Machine-Generated Text Detection for ChatGPT

Haoquan Zhou and **Kexuan Huang** and **Haoyang Ling**
University of Michigan
{haoquanz, hkx, hyfrankl}@umich.edu

1 Introduction

1.1 Background

With significant advances in the field of deep learning, large language models (LLM) were used to solve complicated Natural Language Processing (NLP) tasks like natural language understanding and natural language generation. Recently, OpenAI published its newest conversational AI model named ChatGPT (OpenAI, 2022). With its incredible power to generate text under various topics, ChatGPT ignited heated discussion within the NLP community as well as the whole public. People involved in different fields began to test the performance of ChatGPT with their domain-specific questions. And the model handled most of the questions astonishingly well (Jiao et al., 2023; Biswas, 2023; Dowling and Lucey, 2023).

AI-based natural language text generation is not a newly emerged task. The initial idea of text generation dates back to 1990 when Elman proposed a recurrent neural network (RNN) to generate language (Elman, 1990). Later models including Generative Pre-trained Transformers (GPT) were built on the foundation of similar works. Upon now, these large language models already can generate sophisticated natural language text with clear logic.

Despite the commendable performance, concerns were raised by researchers regarding the usage of ChatGPT as well as the ethics behind it. One major concern is that ChatGPT might unrealistically improve students' performance in online exams (Susnjak, 2022). It's argued that ChatGPT could generate highly realistic text with minimal input, making it a potential threat to the integrity of online exams, which may influence the fairness of the exams. More generally, there are ethical problems that consistently exist in LLMs, which apply to ChatGPT as well. Due to biases hidden in the training dataset, it's argued that ChatGPT has certain ethical risks (Zhuo et al., 2023).

1.2 Project Goals

As mentioned above, with the rising fluency and factual knowledge of large language models, many schools have banned ChatGPT from school networks and devices over concerns about students' potential cheating. In this project, we intend to investigate the ChatGPT model and propose a machine-generated text detector for the GPT-3.5 model. The project goal is composed of three parts: (1) explore statistical information of machine-generated text (word frequency, text length, or even sentence structure); (2) build the discriminator model on a mixture of machine-generated text and human-written text; (3) propose data augmentation methods to downstream tasks like Q&A systems or text completion based on the analysis of the obtained discriminator and visualize the weights/attention.

To reach our project goal, we construct a BERT-based classifier and an improved TF-IDF classifier to classify a given corpus into two classes: machine-generated text and human-written text. The model is first trained on the HC3 dataset and some similar datasets, which contains Q&A pairs both from human and machine. We then continuously fine-tune the dataset by taking out easily recognized patterns or words or statistical information. We hope by this data argumentation step, the dataset will be able to train more robust discriminators.

As the discriminator model will learn the patterns of machine-generated text, it will be helpful for schools to identify those effortless work and appeal to parents to pay attention to their children's academic performance. As for the NLP community, all the tasks will contribute to understanding the machine-generated text and arouse ethical concerns about the power of AI.

2 Data

2.1 Dataset

Regarding the ChatGPT data set, the most recent GPT-3.5 API imposes an upper limit on the frequency and amount of text generation, so we are going to use the Human vs. ChatGPT Comparison Corpus (HC3) published by Guo et al. (2023) under CC BY-SA 4.0 license, MGTBench published by He et al. (2023) under MIT license, and two similar open dataset ChatGPT-RetrieveQA by Askari et al. (2023) and GPT-Wiki by Aaditya Bhat (2023), allowing us to use, adapt and share the data set under the same license. The Tbl. 2 below provides basic information for each dataset.

The motivation for Guo et al. (2023) to build this data set was to study the features of ChatGPT responses, the differences and gaps from human experts, and future directions for LLMs. In contrast, we focus more on machine-generated text detection and data augmentation. The intention for He et al. (2023) is to present a benchmark for detecting machine-generated text. It re-examines six metric-based methods including log rank and entropy and some model-based models like OpenAI detector.

In Tbl. 2, we list the basic information of the dataset including the number of human text and the number of machine-generated text.

Dataset	Human Text	Machine Text
SQuAD1	1000	1000
TruthfulQA	817	817
NarrativeQA	2000	1000
HC3	58546	26885
RetrieveQA	58546	26885
GPT-Wiki	150000	150000

Table 1: Basic Statistics of Dataset: dataset size

2.2 Answer Length

For the data exploration part, we visualize the average answer length of the human text and machine-generated text. Astonishingly, we find that the human response is shorter than the machine-generated one shown in Fig. 1, which may be a potential issue in these datasets. However, it is very surprising that GPT-Wiki has less length than human text, which is different from others. It can be a breakthrough point in detection. Also, we notice that the prompt asks the ChatGPT to generate around 200 words. However, the text is on average shorter than 200, which indicates that GPT can't well understand the number in the prompt.

2.3 Sentence Length

In addition to answer length, we also explore distribution of the average number of word in a sentence for machine-generated text and human text. We find that the human responses have shorter sentences than the machine-generated ones shown in Fig. 2. The reason could be machine-generated texts tend to use more complex or compound sentence structures than human texts, while human tend to use simplified grammar to form short sentences or split one long sentence into multiple short sentences. However, in GPT-Wiki dataset this is also not the case. The reason could be that human use different styles of writing in Q&A and Wiki-documenting. The expression for Q&A might be more casual, but for Wiki-documenting, it might be more formal and rigorous.

