I2C-Huelva at SemEval-2024 Task 8: Boosting AI-Generated Text Detection with Multimodal Models and Optimized Ensembles

Alberto Rodero Peña, Jacinto Mata Vázquez, Victoria Pachón Álvarez
I2C Research Group, Universidad de Huelva

Abstract

With the rise of AI-based text generators, the need for effective detection mechanisms has become paramount. This paper presents new techniques for building robust models and optimizing training aspects for identifying synthetically produced texts across multiple generators and domains. The study, divided into binary and multilabel classification tasks, avoids overfitting through strategic training data limitation. A key innovation is the incorporation of multimodal models that blend numerical text features with conventional NLP approaches. The work also delves into optimizing ensemble model combinations via various voting methods, focusing on accuracy as the official metric. The optimized ensemble strategy demonstrates significant efficacy in both subtasks, highlighting the potential of multimodal and ensemble methods in enhancing the robustness of detection systems against emerging text generators. This strategy was applied to subtask A, monolingual classification, ranking 47th with an accuracy of 0.8079, and subtask B, multilabel classification, ranking 18th with an accuracy of 0.789.

1 Introduction

In the era of digital communication, AI-based text generators have become increasingly sophisticated, necessitating advanced detection methods to differentiate between human and machine-generated content (Radford et al., 2019; Brown et al., 2020). This paper addresses the challenge within the scope of English language texts, emphasizing the importance of reliable detection mechanisms in maintaining the integrity of digital discourse. The task at hand is crucial for various applications, including content moderation, misinformation prevention, and ensuring the authenticity of digital communication.

The core strategy of this system lies in its robustness and the optimization of model training. By limiting the size of the training dataset, the approach prevents models from overfitting to specific text generators, thereby enhancing their generalizability to novel content. Furthermore, the system leverages multimodal models that integrate traditional NLP techniques with numerical text features, such as lexical diversity and sentence structure, to enrich the detection capabilities. This is complemented by a rigorous exploration of ensemble methods and voting mechanisms to optimize model performance. Specifically, our conception of multimodal entails the strategic fusion of traditional, fine-tuned language models like RoBERTa with extracted numerical values from the text, such as number of grammatical errors and average sentence length, thereby enriching the language models with quantifiable text insights to enhance detection precision.

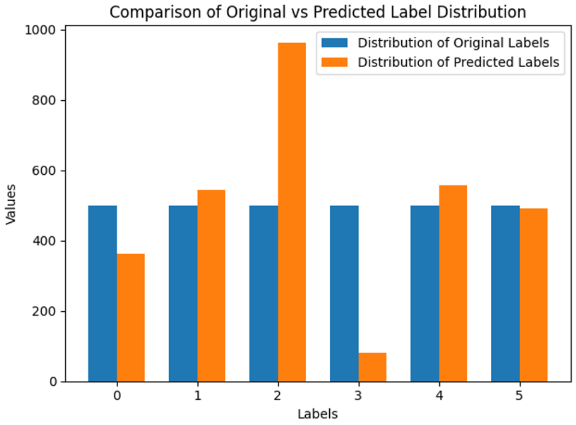


Figure 1: Comparison of Original vs Predicted Label Distribution Subtask B

Participation in this task led to the system ranking 47th in the monolingual subtask A with an accuracy of 0.8079 and 18th in the multilabel subtask B with an accuracy of 0.789 (Wang et al., 2024). These outcomes affirm the system's robustness in handling diverse generative models. However, the primary challenge encountered was distinguishing between texts produced by similar generators. The system struggled to consistently differentiate between certain generators, often misattributing texts to one over another when faced with stylistically comparable outputs. This difficulty in discerning subtle variations between generator styles points to the need for further refinement in the detection algorithm, suggesting an area for future research to enhance the sensitivity and specificity of the model. As shown in Figure 1, the double bar graph compares the original label distribution and the predicted label distribution. The labels represent the model that generated each text of the samples, being 0 human, 1 chatGPT, 2 cohere, 3 davinci, 4 bloomz and 5 dolly.

2 Background

The task of detecting machine-generated text has garnered significant attention due to the rapid advancement and widespread use of AI-based text generators. The input for this task consists of textual content, with the output being a classification decision indicating whether the text is human or machine-generated in subtask A and which generator created the text in subtask B.

