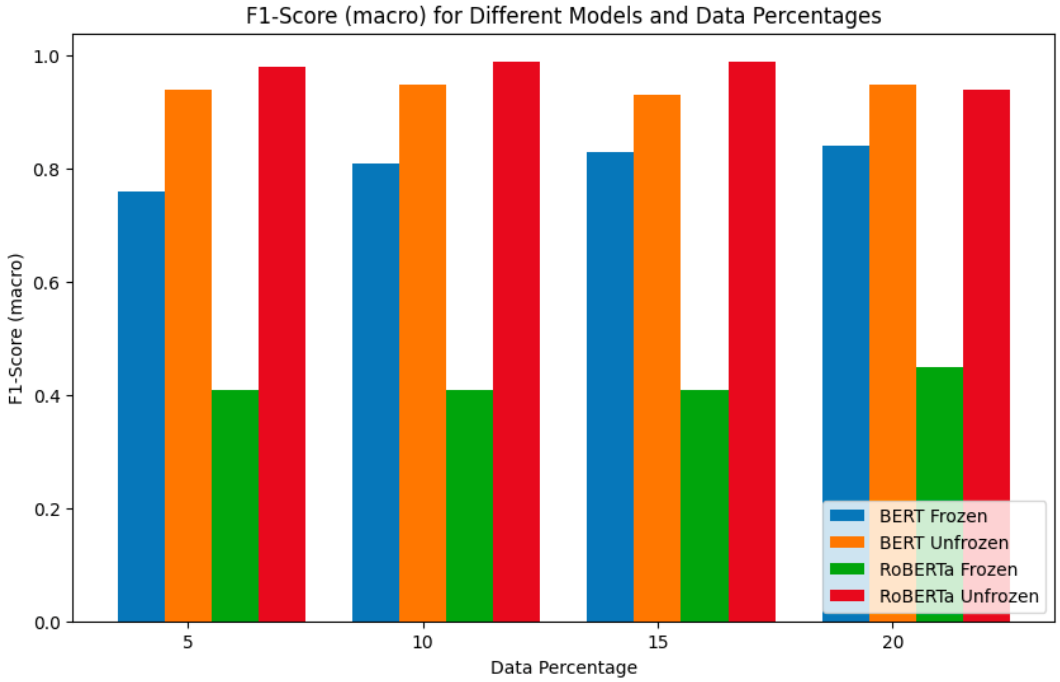

Figure XIII: Comparison of F1-Score for Both the BERT and RoBERTa Models in Frozen and Unfrozen Increasing at Various Data Percentage Levels

Figure XIII compares the F1-Score across the four data percentages for the four models. Similarly to Figure XII there are almost no differences across the four data percentages. BERT frozen similarly continues to increase across the four data percentages. And RoBERTa unfrozen shows another drop off at the 20% data percentage of 5% loss for this metric.

4.2 Perplexity and t-SNE Analysis:

The following section utilizes a perplexity distribution and t-SNE visualizations in order to further analyze the characteristics of the HC3 dataset along with comparing and contrasting the BERT and roBERTa models. These methods of analysis are used to gain deeper insight into the difference between human generated and machine generated text on a statistical and syntactic level. Additionally, the degrees of effectiveness and ways in which each model perceives and classifies the dataset are investigated.