2.4 Word Frequency

In this part, we use the Brown Corpus by Francis and Kucera (1979) as the corpus to evaluate the word frequency to see how rare words will appear in the ChatGPT-generated text accompanied by the average length of words. From Fig. 3, we can find that the distribution of human words and the distribution of ChatGPT answers are similar to each other. However, ChatGPT is more likely to generate frequent words. It may be because the loss function of GPT is to achieve the high likelihood of the generated sentence which may value frequent words.

2.5 POS Tagging

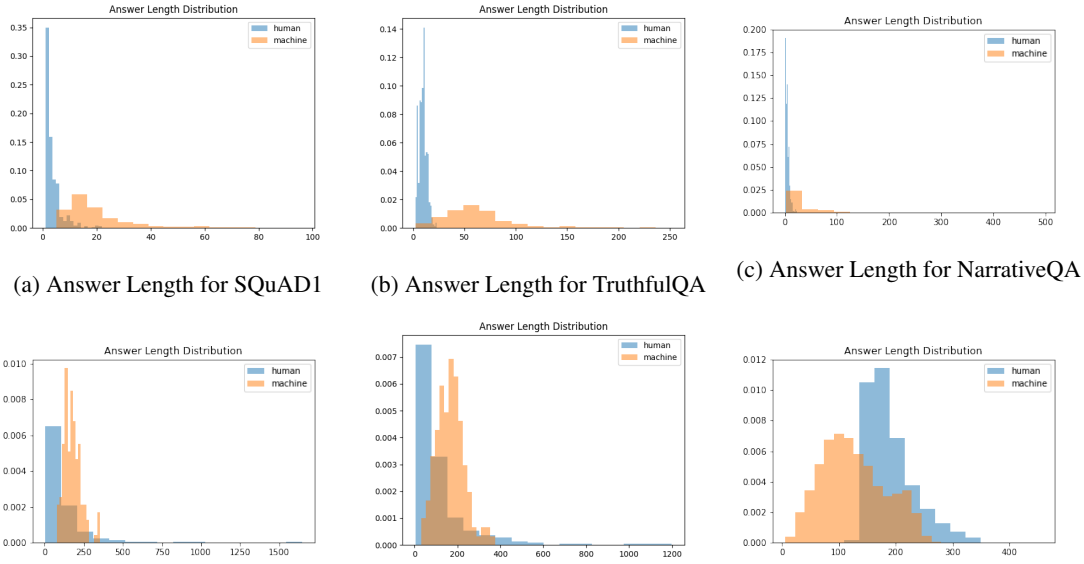
We use the POS tagging embedded in the Stanford stanza Python library by Qi et al. (2020) to decode the sentence and find that the distribution of machine-generated text and human text are similar to each other which means that ChatGPT has already learned the ability to mock human text at the word level.

2.6 Dependency Parsing

For dependency parsing, we can observe the clear difference between human text and machine-generated text. It may be a good indicator of machine-generated text. It may be because the machine emphasizes so much on some frequently-seen sentence structures that it leads to some deviation.

2.7 Data Preprocess

MGTBench: as this dataset is published online, we adopt the preprocessing function by He et al.



(a) Answer Length for SQuAD1 (b) Answer Length for TruthfulQA (c) Answer Length for NarrativeQA

(d) Answer Length for HC3 (sampled) (e) Answer Length for RetrieveQA (sampled) (f) Answer Length for GPT-Wiki

Figure 1: The Distribution of Answer Length on Different Datasets

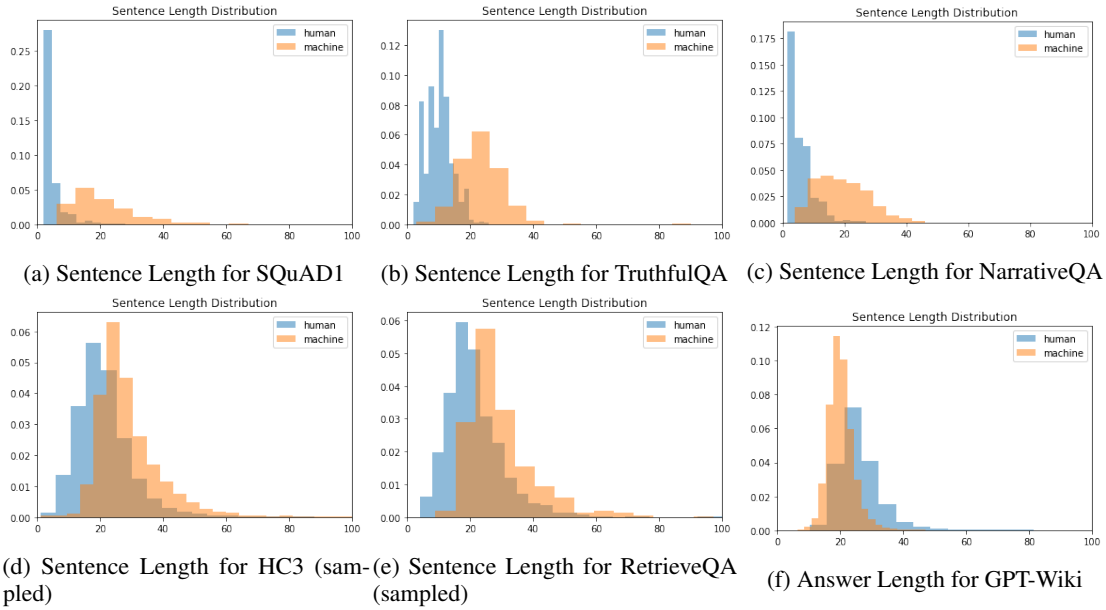
(2023). We split 80% for training data and 20% in the development set. Since the dataset in MGT-Bench generated from SQuAD1, TruthfulQA, and NarrativeQA only contains around 1000 lines, they will serve as the benchmark to test the performance of our model against the baseline. We follow the almost same pre-process for HC3 and focus our attention only on the answers and extract human answers from the dataset to ensure that the number of human text and machine-generated text is 50-to-50. The sample from one of the benchmarks, NarrativeQA, is shown in Tbl. 2.

