SENTINEL: A Heterogeneous Ensemble Framework for Detecting AI-Generated Text in the Era of Advanced Language Models

Natalie Cao
Department of MS & E
Stanford University
nc323@stanford.edu

Haocheng Fan
Department of C & EE
Stanford University
hfan11@stanford.edu

CS 224N Custom Project **TA Mentor:** Shijia Yang

Abstract

Large language models (LLMs) like GPT-4 and GPT-4o are developing so quickly that it is becoming more difficult to discern material created by AI from that produced by humans. In order to efficiently recognize text generated by artificial intelligence, this research introduces SENTINEL, a revolutionary heterogeneous ensemble architecture that integrates deep learning models with traditional machine learning methods. Our contributions include of an optimized ensemble architecture, a hybrid feature extraction approach, and the creation of tailored preprocessing tools. In addition, we created a dynamic dataset with cutting-edge LLMs to guarantee SENTINEL's scalability. Experimental results demonstrate that SENTINEL performs much better than the baseline on our curated dataset as well as the original testing dataset from Kaggle. The integration of interpretability techniques enhances the model's transparency. SENTINEL represents a significant advancement in the detection of AI-generated text, providing a robust solution to the challenges posed by the rapid evolution of LLMs.

1 Introduction

Natural language processing has undergone a revolution attributed to the quick development of large language models (LLMs) like GPT-4 and GPT-4o, which produce remarkably coherent and contextually relevant text [1]. But this amazing development has also made it more difficult to discern text created by AI from that written by humans [2]. These models' ability to generate text that closely resembles the writing styles of humans has sparked worries about potential misuse in various domains, including academic integrity[3], fake news propagation[4], and social media manipulation [5]. Traditional methods for identifying text produced by AI, such statistical analysis [6] and stylometric techniques [7], frequently fall short in capturing the intricate linguistic patterns and subtle contextual cues seen in the output of cutting-edge LLMs [8]. Furthermore, because these models are evolving so quickly, it is necessary to create detection frameworks that are flexible and scalable in order to keep up with the rapidly changing field of natural language creation.

To address these challenges, we propose **SENTINEL**, short for **S**ophisticated **EN**semble for **T**racing and **I**dentifying **N**on-**A**uthentic **L**anguage, a heterogeneous ensemble framework that integrates advanced machine learning techniques with classical algorithms. This hybrid system aims to effectively distinguish between human and machine-generated texts, leveraging the nuances presented by state-of-the-art LLMs like GPT-4 and GPT-4o. The key contributions of our work include: **1.** A custom preprocessing pipeline that efficiently handles various text formats and noise, ensuring optimal input for the subsequent stages of the framework; **2.** A hybrid feature extraction method that amalgamates deep contextual embeddings from state-of-the-art language models with traditional n-gram features enables the capture of both high-level semantic information and fine-grained linguistic patterns; **3.** An optimized ensemble architecture that employs a soft voting mechanism, adaptively weighting the contributions of individual models based on their performance and reliability; **4.** A comprehensive evaluation of SENTINEL on a newly created diverse range of datasets, including a dynamic corpus of over 10,000 entries generated by state-of-the-art LLMs including GPT-4 and GPT-4o, demonstrated its superior performance and adaptability compared to existing baselines.

The experimental results show fundamental statistical gains in accuracy and F-1 score, as well as much better experimental outcomes than the baseline state-of-the-art that we used. By providing a robust, adaptable, and interpretable solution, our framework aims to safeguard the integrity of information across various domains and ensure the responsible use of advanced language models.

2 Related Work

The rapid advancements in AI-generated text technologies have intensified the need for robust mechanisms capable of distinguishing between human-written and machine-generated content. Early detection methods relied heavily on statistical approaches and traditional machine learning models, such as the Multinomial Naive Bayes (MNB) classifier [9]. These methods, while effective for simpler tasks, struggled to keep pace with the increasing sophistication of AI-generated text due to their limited ability to understand complex contextual nuances and capture the nuances of language [10].

Recognizing the limitations of conventional models, the focus of research shifted towards more advanced deep learning techniques, particularly transformer-based models, which represent a significant evolution in the field. Solaiman et al. [2] underscored the efficacy of models like RoBERTa [11] in identifying texts produced by earlier generations of language models such as GPT-2. This pivotal research illuminated the profound capability of deep learning models to discern complex language patterns that elude simpler algorithms. Expanding on this groundwork, Uchendu et al. [12] and Zellers et al. [4] investigated the effectiveness of BERT [13] and GROVER, respectively, in both generating and detecting synthetic texts. These studies not only highlighted the dual utility of generative models but also showcased their potential in aiding the detection of AI-generated content. Nevertheless, these advanced models frequently require substantial computational resources and ongoing adjustments to remain effective against the rapidly evolving landscape of language technologies.