4.2.1 Perplexity Distribution:

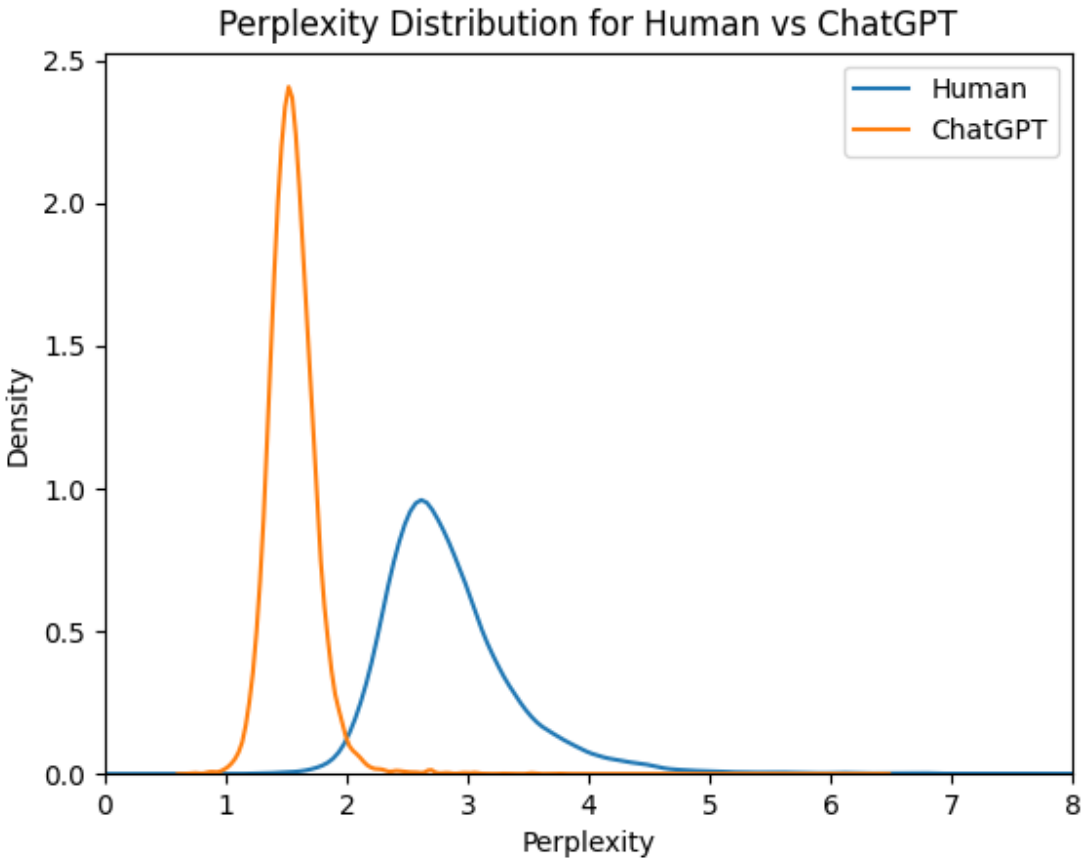

Figure XIV: Distributions of Perplexity Values for the HC3 Dataset

Perplexity is a statistical metric used to measure the effectiveness and compare different language models on a certain dataset. Perplexity score is the average uncertainty of predicting the next word in a sentence given the previous words. Higher perplexity means greater uncertainty, which is the result of a more diverse usage of vocabulary in the given dataset. Conversely, lower perplexity means less uncertainty, which can be attributed to a statistically less diverse average

usage of vocabulary and thus leads to more consistent valid predictions. Mathematically, perplexity is the exponentiated average negative log-likelihood of a sequence of tokens.

Figure XIV illustrates the perplexity values across the HC3 dataset for human and Chat-GPT generated responses. The graph shows that Chat-GPT generated responses have a lower perplexity and a much more dense distribution compared to that of human responses and are generally between a perplexity of one and two. In contrast, human responses have a wider distribution along the perplexity axis, and are not as densely populated as the Chat-GPT perplexity distribution. The majority of human responses are distributed between a perplexity of two and four.

The general trend that can be analyzed from this graph is that predicting the word usage in Chat-GPT generated responses carries less uncertainty than predicting the word usage in human generated responses. If a classification experiment were to be run utilizing the perplexities of this dataset then a clear threshold can be seen at a perplexity score of two in distinguishing between human and machine generated responses. Additionally, based on the perplexity statistical metric, the graph supports that the Chat-GPT and human responses in the HC3 dataset have clear characteristic differences that can be distinguished by the distribution and commonality of varied vocabulary used in a sequence of words.