HC3 Dataset: HC3 contains nearly 40K questions and their corresponding human/ChatGPT answers. Consider putting 80% of the data in the training set and 20% in the development set, we have more than 80k data for training, which can be considered large enough for us to build our models. Also, the content inside the data set ranges from a wide variety of domains, including open-domain, financial, medical, legal, and psychological areas, which can extensively decrease the bias brought by experimenting on a certain knowledge domain. All datasets in HC3 are separate JSONL files with the same fields each line, and we convert them to CSV files before experiments. The dataset follows the same format as the MGTBench samples in Tbl. 2.

For GPT-Wiki and RetrieveQA, we follow the same data preprocessing methods.

3 Related Work

Some researchers have approached the problem with supervised learning from scratch. Solaiman et al. (2019) used a simple baseline model based on the bag of words and TD-IDF vector that may encounter the curse of dimensionality (Fagni et al., 2021). Gehrmann et al. (2019) applied statistical analysis with text visualization by assuming a sampling distribution in LLMs. As one main goal of the text generative models is to generate convincing and on-topic text, the authors of the GROVER model studied its ability to detect its generated news article by fine-tuning itself (Zellers et al., 2019) surpassing strong discriminators like BERT. Solaiman et al. (2019) also experimented with the fine-tuning of the RoBERTa language model with nucleus sampling. However, Mitchell et al. (2023) have pointed out that this kind of approach may lead to the over-fitting of their training distribution of domains or source models. Therefore, Mitchell et al. (2023) proposed a zero-shot detector DetectGPT based on the assumption that the perturbation of machine-generated text tends to have a lower probability under the model than the original text (Mitchell et al., 2023). Its approach employs the features of the loss-like function. Additionally, Kirchenbauer et al. (2023) explored watermark to generate easily detected text, which is not the scope of this project. Some existing issues involve generalizability, interpretability, robustness,



(a) Sentence Length for SQuAD1 (b) Sentence Length for TruthfulQA (c) Sentence Length for NarrativeQA

(d) Sentence Length for HC3 (sampled) (e) Sentence Length for RetrieveQA (sampled) (f) Answer Length for GPT-Wiki

Figure 2: The Distribution of Sentence Length on Different Datasets

text	label
Because she was actually married to Arthur Huntingdon.	0
Pozdnyshev was acquitted of murder because of his wife's Drexl was killed by Clarence Worley.	1
He killed one of Almayer's slaves and put his ring and ankle bracelet on the corpse to make it look like himself.	0
The disk used by Starman to understand English is the Voyager 2 space probe's gold phonographic disk.	1
Mary Horowitz is a crossword puzzle writer for the Sacramento Herald, as mentioned in the given context.	1

Table 2: Sample Dataset after Preprocessing (0 for human-written text, 1 for machine-generated text)

and sentence-level accuracy (Jawahar et al., 2020; Guo et al., 2023). Recently, (He et al., 2023) published a benchmark called MGTBench for evaluating machine-generated text detection with some baseline models including statistical analysis and transformer-based models. However, compared with the HC3 dataset, those datasets only include around 1000 machine-generated text, so they are small datasets.

Innovation: Our project proposes to improve the performance of the discriminator model with statistical learning that is comparable to BERT-based models and then implement data augmentation that previous ones seldom focus on like truncation, splitting, and summarization.

4 Methodology

4.1 NLP Task Definition

The Machine-Generated Text Detection task can be defined as follows: given a paragraph of sentences $I = (s_1, s_2, ..., s_n)$ as input, the model f is a map from input to output $O \in \{0, 1\}$, which represents whether the text is generated by large language models, with maximum likelihood. In other words, the model f is defined by

$$f : I = (s_1, s_2, ..., s_n) \rightarrow O \in \{0, 1\}$$

and

$$f = \textbf{argmax}\ \Pr[O \mid (s_1, s_2, ..., s_n)]$$

For data augmentation part, the set $S = \{g_1,\ g_2,\ ...,\ g_n\}$ contains the data augmentation methods g that