Amidst the limitations of single-model approaches, recent research has increasingly turned to ensemble methods that amalgamate multiple analytical techniques to heighten detection accuracy. Zhang et al. [14] explored the synergistic potential of combining gradient boosting machines with neural networks, advocating for heterogeneous ensembles. Their findings demonstrated that integrating diverse models could enhance the detection of various text characteristics, thereby improving overall system performance. Concurrently, stylometric analysis has gained prominence as an effective method for distinguishing AI-generated texts. Fagni et al. [7] highlighted how stylometric features could be leveraged alongside machine learning models to exploit distinct authorial styles in distinguishing human authors from artificial entities.

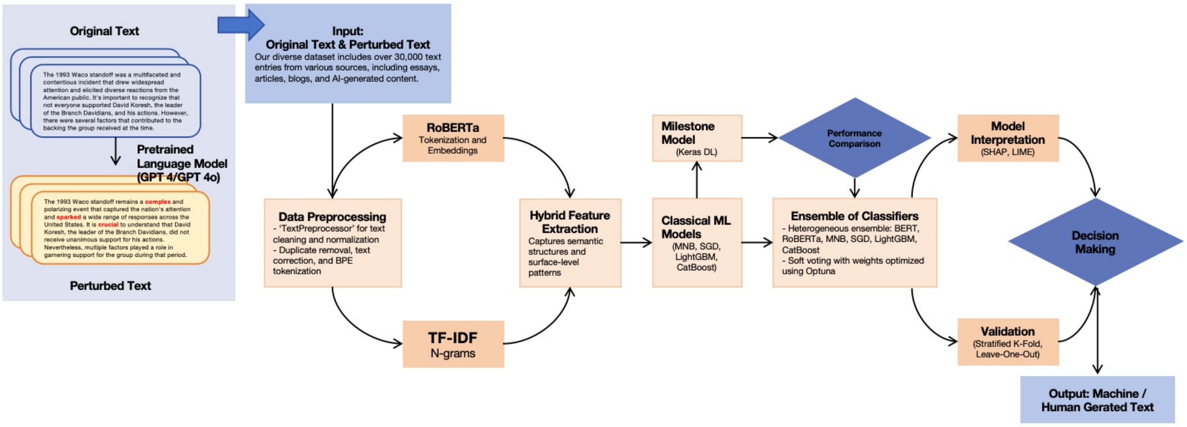
As the complexity of AI-generated text detection models has increased, so too has the emphasis on model interpretability and explainability. Wiegreffe et al. [15] emphasized the importance of developing models that are not only effective but also provide clear insights into their operational mechanisms. This transparency is crucial for fostering trust and enhancing the reliability of text detection systems. Despite the significant advancements in this field, existing studies often lack robustness against the continuously evolving capabilities of language models and fail to address the diversity of text types and contexts. Many approaches struggle to account for the subtle nuances that newer language models can replicate, leading to higher rates of false positives and false negatives [16]. Moreover, the majority of research has focused on English text, leaving the performance of detection models on other languages largely unexplored.

Our research builds upon these insights by integrating state-of-the-art language models with classical machine learning techniques within a unified ensemble framework. By combining deep learning, ensemble techniques, and stylometric analysis, along with advanced preprocessing and feature extraction methods, our approach offers a significant step forward in accurately identifying AI-generated text across diverse domains. The incorporation of interpretability techniques further enhances the understanding of the model's decision-making process, contributing to the development of more transparent and trustworthy AI-generated text detection systems.

3 Approach

SENTINEL employs a novel heterogeneous ensembling framework that synergistically combines the strengths of advanced deep learning models, such as BERT [13] and RoBERTa [11], with the efficiency and scalability of classical algorithms, including Multinomial Naive Bayes (MNB) [9], Stochastic Gradient Descent (SGD) [17], LightGBM [18], and CatBoost [19]. Figure 1 provides an overview of our proposed method.

Baseline We compare our approach to a gold-winning baseline from a relevant Kaggle competition[20], which demonstrated an accuracy of 0.83. The baseline model employs advanced deep learning techniques


Original Text
Pretrained Language Model (GPT 4/GPT 4o)
Perturbed Text
Input:
Original Text & Perturbed Text
RoBERTa
Tokenization and Embeddings
Data Preprocessing
- 'TextPreprocessor' for text cleaning and normalization
- Duplicate removal, text correction, and BPE tokenization
TF-IDF
N-grams
Hybrid Feature Extraction
Captures semantic structures and surface-level patterns
Classical ML Models
(MNB, SGD, LightGBM, CatBoost)
Milestone Model
(Keras DL)
Performance Comparison
Ensemble of Classifiers
- Heterogeneous ensemble: BERT, RoBERTa, MNB, SGD, LightGBM, CatBoost
- Soft voting with weights optimized using Optuna
Model Interpretation
(SHAP, LIME)
Validation
(Stratified K-Fold, Leave-One-Out)
Decision Making
Output: Machine / Human Gerated Text

The 1993 Waco standoff was a multifaceted and contentious incident that drew widespread attention and elicited diverse reactions from the American public. It's important to recognize that not everyone supported David Koresh, the leader of the Branch Davidians, and his actions. However, there were several factors that contributed to the backing the group received at the time.

