Classifying AI LLM vs Human-Written Texts

Bryce Hiraoka
Boston University, CS 506
bhiraoka@bu.edu

1 Introduction

For my 506 project, I chose to tackle the challenge of detecting human-written text versus text generated by large language models (LLMs). I worked with six datasets: five sourced from Kaggle (LLM-Detect-AI-Generated-Text, PaLm-Generated-Essays, Combined-Set, AI-vs-Human-Text, and Human-vs-LLM-Corpus-Bloom-7B-and-GPT) and one from Hugging Face (AI-Text-Detection-Pile). To build the initial model, I combined multiple datasets to cover a diverse range of topics and LLMs, ensuring comprehensive detection capabilities. However, for subsequent models, I focused on a single dataset to achieve more refined and specific results.

Github Link:

1.1 Method & Relevant Results

I chose logistic regression for this project due to its effectiveness in binary classification and its widespread use in similar research problems. Using a single dataset containing 64,000 texts and five distinct features, one of my models achieved an overall accuracy of 80

1.2 Why is AI detection important?

Distinguishing between human-written and LLM-generated text has become increasingly important as AI continues to integrate into various aspects of our daily lives. AI detection is relevant to everyone, from children in schools to the elderly consuming news. One key application of AI detection is protecting readers from misinformation. Since free access to large language models like ChatGPT became available, the number of LLM-generated fake articles has risen by over 1000 percent [1]. With the growing prevalence of AI-generated fake news, the number of people impacted by such misinformation has likely increased as well. Implementing an AI detection model to alert readers about potentially inaccurate LLM-generated news could help protect the public and ensure they are better informed.

2 Related Work (Model and Features)

2.1 Decision Trees (Model)

The key distinction between logistic regression and decision trees lies in how they fit to data. Decision trees partition the data into progressively smaller regions, while logistic regression fits a single line to separate the space into two categories. I chose logistic regression over decision trees because it is more efficient to train, particularly for binary classification problems like mine. Decision trees excel at handling complex, non-linear relationships, but logistic regression is better suited for predicting binary outcomes. Ultimately, logistic regression aligned more closely with both my project's objectives and the training dataset I used.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

2.2 Support Vector Machines (Model)

The main difference between support vector machines (SVMs) and logistic regression is that SVMs use support vectors to identify the optimal dividing line between data points. While the two approaches are not vastly different, I chose logistic regression because it aligns better with the typical NLP features outlined in existing research.

2.3 Neural Networks (Model)

The main difference between neural networks and logistic regression is that logistic regression forms the foundation of a neural network, representing a single layer with one decision boundary, whereas neural networks can have multiple layers and decision boundaries. I opted not to use neural networks due to their complexity and the lack of necessary hardware to train the model efficiently. Additionally, logistic regression could theoretically achieve comparable results for this problem without the added complexity.

2.4 Uppercase Word Count (Feature)

Some popular AI detectors use uppercase word count to predict whether or not a piece of text is human or LLM generated. I decided not to include this feature in my model based on the previous experiments in "How to Detect AI-Generated Texts?" [3]

2.5 Number of Parts of Speech (Feature)

Some popular AI detectors use the number of different parts of speech (nouns, verbs, adjectives, etc.) to identify LLM generated text. I decided not to include this feature in my model based on the previous experiments in "How to Detect AI-Generated Text?" [3]

2.6 Readability (Feature)

I do have a readability score as one of my features however, I only use the Coleman Liau score to determine its readability. There are many other readability scores including but not limited to Flesch, Gunning Fog, Dale Chall, etc. The reason I decided to use Coleman Liau is it was the most accurate in deciphering whether a text was written by an LLM or a human. This was also based on the experiments outlined in "How to Detect AI-Generated Texts?" [3]

3 Resources

3.1 Hardware

Laptop: Apple Macbook M1 Pro 2021 CPU: Apple M1 (10 Cores) Memory: 32 GB unified memory (shared between GPU and CPU)

All programming was done in VSCode in Python, code being run on .ipynb files, pandas dataframe manipulation and sklearn logistic regression ran on CPU.