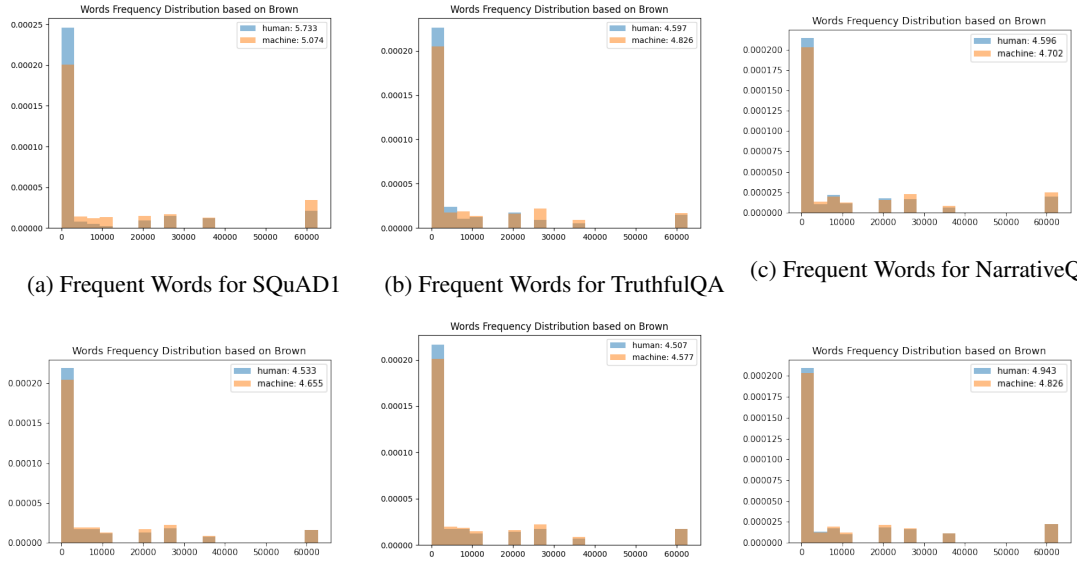
$$g : I_{\text{machine}} \rightarrow I'_{\text{machine}}$$

where $I_{machine}$ is the machine-generated dataset and $I'_{machine}$ is the dataset that contains the augmented machine-generated text. If we use $\texttt{Sim}$ to represent similarity, I' has the following properties:

$$\texttt{Sim}(I_{\text{human}}, I_{\text{machine}}) < \texttt{Sim}(I_{\text{human}}, I'_{\text{machine}})$$

so that the discriminator trained in the first stage may find it difficult to discriminate the text. For the data augmentation methods, we intend to investigate the weight and attention to find some best augmentation methods g

$$g = \textbf{argmin}\ \texttt{Sim}\left(I_{\text{machine}}, I'_{\text{machine}}\right)$$



(a) Frequent Words for SQuAD1 (b) Frequent Words for TruthfulQA (c) Frequent Words for NarrativeQA

(d) Frequent Words for HC3 (sampled) (e) Frequent Words for RetrieveQA (sampled) (f) Frequent Words for GPT-Wiki

Figure 3: The Distribution of Word Appearance in Drown Corpus on Different Datasets

or

$$g = \textbf{argmax}\ \texttt{Sim}\left(I_{\text{human}}, I'_{\text{machine}}\right)$$

4.2 Classifiers

We have trained several classification models to discriminate machine-generated text. Some of them have already reached extraordinary performance on the datasets. For the downstream tasks, we actively looked for possibilities to train a neural network to modify the text.

4.2.1 Dummy Classifier

We trained two dummy classifiers that either randomly or uniformly assign the class number to the corpus. The AUC value for these classifiers is served as the baseline of our model.

4.2.2 TF-IDF Classifier

In this setting, we train a logistic regression model with sklearn library. We first vectorize the text with TF-IDF, the product of two statistics, term frequency, and inverse document frequency. Considering those infrequent words may affect the accuracy, we only keep those words appearing in at least ten documents. Logistic regression helps to turn the document vector representation into a probability to judge whether the document is generated by the machine. It turns out that the results is quite above our expectation. It is worth investigating further.

4.2.3 Word2Vec & Attention Classifier

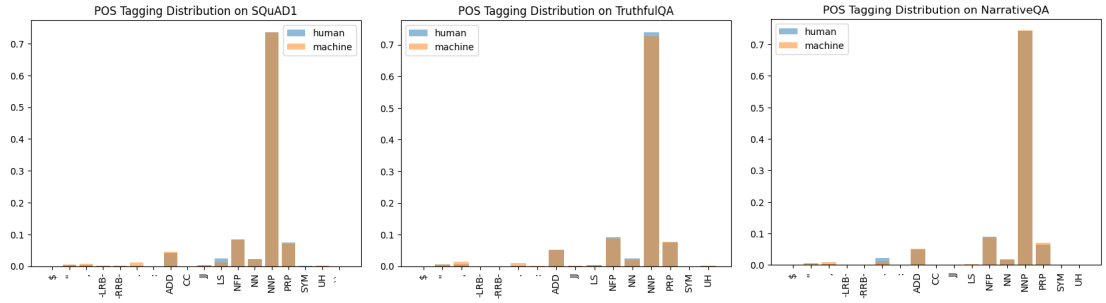
Considering the differences in wording between machine-generated text and human text, the Word2Vec approach might perform well in our problem setting. We trained a Word2Vec model to transform the words into embeddings and used a multi-head attention classifier to discriminate the texts. We also printed out heat maps of attention heads to search for some potential wordings that are universally paid attention to. We expected to learn useful strategies for downstream tasks.

4.2.4 Bert-based Model

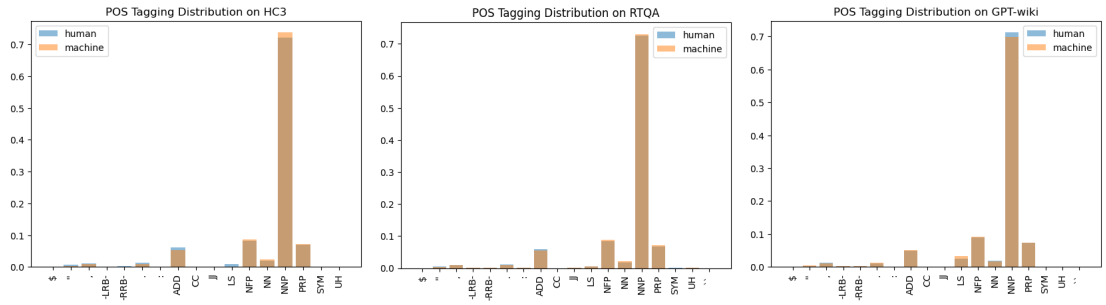
In this setting, we adopt the pre-trained Bert-based model from Hugging Face called MiniLM in the upstream to extract embeddings and apply the classification model as the downstream task. We trained the model based on the text only. As ChatGPT is a generative pre-trained transformer, we'd like to see whether a transformer-based model will be a good discriminator than other kinds of models.