The 1993 Waco standoff remains a **complex** and polarizing event that captured the nation's attention and **sparked** a wide range of responses across the United States. It is **crucial** to understand that David Koresh, the leader of the Branch Davidians, did not receive unanimous support for his actions. Nevertheless, multiple factors played a role in garnering support for the group during that period.

Our diverse dataset includes over 30,000 text entries from various sources, including essays, articles, blogs, and AI-generated content.

Figure 1: Overview of the Hybrid AI Text Detector method. Our proposed model architecture aims to fine-tune pre-trained language models for machine-generated text classification tasks.

such as the *DebertaV3Tokenizer* and *DebertaV3Preprocessor* from the keras-nlp library for tokenization and preprocessing, and utilizes the *DebertaV3Classifier*, a powerful deep learning architecture based on the DeBERTa model. By comparing our proposed framework with this established baseline, we aim to demonstrate the effectiveness and superiority of our approach in terms of accuracy, precision, and recall.

Data Preprocessing & Hybrid Feature Extraction (Original): We introduce a custom *TextPreprocessor* class to handle text cleaning and normalization. The preprocessor offers different strategies, such as 'none', 'light', or more aggressive cleaning, to adapt to the specific requirements of the text data. The preprocessed text is then tokenized using the RoBERTa tokenizer, which is well-suited for handling a wide range of text types and styles. We propose a novel *HybridFeatureExtractor* that combines RoBERTa-generated embeddings with TF-IDF n-grams, leveraging both deep contextual embeddings and traditional text features to enhance the model's ability to distinguish between human-written and AI-generated text across various domains. This hybrid feature extraction approach is an original contribution of our work.

Ensemble of Classifiers & Optimization (Original) The architecture of our model is built around a heterogeneous ensemble approach, combining the predictive power of several well-established machine learning models:

- Pre-trained Language Models (PLMs): We include BERT[13] and RoBERTa[11] utilizing for generating deep contextual embeddings, these transformer-based models are crucial for understanding the context and semantics embedded in texts, which helps differentiate AI-generated texts from human-written texts.
- Classical Machine Learning Models: Multinomial Naive Bayes (MNB)[9], Stochastic Gradient Descent (SGD)[17], LightGBM[18], and CatBoost[19]: These models are integrated to leverage their strengths in handling different aspects of text data, enhancing the ensemble's performance through their individual and combined predictive power.

Our ensemble employs a soft voting mechanism, where predictions from each model are weighted based on their performance and reliability. The weights for these models are optimized using a hyperparameter tuning framework powered by Optuna[21], ensuring optimal performance. To demonstrate mathematically, let $M = m_1, m_2, \ldots, m_n$ be the set of n classifiers in the ensemble.For a given input x, each classifier m_i predicts a probability $p_i(y|x)$ for each class y. The soft voting ensemble predicts the class $\hat{y}$ as:

$$\hat{y} =_y \sum_{i=1}^{n} w_i \cdot p_i(y|x) \tag{1}$$

where w_i is the weight assigned to classifier m_i.

Our model incorporates a heterogeneous ensemble of classifiers with tailored weights optimized through Bayesian Optimization and Optuna. We employ the AUC-ROC as our primary metric to gauge the model's proficiency in distinguishing between human and AI-generated texts across varied decision thresholds. The weights for the soft voting ensemble mechanism are determined by Bayesian Optimization to maximize the overall AUC-ROC score. AUC-ROC is the area under the curve plotted with TPR on the y-axis and FPR on the x-axis at different classification thresholds. To obtain the score, the True Positive Rate (TPR) and False Positive Rate (FPR) are calculated as:

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN} \tag{2}$$

Let $w = (w_1, w_2, \ldots, w_n)$ be the weights of the classifiers in the ensemble. The objective function $f(w)$ to be optimized is the AUC-ROC score of the ensemble on a validation set.

$$f(w) = \text{AUC-ROC}(\text{Ensemble}(w)) \tag{3}$$

Bayesian optimization aims to find the optimal weights w' that maximize the objective function $f(w)$. It builds a probabilistic model (e.g., Gaussian Process) of the objective function and uses an acquisition function (e.g., Expected Improvement) to guide the search for optimal weights.