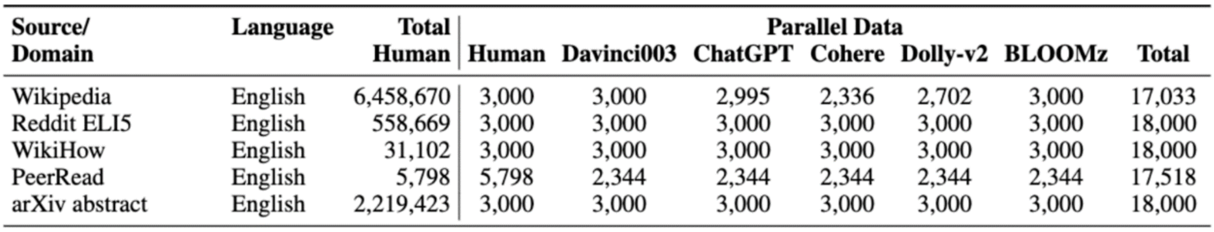
For this study, the dataset comprised English texts from diverse sources, including Wikipedia (March 2022 version), WikiHow, Reddit (ELI5), arXiv, and PeerRead (Koupaee and Wang, 2018; Kang et al., 2018). The machine-generated texts were produced using leading multilingual Large Language Models (LLMs) such as ChatGPT, textdavinci-003, LLaMa, FlanT5, Cohere, Dolly-v2, and BLOOMz. These models were prompted to create content resembling the human-written texts from the mentioned sources, ranging from Wikipedia articles to peer reviews and news briefs, ensuring a rich variety of genres and styles within the dataset. This richly varied dataset forms the foundation of the analysis, drawing on the comprehensive compilation of machine-generated texts as detailed in the work by Wang et al. (Wang et al., 2023). As shown in Figure 2, the training data distribution illustrates the sources and quantity of data used in the study.

This work focuses solely on the English portion of the dataset, engaging in the monolingual classification track. The choice of English allows for a concentrated examination of the nuances in detecting machine-generated texts in a language with extensive generative model research and development. The task setup and dataset composition are pivotal in understanding the challenges and innovations presented in this study.

This work builds upon foundational efforts in the field, such as "Machine-Generated Text Detection using Deep Learning" by Raghav Gaggar et al. (Gaggar et al., 2023), which emphasizes deep learning approaches for distinguishing AI-generated content. Gaggar's methodology leverages traditional neural network architectures, providing a critical basis for understanding how machine learning can be applied to text detection challenges. Similarly, "On the Possibilities of AI-Generated Text Detection" by Souradip Chakraborty et al. (Chakraborty et al., 2023) contributes to the discourse by establishing theoretical frameworks based on information theory, highlighting the nuanced differences between human and AI-generated texts and the implications for detection mechanisms. This paper underscores the importance of sample complexity and the robustness of detection systems to new and evolving text generators. "Ghostbuster: Detecting Text Ghostwritten by Large Language Models" by Vivek Verma et al. (Verma et al., 2023) methodology employs a series of weaker language models to compute token generation probabilities, offers a specialized perspective on model-agnostic detection. In contrast, this work extends the discourse by incorporating numerical text features alongside conventional NLP techniques within a multimodal framework, providing a more holistic analysis of text characteristics. This integration allows for a more nuanced distinction between human and AI-generated texts, addressing the challenges of style and generator diversity that single-model systems may struggle with.

3 System Overview

The system is designed to detect machine-generated text, combining an ensemble of finely-tuned transformer models such as RoBERTa ('FacebookAI/roberta-base') (Liu et al., 2019), ELECTRA ('google/electra-base-discriminator') (Clark et al., 2020), ALBERT (albert/albert-base-v2) (Lan et al., 2020), roberta-base-openai-detector ('roberta-base-openai-detector'), chatgpt-detector-roberta ('Hello-SimpleAI/chatgpt-detector-roberta') and BERT ('bert-base-uncased') (Devlin et al., 2018) with custom adaptations of RoBERTa including a one-vs-all system, that independently



