A Hybrid Stylometry-Transformer Approach for AI Text Detection

Rohith Venkatesh, Rishi Padmanabhan, Shayan Ravari
Department of Computer Science
University of California, Los Angeles
{rvenkatesh2025, rpadmanabhan20, shayanravari}@g.ucla.edu

Abstract

This document is our final report. It continues from our mid-project report, where we described our modeling approach, training data we gathered for this project, and results (at the time). At that time, we had implemented a Naive Bayes baseline and an initial BERT-based classifier. The models were trained on a dataset comprising of human-written and GPT-2 generated text. Since then, we have fine tuned our model through feature engineering, achieving higher accuracy. Furthermore, we introduce a hybrid model which builds upon our pre-existing BERT model. This uses the stylometric feature data from any given corpus to make stronger predictions. Finally, we tested our model on an ethics development set to see if our model was biased toward proficient English data.

1 Introduction

The recent advent and widespread accessibility of Large Language Models such as Google's Gemini 2.5 and Meta's Llama series have fundamentally altered the landscape of digital content creation. These models provide unprecedented capabilities for summarizing information and assisting in creative writing. However, this technological leap has introduced several societal challenges, the most notable one being distinguishing between human and machine-generated text. This ambiguity poses critical risks to academic integrity and the overall trust in online information (Gehrmann et al., 2019). Consequently, the development of reliable AI detection systems is a necessity for maintaining a transparent digital ecosystem.

Early approaches to text classification relied on statistical methods, often treating documents as a bag-of-words. Models like Naive Bayes analyze the frequency of words (n-grams) to make predictions, a method that, while effective, largely ignores the crucial role of word order and context. The modern paradigm for text understanding is dominated by transformer architectures (Vaswani et al., 2017), the most predominant being BERT (Devlin et al., 2018) and its variants. The core innovation of BERT is its ability to generate deep contextual embeddings. Through its multi-layer self-attention mechanism, BERT examines the entire input sequence from both directions, allowing it to understand that the meaning of a word is defined by the words surrounding it.

In this project, we aim to create an AI detection system that can evaluate a hidden test set of human-written and AI-generated text. The system should be able to take in a text and see if it is AI or human written. As LLMs become integrated with academia and industry, proetecting human work becomes more important. Our project aims to create a solution for protecting academic and workmanship integrity.

2 Modeling Approach

2.1 Naive Bayes Baseline

We began with a simple Multinomial Naive Bayes classifier to set a performance baseline. This model was chosen for its simplicity, computational efficiency, and effectiveness in text classification tasks (Manning et al., 2008). The core assumption of Naive Bayes is the conditional independence of features, given the class. It calculates the probability of document D belonging to a class c using Bayes' Theorem:

$$\Pr(c|D) \propto \Pr(c) \prod_{i=1}^{n} \Pr(w_i|c) \tag{1}$$

Here, $\Pr(c|D)$ is the posterior probability we aim to find. $\Pr(c)$ is the prior probability of the class, calculated from its frequency in the training data. And $\Pr(w_i|c)$ represents the likelihood of word w_i appearing in a document of class c. For our specific application, we used Laplace smoothing with

$\alpha = 0.1$ to prevent the zero-frequency problem which would make the entire probability product zero. Moreover, we used the Scikit-learn library (Pedregosa et al., 2012), and represented documents using TF-IDF vectors derived from unigrams and bigrams, which provides a more useful feature representation than raw word counts.

2.2 Fine-tuned BERT Classifier

Our second, more powerful baseline is a fine-tuned BERT model. A Transformer Encoder consists of two main sub-layers: a multi-head self-attention mechanism and a position-wise feed-forward network. Let the input to a layer be a matrix of word embeddings $X \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the embedding dimension. Instead of performing a single attention function, the model linearly projects the input embeddings into h different heads, allowing it to jointly attend to information from different representation subspaces (Devlin et al., 2018). For each head i, the input X is projected into query, key, and value matrices: $Q_i = XW_i^Q$, $K_i = XW_i^K$, and $V_i = XW_i^V$, where W_i^Q, W_i^K, and W_i^V are learned weight matrices. The attention for a single head is then calculated:

$$\text{head}_i = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \qquad (2)$$

where d_k is the dimension of the key vectors. The outputs of all heads are concatenated and linearly projected to produce the final attention output, which is then passed through a residual connection and layer normalization. This output is then used as the input to a simple two-layer fully connected network:

$$\text{FFN}(x) = \max\{0, W_1 x + b_1\} W_2 + b_2 \qquad (3)$$

where W_1, W_2, b_1 and b_2 are the respective weight matrices and bias vectors. Once again, this is followed by a residual connection and layer normalization. The specific model that we used was `bert-base-uncased`, which consists of twelve stacked Transformer Encoder blocks. For our task, we add a simple classification head on top of the final layer's output. Specifically, the final hidden state corresponding to the special `[CLS]` token, denoted $C \in \mathbb{R}^d$, is used as an aggregate sequence representation. This vector is passed through a linear layer with learned weights $W_{cls} \in \mathbb{R}^{k \times d}$ (where $k = 2$ for our binary task) and a softmax function to produce a probability distribution over the classes:

$$\Pr(c|D) = \text{softmax}\left(CW_{cls}^T\right) \qquad (4)$$

During fine-tuning, the entire model, including the pre-trained BERT weights and the new classification layer, is trained to minimize a cross-entropy loss function on a split of data from the two training datasets we collected.

3 Hybrid Model

3.1 Stylometric Feature Engineering

While transformer models implicitly learn some stylistic patterns, we developed a set of explicit features to directly quantify the statistical properties of the writing style. These features are designed to capture artifacts commonly associated with machine-generated text. Using the `textstat` library (Bansal and Aggarwal, 2019), we created an eleven-dimension feature vector, $\mathbf{x}_{\text{style}} \in \mathbb{R}^{11}$, for each document, targeting three key areas:

Readability: We included five standard readability scores (e.g., Flesch Reading Ease, Gunning Fog). Our hypothesis was that AI-generated text might exhibit less complexity and a more uniform readability compared to the wider variance found in human writing.

Lexical Diversity: We measured lexical diversity using the Type-Token Ratio (TTR). We also included the raw lexicon count as a basic feature.

Syntactic Structure: Features like sentence count and average sentence length were included to capture sentence construction patterns.

3.2 Linear Concatenation

For a given document, we first extracted its 768-dimensional `[CLS]` token embedding from our fine-tuned BERT model, denoted as $\mathbf{x}_{\text{BERT}} \in \mathbb{R}^{768}$. This vector was then concatenated with its corresponding eleven-dimensional stylometry vector $\mathbf{x}_{\text{style}}$ to form a single hybrid feature vector $\mathbf{x}_{\text{hybrid}} = \mathbf{x}_{\text{BERT}} \oplus \mathbf{x}_{\text{style}} \in \mathbb{R}^{779}$. During the first iterations of development, this hybrid vector was used as input to a standard Logistic Regression classifier. The classifier models the log-odds of the positive class (AI) as a linear function of the input features.

Counterintuitively, this linear hybrid model performed significantly worse than the BERT-only

Model	Acc.	Prec.	Rec.	F1
Mid-Project	*0.647*	*0.703*	*0.469*	*0.494*
Naive Bayes	0.5611	0.5691	0.3321	0.3715
BERT	0.9085	0.9732	0.8096	0.8812
Hybrid	**0.9273**	**0.9531**	**0.8754**	**0.9116**

(a) Average performance across four development subsets

Dev Set	Naive Bayes	BERT	Hybrid
arxiv_chatGPT	0.1336	0.9551	0.9507
arxiv_cohere	0.5249	0.8744	0.8916
reddit_chatGPT	0.3863	0.8615	0.9205
reddit_cohere	0.4413	0.8336	0.8837

(b) F1-Score breakdown by development subset

classifier. After applying `StandardScaler` to the features, the absolute magnitudes of the coefficients corresponding to the eleven stylometric features were disproportionately larger than those for the 768 BERT features. This suggested to us that the linear model over-relied on the low-dimensional, high-variance stylistic features. It was unable to properly balance the two feature sets, effectively treating the high-dimensional BERT signal as less important. Thus, we concluded that the model's linearity was its primary limitation. Once we replaced the Logistic Regression classifier with a Multi-Layer Perceptron (MLP) there was a major improvement in the model's performance.