4.2.5 TF-IDF Classifier with Dependency Parsing

In this setting, we use a logistic regression model with sklearn library with the information about dependency parsing of the text. We follow the same method mentioned in section TF-IDF Classifier where we use the TF-IDF to vectorize the text.



(a) POS Tagging Distribution for SQuAD1 (b) POS Tagging Distribution for TruthfulQA (c) POS Tagging Distribution for NarrativeQA



(d) POS Tagging Distribution for HC3 (sampled) (e) POS Tagging Distribution for RetrieveQA (sampled) (f) POS Tagging Distribution for GPT-Wiki (sampled)

Figure 4: The Distribution of POS Tagging in Drown Corpus on Different Datasets

4.3 Test the Generalization Ability

Our classifiers mentioned above are all trained under Q&A tasks. It is doubtful whether the performance can still be high when it comes to classifying texts under other scenarios like terminology explanation.

To identify the actual performance when generalizing the model, we trained a BERT-based model on the HC3 dataset and evaluate its performance on a sampled subset from the GPT-wiki data set. We expect the performance of the model to drop by a certain degree. And our further data augmentation procedures can be built on the purpose of minimizing this performance drop.

5 Evaluation and Results

5.1 Evaluation Methods

Since there are two NLP-related tasks, we may set different criteria and baselines for them. For training the discriminator model,

1. **Evaluation Metric**: the area under the receiver operating characteristic (AUROC).

2. **Randomized Baseline**: randomly assign the label based on the ratio of the machine-generated text in the data set. The AUC value for randomized baseline is 0.498.

3. **Baseline**: use the logistic regression to the word frequency matrix (perhaps after singular value decomposition). The AUC value for the linear regression baseline is 0.946.

For the downstream task,

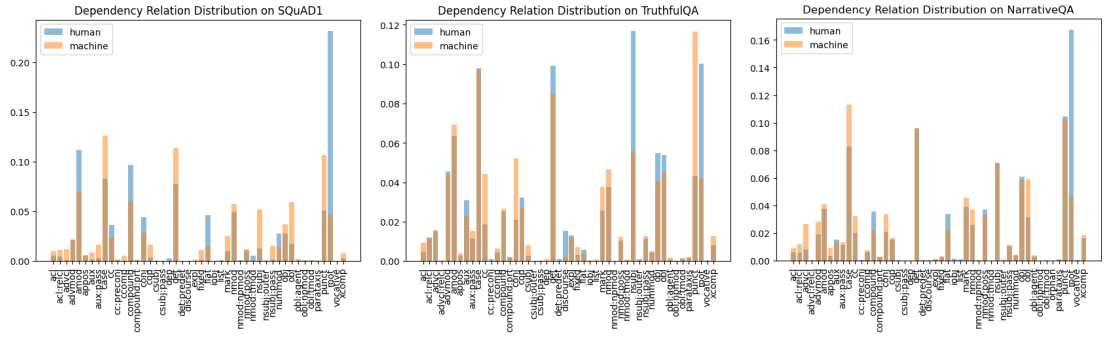
1. **Evaluation Metric**: the area under the receiver operating characteristic (AUROC).

2. **Randomized Baseline**: Randomly concatenate human-like/subjective words like "I think", and "hmmm" on the machine-generated text shown in Tbl. 3 and results are shown in Rand-HC3 column in Tbl. 5.

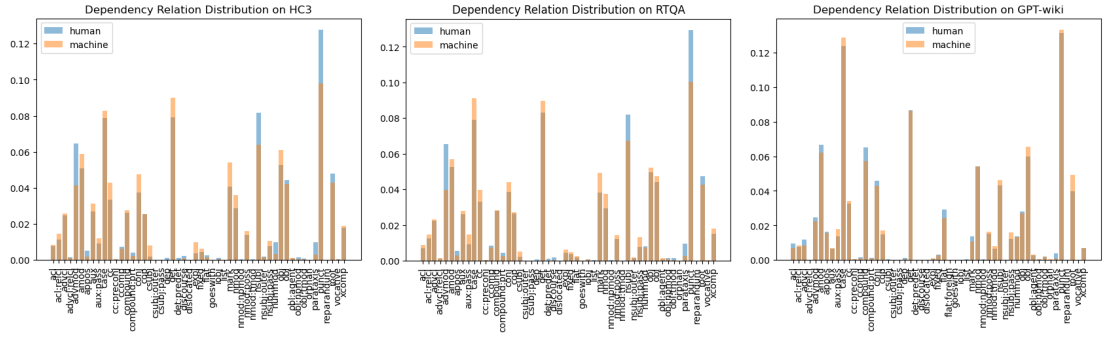
3. **Baseline**: as the machine-generated text tends to be longer than the human text, we try to truncate the answer within 100 characters and results are shown in Trun-HC3 column in Tbl. 5.