3.2 Software

Python Libraries: NumPy, Matplotlib, Pandas, Seaborn, textwrap, NLTK, collections, textstat, sci-kit learn, language-tool-python, datasets

LLM Tools Used: GitHub Copilot for writing repetitive code blocks i.e. generating graphs/statistics, Chat-GPT for generating custom test cases

3.3 Datasets

AI-Text-Detection-Pile (Hugging Face) - 1.418m [990k:340k]

`https://huggingface.co/datasets/artem9k/ai-text-detection-pile/viewer/default/train?p=5`

LLM-Detect-AI-Generated-Text (Kaggle) - 27k [17k:11k]

`https://www.kaggle.com/datasets/sunilthite/llm-detect-ai-generated-text-dataset`

PaLm-Generated-Essays (Kaggle) - 1.3k [0:1.3k]

`https://www.kaggle.com/datasets/kingki19/llm-generated-essay-using-palm-from-google-gen-ai`

Combined-Set (Kaggle) - 87k [55k:32k]

`https://www.kaggle.com/datasets/jdragonxherrera/augmented-data-for-llm-detect-ai-generated-text`

AI-vs-Human-Text (Kaggle) - 500k [305k:195k]

`https://www.kaggle.com/datasets/shanegerami/ai-vs-human-text`

Human-vs-LLM-Corpus-Bloom-7B-and-GPT (Kaggle) - 800k [360k:440k]

`https://www.kaggle.com/datasets/starblasters8/human-vs-llm-text-corpus`

Total Dataset Distribution [1.73m:1m] [Human:AI]

4 Methods

4.1 Featurization

I used 5 standard NLP features that have the largest influence on a logistic regression binary classifier for LLM vs Human written text. [3] [2]

The following describes how each feature was calculated in python. Everything was done within one function, and applied to the dataset's dataframe using df.apply().

The text is also tokenized, where special characters, linking words, and stop words are removed. This is what specifies token vs word in the following.

4.1.1 Coleman Liau Index (Readability)

Using the textstat library, calculate the readability of the given text.

4.1.2 Word Density

Divide the number of characters by the number of words in the given text.

4.1.3 Matches (Grammatical Errors)

Using the language-tool-python library, count the number of grammatical errors in the given text.

4.1.4 Title Word Count

Count the number of tokens that start a sentence (title words) in the given text.

4.1.5 Text Words (Text Length)

The number of words in the given text.

4.2 Logistic Regression

Logistic regression is a type of regression that is often used to predict classes (typically binary like in this case). It does so by predicting the probability of a certain outcome (or class) based on predictor variables. The reason it's called "logistic" regression is due to its use of the logistic or "sigmoid" function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

where z is the linear combination of the predictor variables and their coefficients:

$$z = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n$$

$\sigma(z)$ represents the probability that the dependent variable y belongs to the class 1 (the positive class, which in this case is LLM generated).

Given these two equations, this is the function that is minimized in logistic regression (logistic loss) [4]:

$$f(w) = -\frac{1}{n}\sum_{i=1}^{n}\log(1 + \exp(-y_i(w^T x_i))) + \frac{\lambda}{2}\|w\|^2$$

where: w represents the weights, $\frac{\lambda}{2}|w|^2$ is the l2 regularization term, and $\log(1 + \exp(-y_i(w^T x_i)))$ is the logistic loss for each point (x_i, y_i) where x_i is the feature vector and y_i is the actual class. $\frac{1}{n}$ is done to get the mean loss across the entire dataset.

During training, the model manipulates the coefficients w to minimize the equation.

The model then predicts the new sample's class based on this sigmoid function using the calculated weights:

$$\hat{y} = \sigma(w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n)$$

where $\hat{y}$ is the predicted probability and $x_1, x_2, \ldots, x_n$ are the individual features (not the feature vector) of the new sample.

5 Experimental Results

5.0.1 Dataset Selection Methodology

I trained three separate models using three separate datasets:

- (1) A random sample of 10000 texts from a concatenated set of all datasets listed previously (LLM-Detect-AI-Generated-Text, PaLm-Generated-Essays, Combined-Set, AI-vs-Human-Text, Human-vs-LLM-Corpus-Bloom-7B-and-GPT, AI-Text-Detection-Pile). Around 6500 essays were human written, 3500 were LLM written. This ratio is a result of the ratio of human to LLM written texts found in the total concatenated data set of around 3.1m texts.
- (2) 20000 texts, evenly split between human and LLM-written text, randomly sampled from the AI-Text-Detection-Pile dataset.
- (3) 64000 texts, evenly split between human and LLM-written text, randomly sampled from the Combined-Set dataset.

The models were trained in the given order, based off of intuition gained from the previous one. The first model was trained with a portion of the total concatenated dataset (due to computing power constraints). The second model was trained on the largest individual dataset, and the third model was trained on an already curated dataset.

5.0.2 Parameters

Each model was evaluated with a train-test split of 3:2 using the scikit-learn python library's built-in Logistic Regression function initialized with default parameters, with random-state set to 42 (arbitrary) to maintain consistency between runs.