4.2.2 t-SNE Visualizations:

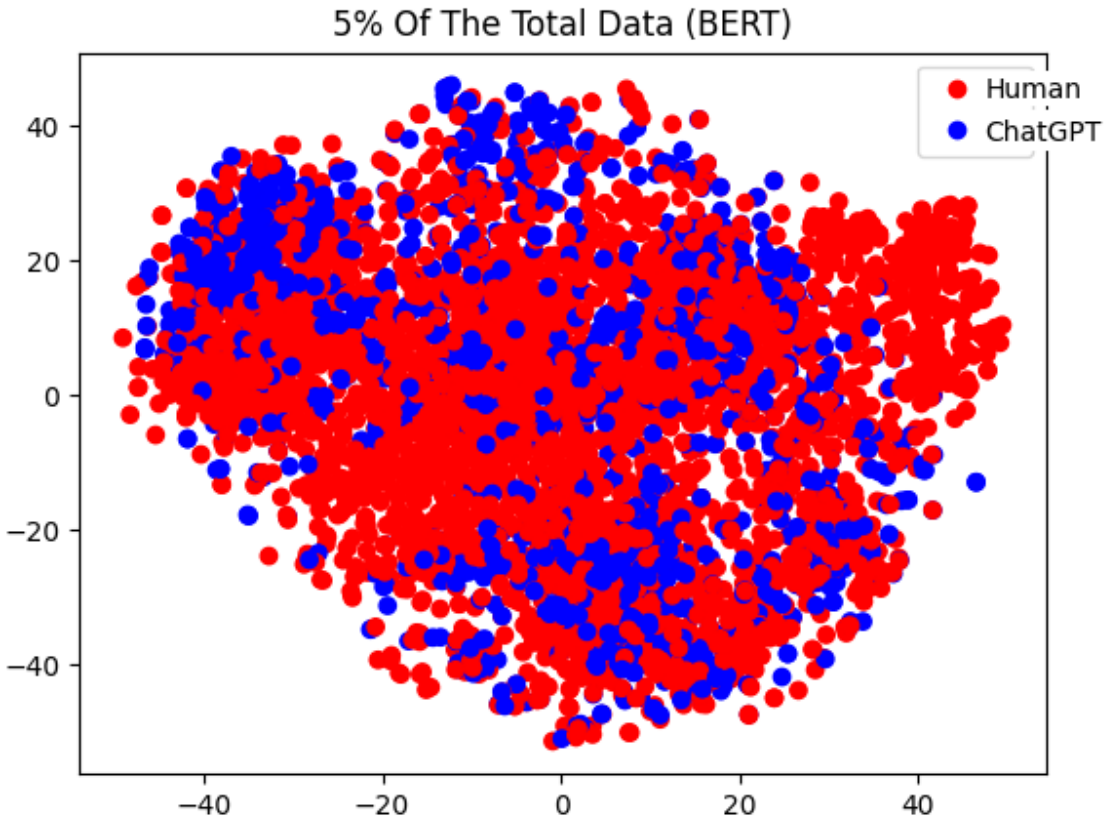

Figure XV: t-SNE visualization on 5% of the HC3 dataset using the BERT language model (Red = Human generated responses; Blue = ChatGPT generated responses)

We further analyzed the HC3 dataset with the algorithm called t-SNE (t-distributed Stochastic Neighbor Embedding). We used t-SNE to project the high dimensional data onto two dimensional space in order to visualize differences between human and Chat-GPT generated responses that can not be separated linearly. The method utilizes similarities between features and data points to compute the distances between data points. Therefore clustering behavior can be interpreted as data points that have similar features.

Figure XV has 2 classes: red dots that represent human response tokens and blue dots that represent Chat-GPT generated response tokens. The tokens are BERT-embeddings of the HC3 dataset extracted from the pretrained BERT model provided by HuggingFace. In accordance with the BERT model, the 2927 human generated responses and 1345 Chat-GPT generated responses are converted into vectors containing 768 features for each sample. t-SNE transposes and displays the dataset on a 2D plane illustrated in Figure XV.

The tokens are organized in a probability distribution that represents similarities between the tokens. The standard deviation (variance) as illustrated by the figure is low, which can be attributed to a low perplexity value. The clusters of both red and blue points are very close together with many overlaps with essentially no space between the clusters. The distinction between the points cannot be said to be reasonably clear and the clusters are quite scattered. As characterized by the HC3 dataset, human answers are present in larger quantities by a factor of two in comparison to the machine generated answer. For this reason, the larger quantity of red data points can be explained.

In the experiment using the BERT language model and unfrozen weights trained on 5% of the dataset, a score of 0.95 was yielded for testing accuracy, 0.93 for precision (macro average), 0.96 for recall (macro average), and 0.94 for F1-score (macro average). From these results, it can be concluded that BERT was able to effectively distinguish between the data illustrated by Figure XV.