5.2 Environment Setup

We split the dataset into 80% for training and 20% for testing and use AUC-ROC to evaluate the machine-generated discriminator. For the data augmentation part, we use the same metrics and compare the results before and after augmentation. We train the data with one GPU in the Great Lakes server.



(a) Dependency Parsing Distribution for SQuAD1 (b) Dependency Parsing Distribution for TruthfulQA (c) Dependency Parsing Distribution for NarrativeQA



(d) Dependency Parsing Distribution for HC3 (sampled) (e) Dependency Parsing Distribution for RetrieveQA (sampled) (f) Dependency Parsing Distribution for GPT-Wiki (sampled)

Figure 5: The Distribution of Dependency Relationship in Drown Corpus on Different Datasets

Phrases for Data Augmentation Baseline
I find that
One can find that
Maybe you can find that
Hmmm, I think that
With my knowledge, I think that
Hope this helps:
From my experience, I believe that
As far as I know
It seems to me that
I've heard that
If I had to guess, I'd say that
I'd like to suggest that
In my opinion,
It's possible that
I'm inclined to think that
After careful consideration, I've concluded that
Perhaps it's worth considering that
I've noticed that
Based on my understanding,
It's my belief that
If you ask me,
Personally, I feel that
In my humble opinion,
Emmm

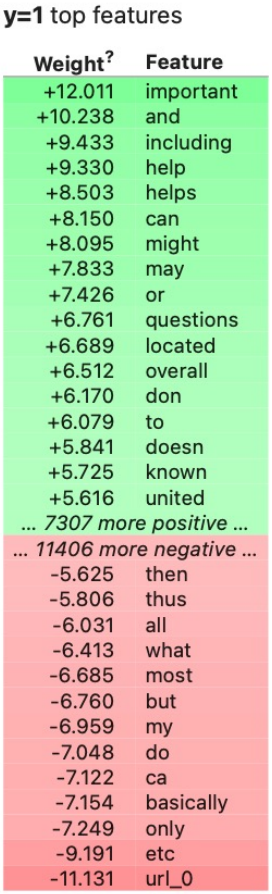
Table 3: Phrases for Data Augmentation Baseline

5.3 Results and Discussion

Model: Tbl. 5 and Fig. 7 show the results of the baseline model (Dummy Classifier and TF-IDF logistic regression) and the models we try (Word2Vec and MiniLM).

The performance of the Word2Vec attention model and the Bert-based model outperforms the baseline models. However, we are surprised by the excellent performance of the TF-IDF classifier. In the baseline models, the TF-IDF logistic regression model implemented with sklearn performs remarkably well, with a huge increase in AUROC from 0.5 to nearly 0.95 than the dummy classifier. Regarding TF-IDF logistic regression model, according to the feature weight in Fig. 6, we guess that since TF-IDF is quite good at emphasizing the importance of words by identifying unique and informative words in a document, human-written texts tend to produce similarly on spoken language, e.g. 'etc', 'basically', 'do' and 'my'. In contrast, machine-generated texts tend to use more neutral and non-absolute words, e.g. 'can', 'might', 'may'.

Basic Augmentation: For the data augmentation part shown in Fig. 7, we use four methods: (1) concatenate human-like words; (2) truncate both



y=1 top features

Weight[?]	Feature
+12.011	important
+10.238	and
+9.433	including
+9.330	help
+8.503	helps
+8.150	can
+8.095	might
+7.833	may
+7.426	or
+6.761	questions
+6.689	located
+6.512	overall
+6.170	don
+6.079	to
+5.841	doesn
+5.725	known
+5.616	united
... 7307 more positive ...	
... 11406 more negative ...	
-5.625	then
-5.806	thus
-6.031	all
-6.413	what
-6.685	most
-6.760	but
-6.959	my
-7.048	do
-7.122	ca
-7.154	basically
-7.249	only
-9.191	etc
-11.131	url_0

Figure 6: Feature Weight for TF-IDF Logistic Regression Model on HC3

answers into the same length; (3) only truncate the machine-generated text into the similar length of human words; (4) split both human text and machine-generated text into three sentences. From the results, we can find that the human-like words may decrease the accuracy, but the method is not significant due to the length of machine-generated text (over 100 words on average). For the truncation (Trun-HC3 and Same-HC3), the performance of the discriminator drops significantly.

One of our guesses is that the shorter response of human-like text leads to a more "sparse" vector representation which may be the potential issue here. The shorter sentence provides less information on sentence structure to the model. For the Split-HC3, breaking the paragraphs into parts also leads to great reduction. It shows that the sentence-level prediction may be harder than the paragraph-level. Controlling the length of the response can well defeat most models in Tbl. 6.

Other Augmentation: The interesting finding on truncation inspires us to dig a little further in other augmentation methods that shorten the machine-generated text. We apply extractive summarization to machine-generated text with two methods: (1) sentence scoring with TF-IDF; (2) TextRank algorithm, which evaluate and extract certain important sentences from the text. In addition to limiting the text length, we limit the sentences length since machine-generated text tends to have longer sentences with compound structures and more words, which might also be one of the factors for discrimination. We apply sentence splitting with two methods: (1) pure grammatical approach; (2) fine-tuned T5 model. Both of the methods are able to generate the split sentences with none or low change in sentiment as shown in Tbl. 4.

However, the augmentation is not significant enough for both extractive summarization and sentence splitting, the performance of the discriminator only fluctuates around the truncation methods. Our primary interpretation is that the change made to the machine-generated text is not large enough to make a different in both lexical and syntactic level. As a result, we proceed to the ablation study with the basic data augmentation method by truncation (Trun-HC3 and Same-HC3) and split (Split-HC3).