Briefly, the default parameters are the use of the L2 penalty term, a stopping tolerance of $1e^{-}4$, a regularization value $C = 1.0$, 100 max iterations, a class weight of 1 for both classes (Human vs LLM), and the Limited-memory BFGS (LBFGS) solver to optimize.

From testing, changing the parameters made little tangible difference, so everything was left as default.

5.1 All datasets 6500:3500

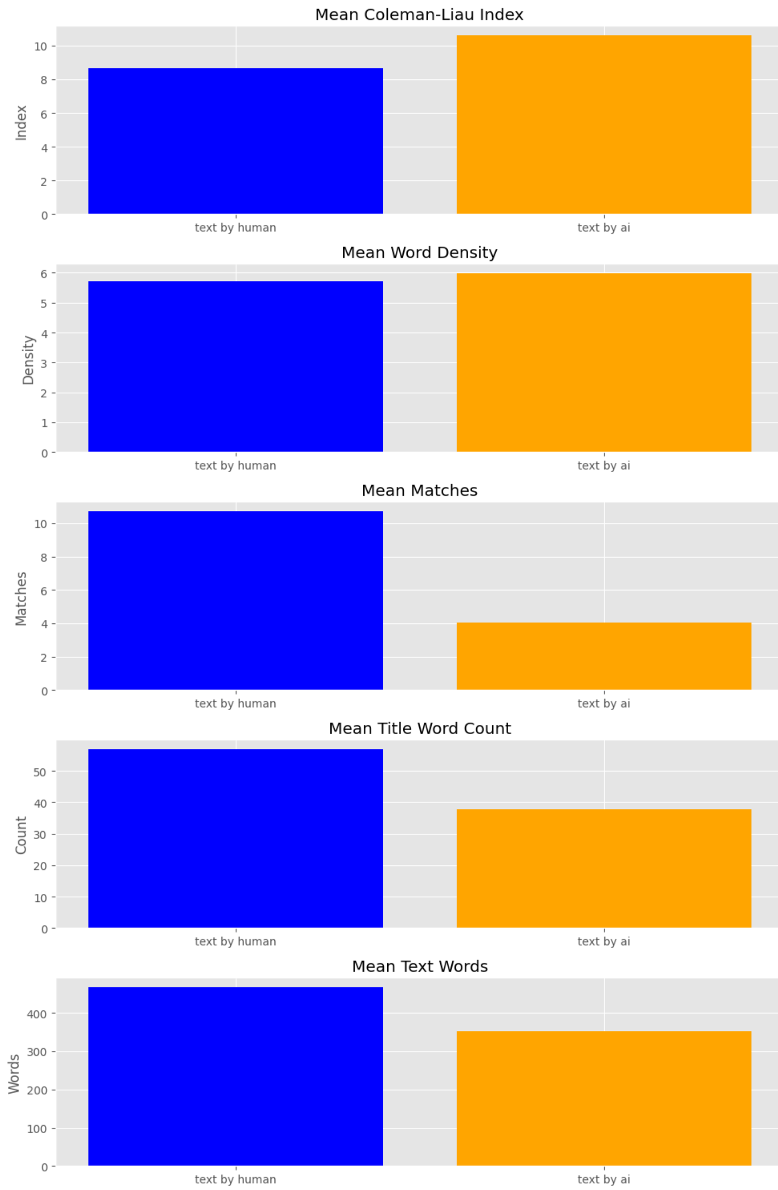

Figure 1: Mean Feature Values for all Datasets

Table 1: Logistic Regression Model w/ All Datasets Metrics

Accuracy: 0.731625
Train Loss: 0.5807722830077537
Test Loss: 0.5770358406982319

Name	Precision	Recall	f1-Score	Support
0 (Human)	0.736433	0. 894161	0.807668	25208
1 (LLM)	0.715959	0.454638	0.556130	14792
Macro Avg	0.726196	0.674399	0.681899	40000
Weight Avg	0.728862	0.731625	0.714649	40000

Table 2: Logistic Regression Model w/ AI-Text-Detection-Pile Dataset Metrics

Accuracy: 0.678875
Train Loss: 0.6397425870016508
Test Loss: 0.6450721579241364

Name	Precision	Recall	f1-Score	Support
0 (Human)	0.707965	0.615996	0.658786	4026
1 (LLM)	0.656215	0.742577	0.696730	3974
Macro Avg	0.682090	0.679286	0.677758	8000
Weight Avg	0.682258	0.678875	0.677635	8000