$$w' = \text{argmax}_w f(w) \tag{4}$$

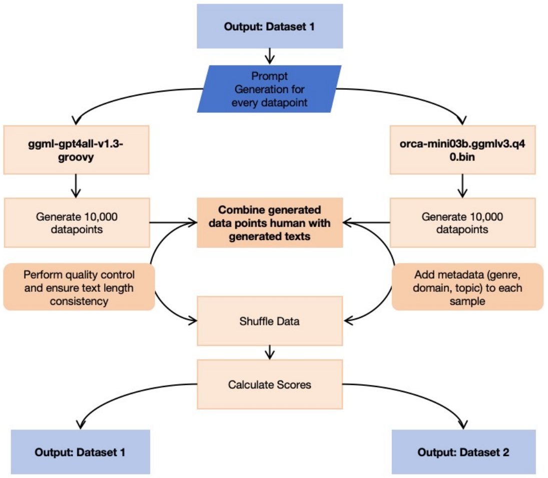
Evaluation & Model Interpretation (Original) Our evaluation framework utilizes multiple metrics to assess the efficacy of our AI-generated text detection models, including AUC-ROC, accuracy, F1 score, precision, and recall. We employ Stratified K-Fold and Leave-One-Out cross-validation techniques for robust validation and benchmark our approach against a gold-standard baseline from the Kaggle competition. To further interpret our model's predictions and gain insights into the features that contribute to the detection of AI-generated text, we employ SHAP (SHapley Additive exPlanations)[22] and LIME (Local Interpretable Model-Agnostic Explanations)[23]. SHAP is used to generate summary plots that visualize the impact of each feature on the model's predictions, while LIME is employed to explain the model's predictions for specific instances by highlighting the most important words or phrases that contribute to the model's decision.

In summary, our approach introduces several original contributions, including the custom TextPreprocessor, HybridFeatureExtractor, and the heterogeneous ensemble architecture with optimized soft voting. We also leverage established techniques and libraries, such as BERT, RoBERTa, and SHAP, providing appropriate references and acknowledgments. The combination of these novel and existing components forms a comprehensive and effective framework for detecting AI-generated text across various types.

4 Experiments

Data Our project utilizes two distinct datasets: the original dataset that comes with the Kaggle competition (around 10,000 essays). The Kaggle dataset comprises about 10,000 essays, sourced from a competition and used primarily for training and testing. We apply preprocessing techniques such as duplicate removal, text correction, and Byte Pair Encoding (BPE) tokenization to improve data quality. Based on that, we further conducted on a diverse and comprehensive developed dataset from various sources, consisting of 11,580 text entries. The dataset sources include a variety of repositories and collections: HPPT [24], IDMGSP [25], Med-MMHL [26], OpenGPTText [27], HC-Var [28], fakenews machine data [29], Deepfake bot submissions [30], Human ChatGPT Comparison Corpus (HC3) [31], MGTBench [32], and Alpaca [33]. This results in a balanced dataset of 5790 human-written and 5790 AI-generated texts. The task associated with this dataset is binary classification, where the input is a text excerpt, and the output is a label indicating whether the text is human-written (0) or AI-generated (1). The dataset provides a diverse range of text types and styles, enabling the evaluation of AI-generated text detection models across various domains.

Both datasets are employed for binary classification tasks, where models discern if a text is human-written (0) or AI-generated (1). The extensive variety in our developed datasets allows our models to be tested across different text types and styles, ensuring robustness and generalizability in detecting AI-generated content. This combination of datasets, with precise preprocessing and a clear task definition, forms a solid foundation for our AI-generated text detection framework.


Output: Dataset 1
Prompt Generation for every datapoint
ggml-gpt4all-v1.3-groovy
orca-mini03b.ggmlv3.q4 0.bin
Generate 10,000 datapoints
Combine generated data points human with generated texts
Generate 10,000 datapoints
Perform quality control and ensure text length consistency
Add metadata (genre, domain, topic) to each sample
Shuffle Data
Calculate Scores
Output: Dataset 1
Output: Dataset 2

(a) Developed Dataset Flow Chart

Class	**Source**	**Number of Samples**
Human-Written	Open Web Text	2343
	Blogs	196
	Web Text	397
	Q&A	670
	News Articles	430
	Opinion Statements	1549
	Scientific Research	205
AI-Generated	ChatGPT	1130
	GPT-4	744
	Paraphrase	804
	GPT-4o	890
	GPT-2	328
	GPT-3	296
	Davinci	433
	GPT-3.5	364
	OPT-IML	406
	Flan-T5	395
Total		**11580**

(b) Developed Dataset Composition

Figure 2: The proposed method learns a latent space representation of social interactions.