5.4 Ablation Study

Model: In this research project, we propose a novel approach to text classification utilizing a TF-IDF logistic regression model in combination with dependency parsing. To evaluate the efficacy of this approach, we conduct an ablation study comparing the impact of dependency parsing on logistic regression performance. We test our approach on several datasets, including SQuAD1, TruthfulQA, NarrativeQA, sampled RetrieveQA, sampled GPT-Wiki, and sampled HC3, and their variations.

For the RetrieveQA dataset, we employ human responses and machine responses to 125 questions, while for HC3, we validate our model by sampling 10% of the original data, approximately 8,000 records. In contrast, for GPT-wiki, we utilize 5% of the original data, around 15,000 records. We present the study's results in Tbl. 7, which demonstrates that a larger discrepancy between human and machine-generated text leads to a significant improvement in the F1 score.

Remarkably, we find that in the SQuAD1, TruthfulQA, and NarrativeQA datasets, the logistic regression approach performs comparably to the Bert-based model. Additionally, we observe significant

Method	Original Sentence	Split Sentences
Grammatical Approach	My father, who is a President, is an honest man.	1. My father is an honest man. 2. My father is a president.
Fine-tuned T5 Model	This movie is produced by Tidy, the company she co-founded with David, who is a director.	1. This movie is produced by Tidy. 2. She co-founded it with David, who is a director.

Table 4: Sample Sentence Splitting with: (1) pure grammatical approach (2) Fine-tuned T5 Model

Classifier	SQuAD1	TruthfulQA	NarrativeQA	HC3
Dummy Classifier	0.41	0.430	0.529	0.508
TF-IDF Logistic Regression	0.924	0.882	0.834	0.946
Word2Vec & Attention Classifier	0.883	0.866	0.778	0.958
Bert-based Model	0.986	1	0.971	0.996

Table 5: AUROC for Different Classifiers on HC3 and MGTBench as Baseline

improvements in smaller datasets where the difference between human and machine-generated text in the dependency relationship is significant.

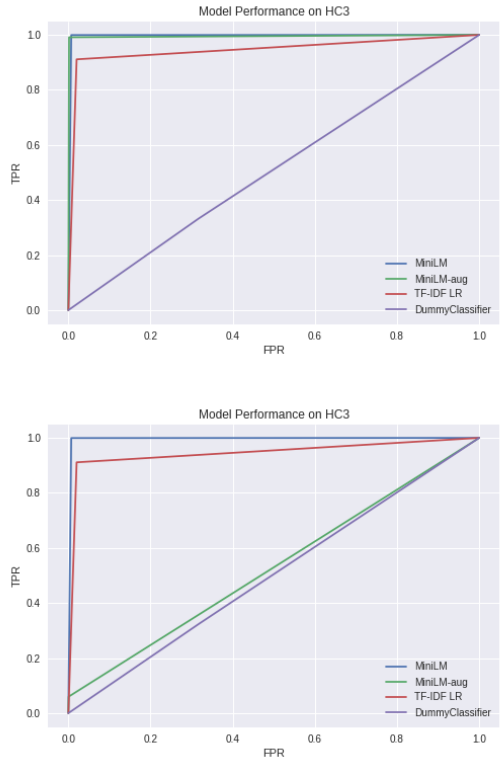


Figure 7: Model Performance on HC3: randomized augmentation (upper) and truncation augmentation (lower)

Robustness: The HC3 dataset was utilized for training the model, whereas the Wiki dataset was utilized for testing. The results revealed a decrease in the area under the curve (AUC) value, which dropped to 0.809. Further experimentation was carried out by employing a training dataset comprising equal proportions of HC3 and Wiki data, with the remaining HC3 and Wiki data utilized as the development dataset. In addition, data from SQuAD1, TruthfulQA, and NarrativeQA were utilized as the test dataset. The findings demonstrated a noteworthy decrease in performance on the test dataset shown in Tbl. 8 and Fig. 8. However, dependency parsing helps balance TPR and FPR.

The findings of our study demonstrate that alterations in the dataset distribution have a notable impact on the performance of the statistical learning model. Specifically, the performance score on the test dataset is lower than that of the development dataset, with the statistical learning model (logistic regression combined with dependency parsing) being the most affected. Although our ablation study highlights the potential benefits of statistical learning, these findings underscore the importance of a well-designed dataset to ensure model robustness beyond the ablation study.

In conclusion, these results suggest that careful attention must be given to dataset design when developing statistical learning models for real-world applications.

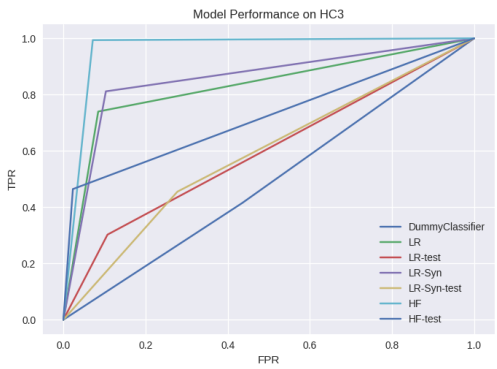


Figure 8: Transferability Test on Different Classifiers

Classifier	HC3	Rand-HC3	Trun-HC3	Same-HC3	Split-HC3
Dummy Classifier	0.508	0.501	0.504	0.498	0.499
TF-IDF Logistic Regression	0.946	0.932	0.627	0.915	0.908
Word2Vec & Attention Classifier	0.958	0.736	0.576	0.932	0.886
Bert-based Model	0.996	0.994	0.529	0.970	0.979