4 Datasets

These are the data sets that we collected:

- real-vs-gpt2-sentences
- HC3 (Human ChatGPT Comparison Corpus)

Our project utilizes two stylometry-relevant datasets sourced from Hugging Face to explore the distinction between human- and AI-generated text (same as discussed in the mid-project report). The first dataset, real-vs-gpt2-sentences, contains human-written and GPT-2-generated sentences. The second dataset, derived from the HC3 corpus (specifically the ELI5 subset) (**?**), includes human written and ChatGPT-generated answers. Together, these datasets provide a balanced corpus of $1,500$ human and $1,500$ AI texts.

Before modeling, all data was preprocessed by converting text to lowercase and removing HTML tags to reduce noise. The combined dataset of $3,000$ samples was then split into training and testing subsets using stratified sampling to maintain label balance, with 80% ($2,400$ samples) allocated for training and 20% (600 samples) for testing.

The development dataset is the same as that provided by the professor. Our project also utilized an ethics developmental set (not included in the mid-project report). The ethics development set includes german text, essays written by non-native English speakers, and essays written by native English speakers. The set aims to address social biases in LLM training data. All of the code for this project can be accessed from the following Github repository: github.com/shayanravari/CS-162-Final-Project.

5 Results and Analysis

We evaluated our three models on the four subsets of the development set. The evaluation remained consistent across all models, using accuracy, precision, recall, and F1-score as the primary metrics for the positive class (AI).

5.1 Quantitative Performance

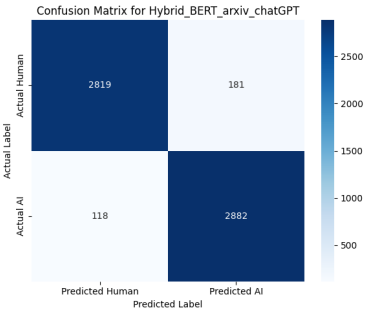
The impact of our improvements was substantial. As shown in Table 1a, our final models represent a dramatic leap in performance over the mid-project baseline. The fine-tuned BERT model, trained on our two training datasets, achieved an average accuracy of over 92%.

Our final hybrid model provided a further, significant boost, achieving a final average F1-score of **0.9116**. This confirms that a properly developed non-linear fusion model can effectively use stylistic features to enhance a strong semantic baseline. Table 1b details this strong performance across the different development subsets, showing consistent gains in comparison to the BERT model, particularly on the `reddit_cohere` dataset.

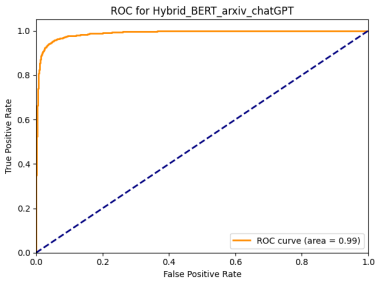
5.2 Error Analysis

Despite the high overall accuracy, an analysis of the hybrid model's misclassifications reveals remaining challenges. The errors primarily fall into two categories:

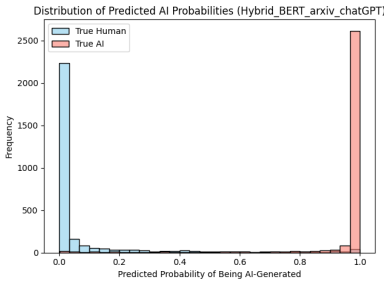
- **False Negatives (AI as Human):** The model struggles most with extremely short or generic AI-generated texts. For example, a response like *"That's a great question, I'll need to think about that"* lacks sufficient semantic or stylis-



(a) Hybrid Confusion Matrix



(b) Hybrid ROC Curve



(c) Hybrid Probability Distribution

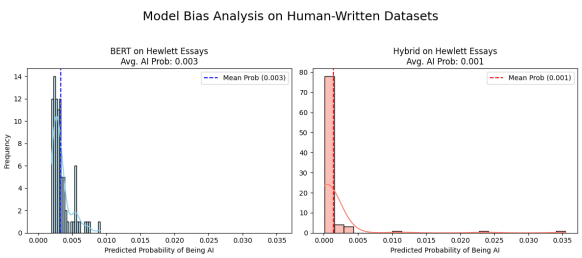
tic content for the model to find any distinguishing artifacts.