5.2 AI-Text-Detection-Pile 10000:10000

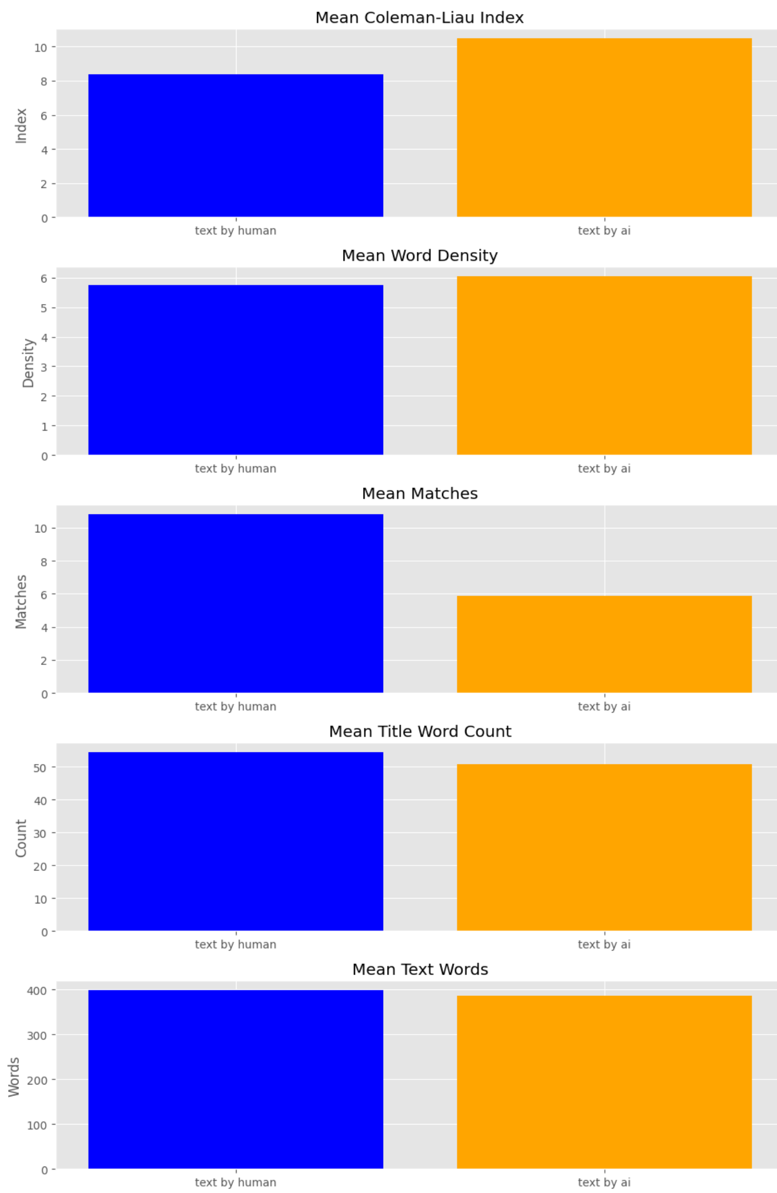

Figure 2: Mean Feature Values for AI-Text-Detection-Pile dataset

Table 3: Logistic Regression Model w/ Combined-Set Dataset Metrics

Accuracy: 0.8012890625 Train Loss: 0.42126754359148816 Test Loss: 0.42277091258149474				
Name	Precision	Recall	f1-Score	Support
0 (Human)	0.799938	0.803127	0.801529	12790
1 (LLM)	0.802649	0.799454	0.801048	12810
Macro Avg	0.801293	0.801290	0.801289	25600
Weight Avg	0.801294	0.801289	0.801289	25600