Source/ Domain	Language	Total Human	Parallel Data Human	 Davinci003	 ChatGPT	 Cohere	 Dolly-v2	 BLOOMz	 Total
Wikipedia	English	6,458,670	3,000	3,000	2,995	2,336	2,702	3,000	17,033
Reddit ELI5	English	558,669	3,000	3,000	3,000	3,000	3,000	3,000	18,000
WikiHow	English	31,102	3,000	3,000	3,000	3,000	3,000	3,000	18,000
PeerRead	English	5,798	5,798	2,344	2,344	2,344	2,344	2,344	17,518
arXiv abstract	English	2,219,423	3,000	3,000	3,000	3,000	3,000	3,000	18,000

Figure 2: Training Data Distribution

predicts each label's presence, treating each label as a separate binary classification problem, and multimodal models . A Random Forest classifier is used to analyze numerical text features. This ensemble integrates outputs from each model by aggregating predictions and confidence levels through various voting mechanisms. The process identifies the best combination of models and voting method, optimizing the ensemble to achieve the highest detection accuracy and robustness.

3.1 Training Sample Optimization for robustness

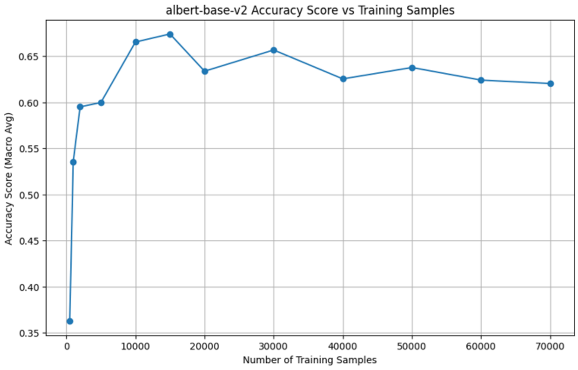


Figure 3: Accuracy by Training Sample Size

To optimize training samples for robustness, the number of samples used to train each model were systematically varied, aiming to find an optimal size that enhances robustness to new text generators while preventing overfitting. For instance, in the case of the ALBERT model, training began with 500 samples, then the model was reset and trained again with increasing sizes: 1000, 2000, 5000 samples, and so on. This process revealed that smaller sample sizes increased the model's robustness. It was determined that the ideal average number of samples for binary classification was 10,000, whereas multilabel classification required a larger average of 48,000 samples to maintain high predictive accuracy without compromising robustness to unseen generators in the evaluation dataset.

As illustrated in Figure 3, the graph shows the relationship between model accuracy and training sample size.

3.2 Numerical Features

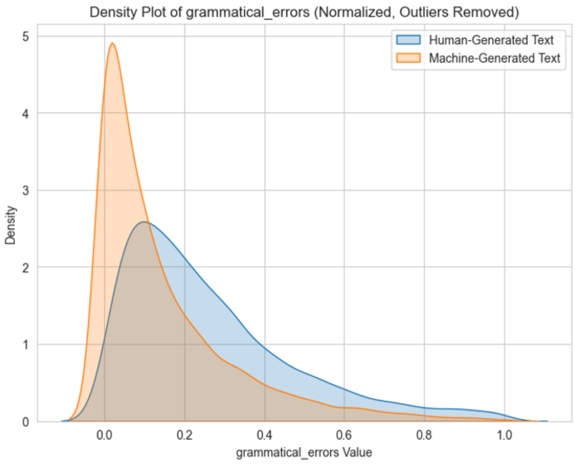


Figure 4: Human vs Machine Grammar Errors Density Plot

In addition to leveraging powerful transformer models, the system uniquely incorporates the extraction of numerical features from text to enhance its analytical depth. These features are: qord count, sentence count, lexical diversity, average sentence length, average word length, lexical density, flesch reading ease, gunning fog index, and grammatical errors in english, and they offer critical insights into the stylistic and structural elements of the text, which might be indicative of its origin. These features were obtained using methods from NLP libraries such as nltk. By analyzing these quantitative aspects, the system can identify subtle patterns and discrepancies that differentiate human-written texts from those generated by AI models, even when the linguistic content is convincingly human-like. This approach not only enriches the model's input but also helps in capturing the essence of text generation techniques used by various AI models, thereby contributing to a more robust detection mechanism. The system's primary aim with numerical features was to supplement the multimodal model with additional information. For the numerical values, a Random Forest classifier was chosen as it is an easy out-of-the-box solution to be used with numerical values (Louppe, 2014). However, this aspect was not the main focus, and further experimentation was not pursued. Future work could explore the use of deep learning and other classification models like XGBoost to analyze these numerical features. As depicted in Figure 4, the density plot illustrates the distribution of grammar errors between human-written and machine-generated texts.