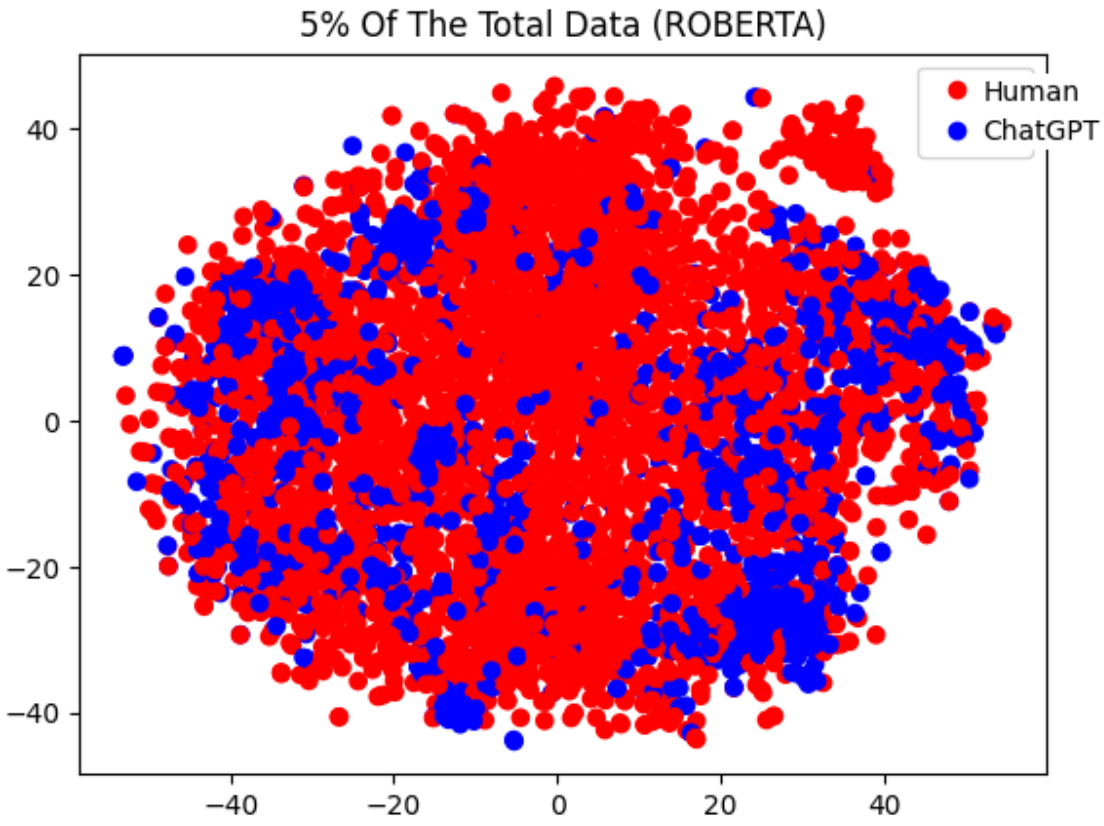

Figure XVI: t-SNE visualization on 5% of the HC3 dataset using the RoBERTa language model (Red = Human generated responses; Blue = ChatGPT generated responses)

Figure XVI shows the t-SNE visualization on 5% of the HC3 dataset using the RoBERTa language model. The method of generating this visualization is the same as the one used in figure XV, however RoBERTa is employed for token embeddings rather than BERT. The visualization is noticeably distinguishable from figure XV, as the shapes, positions, and sizes of the clusters are all different. This means that the way the RoBERTa model understands the dataset is clearly different from the way the BERT model understands the dataset. Compared to figure XV, figure XVI exhibits greater clustering behavior with less scattering. This means the RoBERTA model is better able to distinguish the similarities between the features of tokens of the same data point color, in addition to the differences between the features of tokens of differing data point color.

In the experiment using the RoBERTa language model and unfrozen weights trained on 5% of the dataset, a score of 0.98 was yielded for testing accuracy, 0.97 for precision (macro average), 0.98 for recall (macro average), and 0.98 for F1-score (macro average). From these results, it can be concluded that RoBERTa was able to effectively distinguish between the data illustrated by Figure XV. Furthermore, all of these performance metrics yield a higher score than the BERT model. Thus, it can be concluded that RoBERTa is a more effective natural language processing model for classifying the HC3 dataset than BERT. Both the performance metrics of the language models and t-SNE visualizations support this conclusion.