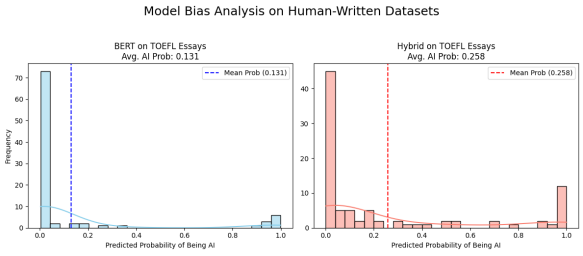
- **False Positives (Human as AI):** Conversely, human-written text that is highly structured and devoid of personal opinion, such as technical abstracts from arXiv or legal text, can sometimes be misclassified as AI-generated. The model seems to associate this objective tone with LLM outputs it has seen during training.

This suggests that while our model is highly effective on general text, texts at the extremes of brevity or formality represent a general difficulty that AI detection systems seem to experience. This points towards future work in models that can better handle low-signal inputs or understand domain-specific human writing styles.

5.3 Evaluation on Ethics Development Set

When testing on the three new ethics datasets, the results were mixed. For the `german_wikipedia` dataset, both the BERT and hybrid model struggled immensely, achieving an accuracy of only 50.1%, which is essentially the same as random guessing. The baseline classifier scored slightly higher with an accuracy of 54.3%. The primary culprit for the low accuracies was the training data that we used. Since all of our training data was in English, the models only understand how to classify in English and not in German.





The prior two figures show the analysis of our model on the Hewlett Essays and TOEFL Essays. The Hewlett Essays consisted of essays written by English speakers, while the TOEFL Essays consisted of essays written by non-native English speakers. The analysis shows that both of our models (the BERT and Hybrid models) performed well on the sets, correctly identifying both datasets as human-written. The BERT and Hybrid models identified the Hewlett Essays as being AI generated with a mean probability of 0.003 and 0.001, respectively, while they identified the TOEFL Essays as AI generated with a mean probability of 0.131 and 0.258, respectively. Since the probabilities are low (< 0.5), we can say that the models performed well, and the model is not biased towards English proficiency. Surprisingly, the BERT model performed better than the Hybrid model on the TOEFL Essays, showing that the Hybrid model may not always be the best choice.

6 Conclusion

Our project illustrates the potential of different modeling approaches for distinguishing between human and AI-generated text. By combining the contextual depth of BERT embeddings with stylometric features in the hybrid model, we achieved strong classification performance, with the hybrid model outperforming both the Naive Bayes and BERT-only baselines across the 4 development subsets.

However, our evaluation on the ethics devel-

opment set also reveals some limitations. The models struggled on non-English text (the German Wikipedia entries). This performance gap highlights the need for more diverse training data that includes samples from all languages. In future work, expanding multilingual capabilities, by including datasets from the most spoken languages, would help our models perform better with all languages, allowing our models to be used by a larger demographic.

Despite the setbacks with the German Wikipedia entries, our models performed well on the Hewlett and TOEFL essays, showing that English proficiency is not a factor in our models' performance. This opens the door for our model to be used in non-native English environments, like ESL (English as a Second Language) programs.

Some applications of our models include anti-cheating software in academia. As LLMs become more integrated with classrooms, the potential for academic dishonesty increases. Our model could be used in conjunction with educational platforms (such as Grammarly or Turnitin) to check for AI usage. Our model could also be used in industry to certify that a person's work is their own and not AI generated, ensuring an equal opportunity environment for professionals.

References

Shivam Bansal and Chirag Aggarwal. 2019. textstat: A python library to calculate readability scores. https://github.com/textstat/textstat.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. 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 62–67, Florence, Italy. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2012. Scikit-learn: Machine learning in python. *CoRR*, abs/1201.0490.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.