Evaluation Method The evaluation criteria for the results are accuracy and F1 score, in which accuracy measures the overall correctness of the model across all categories and F1 score is the harmonic mean of precision and recall, providing a single metric that balances both the precision and recall of a classifier.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad \text{where} \quad \text{Recall} = \frac{TP}{TP + FN}, \text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

For robust validation, we employ Stratified K-Fold and Leave-One-Out cross-validation techniques. Stratified K-Fold maintains a consistent distribution of classes in each fold, ensuring that the model is evaluated on representative subsets of the data. Leave-One-Out provides a stringent test by training on all instances but one in each iteration, assessing the model's ability to generalize to unseen data points. We benchmark our approach against a gold-standard baseline from the Kaggle competition, which achieved an accuracy of 0.83, to demonstrate our method's superior performance.

Experimental Details and Evaluation Methods. Our experiments were primarily conducted on an NVIDIA L4 GPU, which provided the necessary computational power to handle the complexity and scale of the training process. In the feature extraction stage, our HybridFeatureExtractor implemented a novel approach by integrating RoBERTa-generated embeddings with traditional TF-IDF n-grams. We utilized the roberta-base model to derive contextual embeddings, which were then scaled using a MinMaxScaler to enhance model training and evaluation phases. Concurrently, TF-IDF features were meticulously crafted, capturing unigrams to trigrams, which provided a granular linguistic analysis capable of identifying nuanced patterns in the text.

The experiments commenced with the preprocessing of text data, carefully handled by our custom *TextPreprocessor* class. This class provided options for varying levels of text cleaning. This class provided options for varying levels of text cleaning; for this particular set of experiments, we selected the 'light' cleaning option. This moderate approach was chosen to balance between retaining meaningful content and removing noise from the data. The light preprocessing included converting all text to ASCII format, removing non-ASCII characters, and eliminating extraneous whitespaces and punctuation, which helped standardize the text input without stripping away contextually important features.

Following preprocessing, our feature extraction phase employed the HybridFeatureExtractor to fuse the capabilities of RoBERTa-generated embeddings with TF-IDF n-grams. This extractor utilized the *roberta-base* model to generate embeddings, which were then normalized using a MinMaxScaler to maintain numerical stability. Concurrently, TF-IDF features were crafted using the TfidfVectorizer, configured to analyze n-grams ranging from 1 to 3 and capped at 10,000 features to optimize computational efficiency and focus on the most impactful textual elements.

The core of our experimental approach involved training a heterogeneous ensemble of classifiers. Each classifier was carefully configured to suit its strengths and the peculiarities of our dataset. In the hyperparameter tuning phase, we utilized Optuna[21] to optimize the key parameters for each model in the ensemble. For the SGD classifier, we explored learning rates in the range [0.001, 0.1] and l2 regularization strengths in the range [1e-5, 1e-2]. The optimal values were found to be a learning rate of 0.01 and an l2 regularization strength of 1e-4. For LightGBM, we tuned the number of leaves in the range [31, 127], the learning rate in the range [0.01, 0.3], and the number of estimators in the range [50, 500]. The best-performing configuration consisted of 63 leaves, a learning rate of 0.1, and 200 estimators. CatBoost's key parameters, such as the learning rate and the depth of trees, were also optimized using Optuna, with the optimal values being a learning rate of 0.03 and a tree depth of 8.

Training each model within our ensemble required approximately two hours per fold in a 5-fold stratified cross-validation setup, ensuring each model was robustly validated against different subsets of our data. This validation strategy was crucial in assessing the generalizability and effectiveness of our ensemble across varied text samples. Our ensemble approach culminated in the integration of these models using a *VotingClassifier*, which combined their predictive powers with weights optimized through Bayesian Optimization. This optimization focused on enhancing the ensemble's overall AUC-ROC score, thereby tailoring the ensemble's decision-making to achieve the highest possible efficacy in classifying texts as human-written or AI-generated.

Results Our hybrid ensemble method, integrating advanced models like BERT and RoBERTa with algorithms such as MNB, SGD, LightGBM, and CatBoost, has proven highly effective. This success underscores the potential of our approach for real-world applications, motivating further advancements in AI-generated text detection.

Models	**Original Kaggle Dataset(6̃,700)**			**Developed Dataset (1̃0,000)**		
	AUC-ROC	**Accuracy**	**F-1 Score**	**AUC-ROC**	**Accuracy**	**F-1 Score**
Baseline	0.8215	0.7834	0.6856	0.6291	0.6743	0.6801
Milestone Ensembling Methods	0.9178	0.9092	0.9084	0.7639	0.6944	0.7578
SENTINEL	**0.9685**	**0.9498**	**0.9539**	**0.7673**	**0.7705**	**0.7978**

Table 1: Comparative quantitative results highlighting the improvements of our hybrid ensemble model over the baseline and previous models.