5. Discussion:

5.1 Future Work:

Through our experimentation, we have found that there is much to be learned and researched more thoroughly related to particular aspects of our project, results, and goals. More directly, it is important that further effort is put into the different strengths and weaknesses that the BERT and RoBERTa models bring to ChatGPT detection, researching different data sources that would help detect alternate aspects of ChatGPT's response generation, and the implications of the different statistics we used to measure our final model.

The first discussion point of note is the differences in the performances of the BERT and RoBERTa models given different conditions. BERT's frozen performance and RoBERTa's frozen performance are very unexpected as the RoBERTa model performed substantially worse than BERT when using limited data (See Figures X - XII) , even though the RoBERTa model is meant to be a further tweaked upgrade from BERT. It is possible that the data set we chose has some effect on this result. We used data sets related to questions and answers, which is a topic that RoBERTa has a distinct, alternate model for: RobertaForQuestionAnswering. It is not certain that this model would perform better, but this would be worth looking into. It's also interesting that this trend does not continue to the runs where 100% of our data was used. When given larger amounts of data, RoBERTa works better than BERT as one might expect. Another distinct result worth investigating related to this topic is the abrupt increase in precision for RoBERTa's unfrozen 20% run (see figure XI). This result seems so far outside of the trend that it is worth confirming its validity through more experimentation. One final trend of note is the variance in performance for the RoBERTa and BERT unfrozen runs. One would assume that more data to be trained on would lead to a better performance for any given model. However, it would seem that although they still performed better than their frozen counterparts, the unfrozen runs of BERT and RoBERTa possibly overfit the data when given large portions of the dataset to fine tune with.

The next possible topic to investigate with further research is the search for other datasets that involve different ways that ChatGPT responds. Our project's data set was centered around the dataset being made of questions and answers. However, ChatGPT is not designed for one-off questions and answers. It is designed to hold entire conversations and learn from them. A possible solution for this disconnection would be to find a data set of interviews and then go through the topics in the interviews with ChatGPT. This would hopefully keep some kind of repeatable format to the dataset, while implementing ChatGPT in more of its natural state. Even if there were to be other one-off formats that would be run through ChatGPT, such as essay prompts or role play suggestions, those different response types may create a larger basis for what the totality of ChatGPT can do. Further research into this topic would clarify any of these points of interest.

The final topic that we propose to be researched further is how to further optimize particular statistical achievements over others. It is clear that we would like our detection tools to work to the best of their ability. However, not everyone may agree on what statistics should be most catered to. For example, if a school used a ChatGPT detection tool: students would likely desire for precision to be most optimized as that would allow for the least number of human

writers to be flagged as probably generated by ChatGPT, while staff may want to prioritize recall so they could catch all possible people who may have just used ChatGPT for their writing assignments. This conflict cannot be easily resolved but is important to keep in mind for any work on this topic and what the possible strengths or weaknesses of future models could be.

Overall, we believe that through our work, we have collected knowledge that would be useful for any group moving forward and we hope that these topics could be discussed even further in a future project.

5.2 Conceptual Discussion:

From the beginning of the project, this group has been excited mainly by a few key factors:

1. The Wide-spread use of ChatGPT
2. The misuse of ChatGPT in several environments
3. The future possible uses of such a complex text generation model.

Given that these topics are of interest for us, we believe that it is important for the scope of this project that we mention our now-well-researched thoughts on these topics, even though we recognize that it takes more than one project to completely understand such an enormous subject matter.

As shown by the table below, ChatGPT has become extremely popular over the first months of its existence:

Month	Number of Visits
November 2022	152 million
December 2022	226 million
January 2023	616 million
February 2023	1 billion
March 2023	1.6 billion
April 2023	1.8 billion
May 2023	1.8 billion
June 2023	1.6 billion

Table IV: Table of Monthly ChatGPT Users for the First 8 Months of its Existence [10]

As the website grows, so does the conversation surrounding it. According to YouGov [11], almost half the population of the United States have heard about ChatGPT, even when considering older age groups. However, ChatGPT hasn't just blown up in the US, as we are only 12.12% of its

user base [6]. It's difficult to explain how fast this growth is. To give context, if ChatGPT was a social media, it would be the second fastest social media ever to get one million users, just behind the recently released "Threads" [6]. However, because of some difficulties controlling the misuse of ChatGPT, there have been some quite negative side-effects of ChatGPT's widespread use.