3.3 Multimodal Models

The system's architecture is notably enhanced by the inclusion of multimodal models, which not only utilize the capabilities of traditional NLP models like RoBERTa but also integrate numerical text features for a more comprehensive analysis. This approach, applicable to any large language model, involves extending the chosen LLM's architecture with a custom classification head that processes both the LLM's output and additional numerical features from the text. For this study, RoBERTa was selected due to its role in establishing the baseline performance, allowing for a direct comparison of the improvements attributed solely to the multimodal functionality. Two different multimodal models were used. The extended version includes all the numerical features extracted from the text, which performs better in binary classification but not as well in multimodal classification. The second model uses only the features that show a clear difference between texts written by humans and those generated by machines. This model does better in multilabel classification but doesn't do as well in binary classification. The numerical features used in the multimodal model are word count, average sentence length, average word length, gunning fog index and grammatical errors. The extended version also includes sentence count, lexical diversity, lexical density and flesch reading ease. It is also worth mentioning that the performance between multimodal versions is slight.

3.4 Optimization of Ensembles

The optimization of ensembles through various voting mechanisms stands as a testament to the system's strategic design. The system tested every combination of models to make sure each one added value to the ensemble and did not take away any useful information. Specifically, models are chosen for their complementary strengths and diverse natures, ensuring a broad coverage of the linguistic and stylistic features pertinent to text generation detection. It used the predictions and confidence scores from all included models along with the correct labels. Then, it applied different voting methods to see how they compared to the real labels. This way, it found the best mix of models and the best voting method. The voting methods tested included majority voting, majority score tie break voting (confidence based), rank voting, borda count voting and soft voting (Brownlee, 2020). In the binary classification task, a larger variety of models is employed to capture the nuanced differences between human and machine-generated texts, whereas for the multilabel task, only two models are needed, reflecting the different demands of each subtask. Notably, multimodal models, recognized for their high accuracy, are consistently selected across both subtasks, reinforcing the ensemble's performance. The chosen strategy ensures that the ensemble's collective judgment is both robust and sensitive to the nuances of text generation, significantly enhancing the system's overall accuracy and reliability.

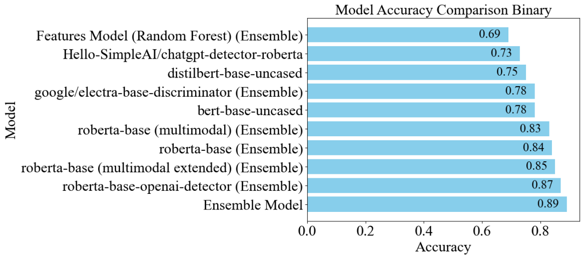


Figure 5: Substask A Models Accuracy in Training

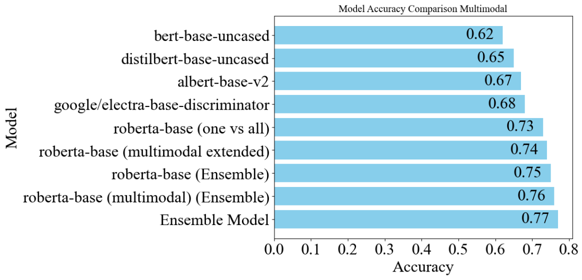


Figure 6: Subtask B Models Accuracy in Training

As illustrated in Figure 5, the bar graph compares the accuracy of individual binary classification models, providing insights into their performance. The final ensemble model accuracy is also included. Similarly, Figure 6 presents a comparison of the accuracy of individual multilabel classification models. The figures illustrate the models that were ultimately chosen to be included in the ensemble using majority voting.

4 Experimental Setup

In this study, the optimization of the training sample size was a critical preliminary step before proceeding with the standard division of the dataset for model training and evaluation. The objective was to determine the most effective training sample size that would enable the models to learn sufficiently from the data without overfitting. This involved iterative testing of various sample sizes to identify the optimal balance that maximized model performance on unseen data. Once the ideal training sample size was established, it was then split following an 80-20 ratio, with 80% of the samples used for training and the remaining 20% for evaluation. The dev dataset served as the test set throughout the experiments, ensuring a consistent benchmark for evaluating the generalization ability of the models across different configurations and optimizations.