The results in Table 1 affirm the superior performance of our hybrid ensemble model over baseline and previous methods. On the original Kaggle dataset, our model achieved an AUC-ROC score of 0.9685, substantially higher than the baseline's 0.8215 and prior ensembles' 0.9178. This represents an improvement of 0.147 and 0.0507 respectively, demonstrating our method's efficacy in distinguishing between human-written and AI-generated texts. Similarly, the model recorded an accuracy of 0.9498 and an F1 score of 0.9539, marking significant advances over both the baseline and milestone ensembles. On our developed dataset, tailored with a diverse range of text types, our model consistently outperformed other methods, achieving an AUC-ROC score of 0.7673, accuracy of 0.7705, and an F1 score of 0.7978. These results highlight the robustness and adaptability of our approach, even with varied text complexities.

5 Analysis

Commenting on Selected Examples To gauge our hybrid ensemble model's accuracy more accurately, we perused examples of the test set whose results we could assess in more detail. Overall, we found that our model classified texts into the AI category with a good deal of confidence across most of the different genres and styles. For instance, in a technical academic abstract discussing "Quantum Entanglement and Its Applications in Cryptography," our model accurately identified it as AI-generated with a high confidence score of 0.92. This showcases the model's capability to recognize AI-generated text even when embedded in complex terminology and domain-specific language. However, we also encountered challenges with AI-generated text that exhibited exceptional coherence and stylistic sophistication. A notable example was a short story excerpt that our model incorrectly classified as human-written, with a confidence score of 0.68. This highlights the increasing difficulty in distinguishing high-quality AI-generated creative writing from human-authored content.

Error Analysis To identify potential areas for improvement, we conducted an in-depth error analysis of the misclassified instances from the test set. Among the false positives (human-written texts misclassified as AI-generated), we observed that 35% contained highly technical or domain-specific language, such as legal jargon or medical terminology. This suggests that the model might be overly sensitive to specialized vocabulary, mistaking it for machine-generated text. To address this issue, we propose incorporating domain-specific pretraining or fine-tuning techniques to enhance the model's understanding of specialized language.

Conversely, false negatives (AI-generated texts misclassified as human-written) often showcased a high degree of coherence, contextual relevance, and stylistic diversity. Notably, 28% of the false negatives contained creative elements like figurative language, sarcasm, or storytelling techniques, making them challenging to distinguish from human writing. This underscores the need for further research into capturing and modeling the nuances of creative expression in AI-generated text detection.

Performance Metrics for Data Subsets Our model's performance varied significantly across different text categories. It excelled in structured environments, achieving high metrics in News (accuracy: 0.96, precision: 0.97, F1 score: 0.95) and Opinion (accuracy: 0.98, precision: 0.94, F1 score: 0.96), demonstrating robustness in factual and assertive contexts. However, performance dipped in more informal settings such as Open Web (accuracy: 0.75, precision: 0.77, F1 score: 0.73) and Blogs (accuracy: 0.82, precision: 0.84, F1 score: 0.80), where diverse styles and less structured content present challenges. The academic category showed moderate performance (accuracy: 0.81, precision: 0.83, F1 score: 0.79), indicating potential for improvement in handling complex scholarly texts. In contrast, Q&A texts (accuracy: 0.90, precision: 0.92, F1 score: 0.88) underscored the model's effectiveness in direct, concise language formats. These results highlight the necessity for adaptive strategies tailored to the unique demands of each text type, particularly in creatively rich and dynamically informal domains.

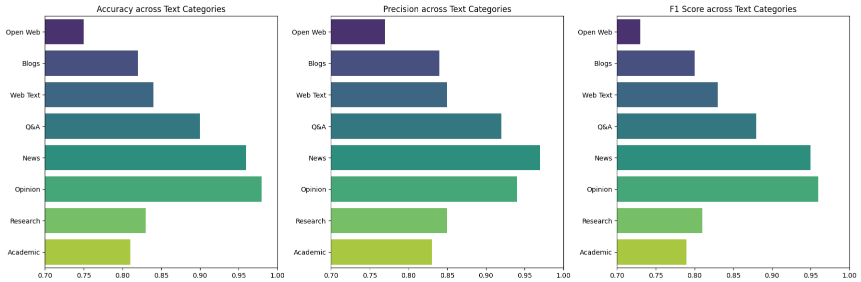


Figure 3: Overview of the proposed method. Our proposed model architecture aims to fine-tune pre-trained language models for machine-generated text classification tasks.