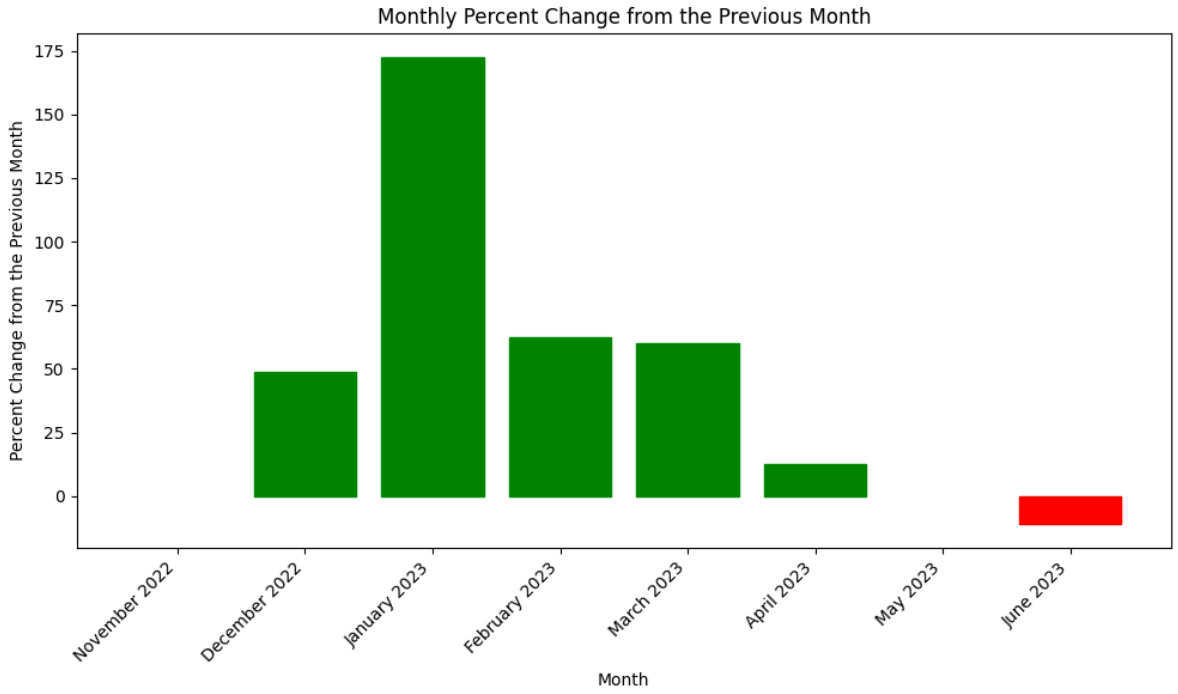

Figure XVII: Percent Change of ChatGPT User Quantity by Month

The text generation ability of Large Language Models over the past few years has been astonishing. Even the base models we used to predict ChatGPT have been developed within the past 5 years. However, there are always people willing to misuse tools for their own benefit, and ChatGPT is no exception. There have been so many examples of people using ChatGPT in immoral ways: a man generated a fake news story about a train crash [12], a lawyer used chatGPT for case research and cited cases that didn't exist [13], and, of course, students have been using ChatGPT to write their essays for them [14]. There are many moral yet unclear reasons to not use ChatGPT as the solution to every problem. However, there are much clearer laws around plagiarism and creating false panic. As the world gains more familiarity with the technology, more laws will come into place that keep malicious actors from misusing it, but for now, it is important to the continued use of this tool that we look further into tools on how we can detect it. However, when properly controlled, it is likely that ChatGPT will be a very incredible tool, as it already has left such an impact. There are many different lives that have been improved greatly by the presence of ChatGPT, and many possible future uses of a software like ChatGPT that are exciting. Writers can use ChatGPT for writing inspiration; Programmers can ask ChatGPT for advice on most popular languages. These Large Language Models are tools made for our collective benefit, but we must work to keep others from using the models for the detriment of others.