Fine-tuning the models was conducted with careful consideration of hyperparameters that directly influence model performance. The hyperparameters were determined by experimenting with a range of values and choosing those that led to better performance metrics. This approach aimed to enhance the model's ability to be robust to new text generators that were present in the evaluation dataset but not in the training dataset. The learning rate was set to 2e-5, a value chosen to ensure steady yet effective model updates without causing large fluctuations in model weights that could hinder learning. The batch size for both training and evaluation phases was maintained at 16, balancing computational efficiency with the need for granularity in gradient updates. The models underwent training for 3 epochs, a decision underpinned by the desire to minimize overfitting while allowing sufficient iterations for the models to converge to an optimal state. Weight decay was applied at 0.01 to regularize the model and further mitigate overfitting. The training process incorporated an epoch-based evaluation and save strategy, enabling continuous monitoring of model performance and retention of the best-performing model state at each epoch's conclusion, as determined by evaluation metrics.

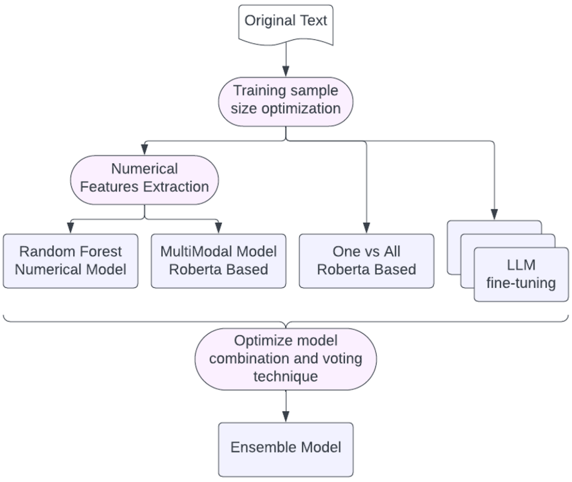

Original Text
Training sample size optimization
Numerical Features Extraction
Random Forest Numerical Model
MultiModal Model Roberta Based
One vs All Roberta Based
LLM fine-tuning
Optimize model combination and voting technique
Ensemble Model

Figure 7: Experimental Setup for Training

The experimental framework utilized PyTorch for implementing the transformer-based models, specifically leveraging the RoBERTa-base model from the Hugging Face Transformers library for both the multimodal models and the one-vs-all classification approach in the multilabel subtask. For numerical feature extraction from text, the NLTK library was employed, enriching the model inputs with linguistic features that provide additional context and depth to the analysis. The numerical model, built using a Random Forest classifier, was optimized using the scikit-learn library, demonstrating the integration of traditional machine learning techniques with advanced NLP models for enhanced predictive performance. As depicted in Figure 7, the diagram illustrates the experimental setup for training, showcasing the steps and pipelines involved in the process.

5 Results

The system demonstrated commendable performance in the task, adhering to the official evaluation metric of accuracy. In Subtask A (monolingual classification), the system attained an accuracy of 0.8079, placing it at the 47th position in the competition. This ranking underscores the system's capability to effectively distinguish between human and machine-generated texts in a monolingual setting. For Subtask B (multilabel classification), the system achieved an accuracy of 0.789, ranking 18th out of the total number of participants. This notable performance highlights the system's robustness and effectiveness in handling more complex multilabel scenarios, despite the inherently

Models	Accuracy	F1	Precision	Recall	AUC
RoBERTa-base-openai-detector	0.87	0.86	0.94	0.78	0.87
RoBERTa-base	0.84	0.84	0.83	0.86	0.84
bert-base-uncased	0.78	0.78	0.78	0.79	0.78
google/electra-base-discriminator	0.78	0.78	0.77	0.79	0.78
distilbert-base-uncased	0.75	0.74	0.83	0.62	0.75
RoBERTa-base (baseline)	0.74	-	-	-	-
Hello-SimpleAI/chatgpt-detector-RoBERTa	0.73	0.72	0.91	0.52	0.73
Features Model (Random Forest)	0.69	0.69	0.71	0.64	0.69
Multimodal Models					
RoBERTa-base (multimodal extended)	0.85	0.85	0.86	0.83	0.85
RoBERTa-base (multimodal)	0.83	0.83	0.88	0.77	0.83
Ensemble model (submission)	0.89	0.89	0.87	0.91	0.89