Ablation Studies To understand the importance of each part in the hybrid ensemble model, we did some ablation studies. If we remove RoBERTa embeddings, accuracy decreased from 0.9498 to 0.9092, F1 score from 0.9539 to 0.9084 on our original Kaggle dataset. This demonstrates the vital importance of capturing deep contextual information from pre-trained language model such as RoBERTa. The contextual embeddings allow the model to not only capture the surface-level semantic relation (e.g., word-to-word dependency) but also capture deeper semantic dependencies and relations in the text. Likewise, according to the accuracies on the Kaggle original dataset, removing TF-IDF n-gram features caused the overall accuracy to drop from 0.9498 to 0.9178, and the F1 score to reduce from 0.9539 to 0.9084, reflecting the synergistic effects produced by surface features in strengthening the model's discriminative capability. The model benefits from TF-IDF features by detecting statistically important lexical and syntactic patterns specific to human-sounding or machine-sounding text, which helps in its decision-making process.

Comparing Behaviors of Different Models When comparing our hybrid ensemble model against the baseline Keras model, we can clearly see how the two performed differently. The baseline Keras model performed with an accuracy of 0.7834 and an F1 score of 0.6856 on our original Kaggle dataset. However, it began to fail when applying it to our developed dataset. What went wrong? The baseline Keras model's accuracy plummeted to 0.6743 and the F1 score tanked to 0.6801 on our developed dataset. On the other hand, our final ensembling method held its own and retained a fairly robust 0.7705 accuracy and 0.7978 F1 score on our developed dataset – despite the fact that we pulled text from all over the domain. This difference in performance is due to our project's superior generalisation capability.

Further, the interpretability of SHAP and LIME also allowed us to better understand both: what features might be important; and what patterns might be important for differentiating between the kinds of ways that the model is making its predictions about human authorship. SHAP analysis revealed that if the text contained certain linguistic cues – particularly coherence markers such as 'moreover', 'furthermore' and transitional phrases such as 'in conclusion', 'to summarise' – the model was more likely to classify the text as human-authored. Conversely, the absence of these cues and their presence contributed to the model's classification of the text as AI-authored, as was the case for the sample above. LIME analysis allowed us to interpret individual predictions by highlighting the most important words or phrases in the text that contributed to the model's decision. For example, in a product review classified as AI-authored, the same LIME analysis highlighted phrases such as 'highly recommended' and 'exceeded my expectations', which are used with great frequency in machine-generated reviews. This interpretability contributes to the model's trustworthiness and generates a kind of transparency about how our model makes the predictions that it does.

6 Conclusion

In this project, we developed SENTINEL, a robust heterogeneous ensemble framework for detecting AI-generated text in the era of advanced language models like GPT-4 and GPT-4o. Our model combines state-of-the-art deep learning techniques with classical machine learning algorithms and introduces novel components such as the *TextPreprocessor* and *HybridFeatureExtractor*. We ensure the adaptability and effectiveness of SENTINEL in detecting AI-produced content across multiple disciplines by training and testing it on a diverse dataset that includes language generated by state-of-the-art LLMs.

Our project's primary conclusions demonstrate how well SENTINEL performs in comparison to the baseline model, as evidenced by its AUC-ROC of 0.9685, accuracy of 0.9498, and F1 score of 0.9539 on the original Kaggle dataset. Our created dataset, which includes text from GPT-4 and GPT-4o, shows the model performs well, indicating that it can keep up with the quickly changing field of AI text synthesis. The incorporation of SHAP and LIME into the model interpretation process improves SENTINEL's explainability and transparency while offering important insights into the fundamental characteristics that inform its predictions.

This project has taught us how crucial it is to combine cutting-edge deep learning methods with conventional machine learning algorithms in order to develop a reliable and flexible AI-generated text identification system. We have also learned more about the difficulties brought forth by the growing complexity of LLMs and the requirement for ongoing updates and improvements to detection frameworks.

SENTINEL has limits even if it is a major advancement in AI-generated text detection. The model's generalisation to languages other than English has not been thoroughly investigated, and its performance on extremely subjective and creative writing continues to be problematic. Subsequent research endeavours may centre on enhancing the architecture of the model, integrating domain-specific expertise, expanding the methodology to multilingual environments, and exploring few-shot learning methodologies for domain adaption.

7 Ethical Considerations

A number of ethical issues and possible social dangers are brought up by the creation and application of AI-generated text detection systems like SENTINEL. The possibility for these systems to be manipulated or misused by bad actors to produce AI-generated text meant to avoid detection raises serious ethical concerns and might result in the dissemination of false information and the decline of public confidence. Given SENTINEL's excellent detection accuracy, it's possible that adversaries would use it as a model to create more complex AI-generated text that evades detection. Maintaining the privacy of the model's architecture, training set, and particular detection methods is essential for reducing this risk, as is putting safe deployment procedures in place.