Table 6: AUROC for Different Classifiers on HC3 with Basic Augmentations

Dataset	LR	LR with SYN
SQuAD1	0.924	0.982
TruthfulQA	0.882	0.975
NarrativeQA	0.834	0.924
HC3 (sampled)	0.888	0.928
Same-HC3 (sampled)	0.855	0.877
Split-HC3 (sampled)	0.855	0.879
RetrieveQA (sampled)	0.694	0.863
GPT-Wiki (sampled)	0.898	0.901

Table 7: AUROC before and after considering dependency parsing

Dataset	Dev	Test
TF-IDF LR	0.828	0.598
TF-IDF with Syn	0.854	0.589
Bert-based	0.960	0.721

Table 8: AUROC on development and test dataset

6 Conclusions

In this project, we first analyze the statistical information of the dataset including answer length, sentence length, word frequency, POS tagging, and dependency parsing. We find that the answer length, sentence length, and dependency parsing may be the breakthrough, while the word frequency and POS tagging are not. It may be because those distributions can be easily collected even naive Bayes, so they are low-level statistical information.

Then, we implement a dummy classifier, TF-IDF classifier, Word2Vec, and the BERT-based model to predict the results where BERT models reach nearly 1 in the AUC score, which is astonishing to us. So, we propose a more explainable model which combines TF-IDF logistic regression with dependency parsing. It increases the performance and is comparable to BERT-based models in small datasets. Therefore, we go with the robustness test by proposed data augmentation including truncation (Trun-HC3 and Same-HC3) and split (Split-HC3) where we find that all the models are most sensitive to the change in text length than other augmentation methods such as extractive summarization and sentence splitting. The proposed model performs well in all the datasets shown in Tbl. 7.

To further test it, we test the transferability of the model where we train on some datasets and test on other datasets. It shows that the change in the statistical information influences our statistical model most which is expected. So, if we can have a well-designed dataset, it can be a good explainable model because it performs well on similar datasets with great performance improvement in small datasets.

For the future work, there are several directions that can be further pursued. Firstly, we can explore more advanced data augmentation techniques that may improve the performance. For example, we can investigate into the impact of changing writing styles on discriminator performance, which would be under the topic of paraphrase generation. Secondly, we can explore the interpretability of the proposed model. Although some models have achieved good performance, it is still important to understand how they arrives at its predictions. Therefore, we can investigate techniques such as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) to provide more transparency and interpretability to our model. Overall, there are many opportunities for future work in this area, and we hope that our research can provide some insights for further investigations into machine-generated text detection more than ChatGPT.

7 Other Things We Tried

In our pursuit to enhance the data augmentation performance without altering the sentiment, we explored several approaches, as mentioned in the Section 5.3 Results and Discussion. We attempted to leverage extractive summarization on machine-generated text using two techniques: (1) sentence scoring with TF-IDF, and (2) TextRank algorithm. These methods aimed to identify crucial sentences in the text, but the results were not significantly better than the basic truncation approach. Additionally, we experimented with two sentence splitting methods: (1) purely grammatical approach, and (2) fine-tuned T5 model that generates split sentences from compound sentences. Despite our best efforts, we found that these approaches were not signifi-

cantly better than basic truncation. Furthermore, the processing time for both summarization and sentence splitting was considerably longer, with approximately 10 seconds required for each text in the dataset and several hours for the whole dataset.

8 What We Would Have Done Differently

In this section, we discuss what we would have done differently if we had the opportunity to repeat the experiments presented in this research.

Firstly, we would have started the project earlier. Although we were able to complete the experiments within the given time frame, we had to rush certain aspects of the project, such as experimenting on Great Lakes due to constrained computing resources. Starting the project earlier would have given us more time to carefully design and execute each step of the project.

Secondly, we would have sought to obtain a larger dataset. While we were able to collect a sufficient amount of data for our experiments on open-source platform such as GitHub and Hugging Face, a larger and diverse dataset would have allowed us to explore more complex models and conduct more extensive analysis.

Thirdly, we would have explored more methods for data augmentation. Data augmentation is an effective technique for increasing the size and diversity of a dataset. Although we used some basic data augmentation techniques in our experiments, we could have explored more advanced techniques such as style transfer.

References

Aaditya Bhat. 2023. Gpt-wiki-intro (revision 0e458f5).

Arian Askari, Mohammad Aliannejadi, Evangelos Kanoulas, and Suzan Verberne. 2023. Chatgpt-retrievalqa: A dataset for training and evaluating question answering (qa) retrieval models on chatgpt responses.

Som Biswas. 2023. Chatgpt and the future of medical writing.

Michael Dowling and Brian Lucey. 2023. Chatgpt for (finance) research: The bananarama conjecture. *Finance Research Letters*, page 103662.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. TweepFake: About detecting deepfake tweets. *PLOS ONE*, 16(5):e0251415.

W. N. Francis and H. Kucera. 1979. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. Mgtbench: Benchmarking machine-generated text detection.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V. S. Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is chatgpt a good translator? a preliminary study. *arXiv preprint arXiv:2301.08745*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature.

OpenAI. 2022. ChatGPT: Optimizing language models for dialogue.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models.

Teo Susnjak. 2022. Chatgpt: The end of online exam integrity?

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news.

Terry Yue Zhuo, Yujin Huang, Chunyang Chen, and Zhenchang Xing. 2023. Exploring ai ethics of chatgpt: A diagnostic analysis.