6. Conclusion:

In this paper, we implemented the BERT machine learning model on a classification problem to detect human and machine-generated text. We compared BERT to previous language models based on unidirectional architecture and highlighted its advantage of bidirectional architecture, allowing tokens to incorporate context from both previous and subsequent tokens. The pre-training methodology of BERT involving masked-language modeling and next-sentence prediction tasks on a large corpus of data was discussed.

To fine-tune the RoBERTa model, we followed a similar methodology and experimented with different data percentages and frozen/unfrozen parameter variations. We evaluated the performance of both BERT and RoBERTa models using metrics such as testing accuracy, precision, recall, and F1-score.

Our experiments revealed interesting trends in the performance of BERT and RoBERTa under varying conditions. We observed that the RoBERTa model showed weaker performance than BERT when trained with limited data but outperformed BERT when trained with larger datasets. The study also highlighted the need for further research into finding appropriate datasets that mimic ChatGPT's natural state of conversation and how to optimize statistical achievements according to different user requirements.

As ChatGPT's popularity continues to grow, there is a pressing need to address the misuse of such powerful language models. We emphasized the importance of detecting machine-generated text to combat issues such as fake news, plagiarism, and unethical practices.

Looking ahead, we propose future work that involves investigating the differences in performances of BERT and RoBERTa models, exploring diverse datasets that represent various aspects of ChatGPT responses, and optimizing statistical metrics according to specific use cases.

In conclusion, ChatGPT and similar large language models have immense potential for positive impact on various aspects of society. However, it is crucial to address the challenges of misuse and ethical considerations to ensure that these tools are harnessed responsibly for the collective benefit of humanity. With further research and development, we can unlock the true potential of large language models and facilitate their integration into various applications while mitigating their potential risks.

7. References:

1. Gehrmann, S., Strobelt, H., & Rush, A. M. (2019). GLTR: Statistical Detection and Visualization of Generated Text. arXiv preprint arXiv:1906.04043 [cs.CL]. https://doi.org/10.48550/arXiv.1906.04043
2. Sadasivan, V. S., Kumar, A., Balasubramanian, S., Wang, W., & Feizi, S. (2023). Can AI-Generated Text be Reliably Detected? arXiv preprint arXiv:2303.11156 [cs.CL]. https://doi.org/10.48550/arXiv.2303.11156
3. Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. arXiv preprint arXiv:2301.11305, 2023.
4. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692 [cs.CL]. https://doi.org/10.48550/arXiv.1907.11692
5. John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. arXiv preprint arXiv:2301.10226, 2023.
6. Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2019.
7. Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., & Wu, Y. (2023). How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. arXiv preprint arXiv:2301.07597 [cs.CL]. https://doi.org/10.48550/arXiv.2301.07597
8. Naseer, M., Windiatmaja, J. H., Asvial, M., & Sari, R. F. (2022). RoBERTaEns: Deep Bidirectional Encoder Ensemble Model for Fact Verification. Big Data Cognition and Computing, 6(2), 33. https://doi.org/10.3390/bdcc6020033
9. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805 [cs.CL]. https://doi.org/10.48550/arXiv.1810.04805
10. Duarte, F. (2023, March 30). Number of ChatGPT Users (2023). Exploding Topics; Exploding Topics. https://explodingtopics.com/blog/chatgpt-users
11. Orth, T. (n.d.). What Americans think about ChatGPT and AI-generated text | YouGov. YouGov; YouGov. Retrieved August 8, 2023, from https://today.yougov.com/topics/technology/articles-reports/2023/02/01/what-americans-think-about-chatgpt-and-ai-text
12. Tan, H. (2023, May 9). *China Detains Man Who Used ChatGPT to Generate Fake News*. Business Insider; Insider. https://www.businessinsider.com/chatgpt-artificial-intelligence-ai-fake-news-tech-china-detains-man-2023-5
13. Armstrong, K. (2023, May 27). ChatGPT: US lawyer admits using AI for case research - BBC News. BBC News; BBC News. https://www.bbc.com/news/world-us-canada-65735769
14. Nolan, B. (2023, January 14). Professors Caught Students Cheating on College Essays With ChatGPT. Business Insider; Insider. https://www.businessinsider.com/chatgpt-essays-college-cheating-professors-caught-students-ai-plagiarism-2023-1