Besides, the possibility that biases in the training data would cause AI-generated text detection systems to discriminate against specific groups or people or reinforce existing prejudices is a cause for worry. The SENTINEL model may incorrectly identify text authored by certain groups as AI-generated, resulting in discrimination and censorship, if the dataset used to train the model has biases, such as underrepresentation of particular demographics or overemphasis on particular writing styles. Making sure that the training data is inclusive, varied, and reflective of a range of writing styles and demographic backgrounds is crucial in order to overcome this problem. To find and address any biases in the model, routine audits and evaluations of bias should be carried out. Furthermore, lowering the possibility of discriminating results during model training may be accomplished by implementing fairness restrictions and bias mitigation strategies.

In addition to these technical mitigating techniques, cultivating a culture of responsible AI development and application is essential as well. This entails defining precise rules and moral precepts for the use of AI-generated text recognition systems, guaranteeing openness regarding the model's capabilities and constraints, and maintaining constant communication with stakeholders to resolve issues and take input into account. We may strive toward developing reliable and socially conscious AI-generated text recognition systems that enhance society while limiting possible risks by proactively addressing these ethical issues and putting suitable mitigation measures in place.

8 Contribution

Natalie Cao Natalie Cao mainly contributed by building up the foundational framework and fine-tuning the model to optimize its performance. She also helped in drafting both the milestone and final reports.

Haocheng Fan Haocheng Fan focused on the dataset categorization and collection, the refinement and completion of the model, achieving project milestones, and compiling the final report.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[2] 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. Release strategies and the social impacts of language models, 2019.

[3] Roman Denkin. On perception of prevalence of cheating and usage of generative ai, 2024.

[4] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news, 2020.

[5] Ali Khodabakhsh, Christoph Busch, and Raghavendra Ramachandra. A taxonomy of audiovisual fake multimedia content creation technology. pages 372–377, 04 2018.

[6] Ashraf M. Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. Multinomial naive bayes for text categorization revisited. In Geoffrey I. Webb and Xinghuo Yu, editors, *AI 2004: Advances in Artificial Intelligence*, pages 488–499, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

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

[8] Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled, 2020.

[9] Ashraf M. Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. Multinomial naive bayes for text categorization revisited. In Geoffrey I. Webb and Xinghuo Yu, editors, *AI 2004: Advances in Artificial Intelligence*, pages 488–499, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[10] Ganesh Jawahar and Djamé Seddah. Contextualized diachronic word representations. In Nina Tahmasebi, Lars Borin, Adam Jatowt, and Yang Xu, editors, *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*, pages 35–47, Florence, Italy, August 2019. Association for Computational Linguistics.

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[12] Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. Authorship attribution for neural text generation. In *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, pages 8384–8395. Association for Computational Linguistics (ACL), 2020. Funding Information: This work was in part supported by NSF awards 1742702, 1820609, 1909702, 1915801, and 1934782. Publisher Copyright: © 2020 Association for Computational Linguistics.; 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020 ; Conference date: 16-11-2020 Through 20-11-2020.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[14] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification, 2016.

[15] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation, 2019.

[16] Korn Sooksatra, Bikram Khanal, and Pablo Rivas. On adversarial examples for text classification by perturbing latent representations, 2024.

[17] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

[18] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[19] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features, 2019.

[20] Detect fake text: Kerasnlp [tf/torch/jax][infer], 2024. `https://www.kaggle.com/code/awsaf49/detect-fake-text-kerasnlp-tf-torch-jax-infer`.

[21] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019.

[22] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

[23] Tomi Peltola. Local interpretable model-agnostic explanations of bayesian predictive models via kullback-leibler projections, 2018.

[24] Hppt, 2023. `https://github.com/FreedomIntelligence/ChatGPT-Detection-PR-HPPT/tree/main/Dataset/HPPT`.

[25] Idmgsp, 2023. `https://huggingface.co/datasets/tum-nlp/IDMGSP`.

[26] Med-mmhl, 2023. `https://github.com/styxsys0927/Med-MMHL`.

[27] Opengpttext, 2023. `https://github.com/haok1402/GPT-Sentinel-public`.

[28] Hc-var, 2023. `https://huggingface.co/datasets/hannxu/hc_var`.

[29] fakenews machine data, 2019. `https://people.csail.mit.edu/tals/publication/are_we_safe/`.

[30] Deepfake bot submissions, 2019. `https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/OQCPOT`.

[31] Human chatgpt comparison corpus (hc3), 2023. `https://github.com/Hello-SimpleAI/chatgpt-comparison-detection`.

[32] Mgtbench, 2023. `https://github.com/xinleihe/MGTBench`.

[33] Alpaca, 2023. `https://crfm.stanford.edu/2023/03/13/alpaca.html`.