5.3 Combined-Set 32000:32000

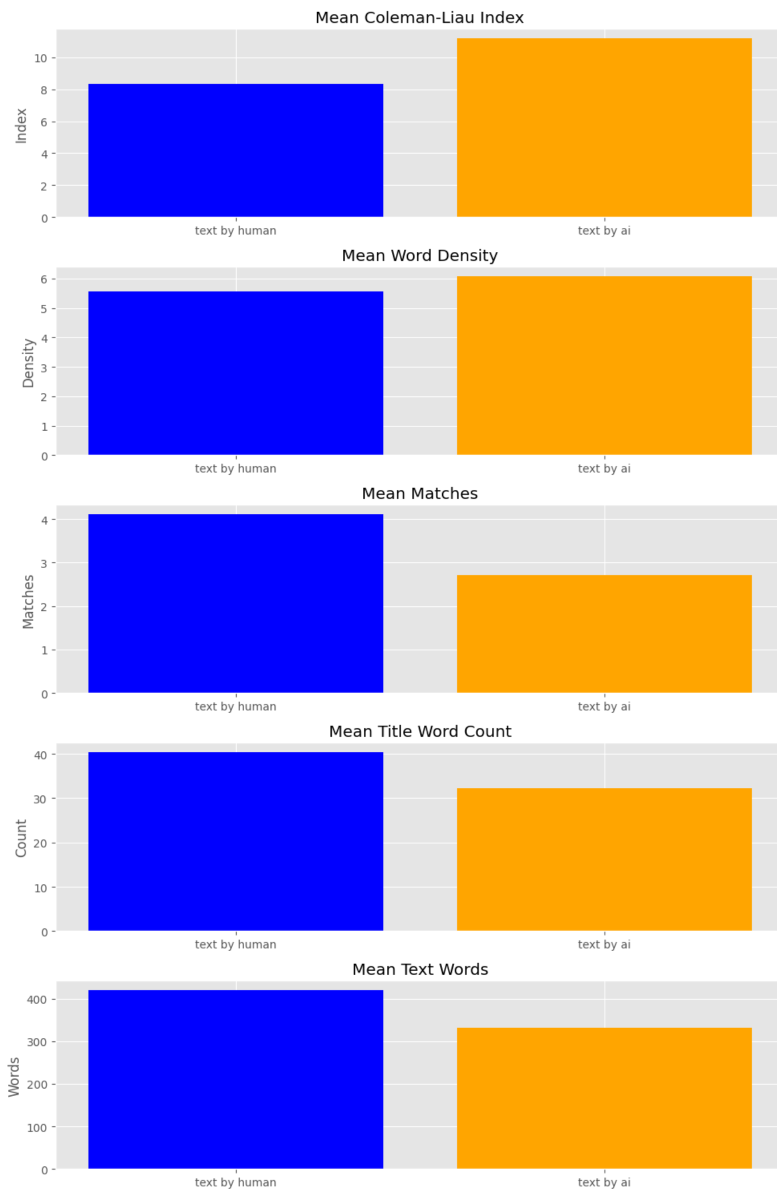

Figure 3: Mean Feature Values for Combined-Set

5.4 Results Analysis

5.4.1 Dataset Influence

Based purely on the mean, all 5 features seem to have a noticeable influence on whether the text is AI-generated or Human-generated.

The worst performing model in terms of accuracy, the model trained on the AI-Text-Detection-Pile, had the least differentiation between the mean for all 5 features.

The best-performing model, on the other hand, had the greatest differentiation between the mean for all 5 features.

Throughout all 3 models, however, the difference was the same: AI text had a higher readability score, greater word density, fewer grammatical errors, fewer title words (and therefore sentences), and shorter length. The text length is only present ultimately to determine the ratio between itself and the number of errors and title words, so just observing the mean in this way doesn't indicate anything. However, given the clear differences they indicate, the model will pick up on its influence on if the text is LLM or human-written.

Undersampling was used for the second and third datasets to attempt to rectify the first model's poor accuracy.

5.4.2 Accuracy and Loss

The most accurate model with the least loss was the last model trained on the Combined-Set dataset, potentially due to 3 reasons:

(1) The given dataset was curated already for a text detection competition, and may have already been optimized for such a task

(2) The number of texts used was the largest, 64000 compared to 20000 and 10000. The first two models were most likely underfitting the data.

(3) The use of undersampling compared to the first dataset to prevent drastic differences in accuracy in classifying both texts.

Due to time and computational power constraints, no further testing could be conducted to see how each of these reasons directly influenced the accuracy, but a combination of the 3 in training future models would likely yield better results.

6 Conclusion

One approach that can be taken in the future is to use datasets with text generated exclusively by one LLM, as this would prevent the model from needing to compare human-written text to text written by several LLMs, which likely all have their own differences in features. This obviously reduces the use case of the individual model, but when used in conjunction with multiple models, one might yield better results. In addition to training on a larger dataset, this is what I believe to be the most effective technique based on my results.

References

[1] The washington post, Dec 2023.

[2] Lorenz Mindner, Tim Schlippe, and Kristina Schaaff. Classification of human- and ai-generated texts: Investigating features for chatgpt. In Tim Schlippe, Eric C. K. Cheng, and Tianchong Wang, editors, *Artificial Intelligence in Education Technologies: New Development and Innovative Practices*, pages 152–170, Singapore, 2023. Springer Nature Singapore.

[3] Trung Nguyen, Amartya Hatua, and Andrew Sung. How to detect ai-generated texts? pages 0464–0471, 10 2023.

[4] Mark Schmidt. Tutorial 8 logistic regression and stochastic gradient descent.