Table 1: Results obtained for Subtask A

Models	Accuracy	F1	Precision	Recall
RoBERTa-base	0.75	0.72	0.73	0.75
RoBERTa-base (baseline)	0.75	-	-	-
RoBERTa-base (one vs all)	0.73	0.7	0.71	0.73
google/electra-base-discriminator	0.68	0.65	0.68	0.68
albert-base-v2	0.67	0.65	0.66	0.67
bert-base-uncased	0.62	0.62	0.62	0.66
distilbert-base-uncased	0.65	0.63	0.66	0.65
Multimodal Models				
RoBERTa-base (multimodal)	0.76	0.72	0.73	0.76
RoBERTa-base (multimodal extended)	0.74	0.71	0.73	0.74
Ensemble Model (submission)	0.77	0.73	0.73	0.77

Table 2: Results obtained for Subtask B

challenging nature of distinguishing between multiple generators. These metrics were obtained after applying the ensemble model for each task.

In a comprehensive evaluation using the evaluation dataset, tables comparing model performances shed light on the system's effectiveness. For Subtask A, comparisons between various models in binary classification, and specifically between the RoBERTa-base model and its multimodal extensions, reveal the somewhat superior performance of the multimodal models. These models, incorporating key numerical features, mostly outperformed other fine tuned classificators. A similar trend was observed in Subtask B's multilabel classification, where multimodal models again demonstrated some enhanced accuracy. This data, while not from the final test set, underscores the potential of multimodal approaches in effectively distinguishing between human and machine-generated texts across different classification scenarios.

Table 1 provides a comprehensive metrics comparison for binary classification including multimodal models. It highlights the performance of fine-tuned LLMs for binary classification, including a Features only model built with a Random Forest classifier, and the performance evolution from base model RoBERTa-base to advanced multimodal models that integrate numerical features. The ensemble model is also included in this table, showcasing its role in the collective modeling approach. Table 2 provides a similar comparison but for multilabel classification scenarios.

5.1 Quantitative Analysis

A series of studies and comparative analyses were conducted to dissect the impact of various design decisions, such as the optimization of training sample sizes, the integration of numerical features, and the selection of models within the ensemble. The dev dataset served as the primary test bed for these analyses, ensuring consistency in evaluating the system's modifications and optimizations.

- A notable finding was the system's increased performance when numerical features were integrated, suggesting the significant value these features add to understanding text beyond mere semantic analysis.

- The ensemble's optimized combination of models, including transformer-based and numerical models, was pivotal in enhancing accuracy. The binary classification required a more diverse set of models to capture the nuances of different text generators, whereas the multilabel task achieved high performance with just two, indicating the strategic importance of model selection based on the task's nature.

5.2 Error Analysis

The examination of errors, particularly for Subtask B, shed light on the complexity of multilabel classification. The system was tasked with identifying multiple generator labels within the same text, a challenge compounded by the nuanced differences between generators' styles. As depicted in Figure 8, the confusion matrix heatmap provides insights into the errors made by the system in multilabel classification.

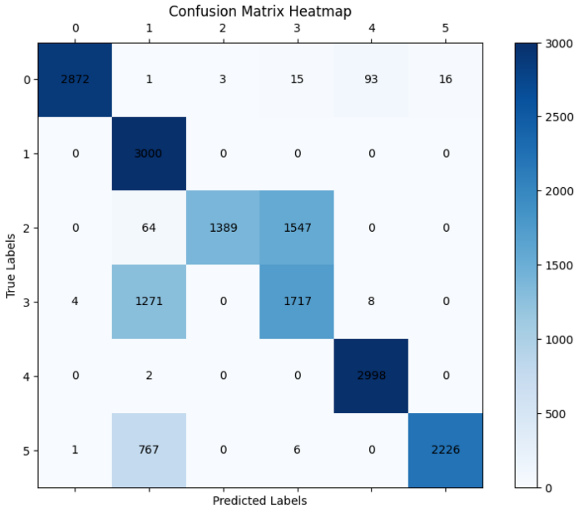


Figure 8: Confusion Matrix Heatmap Multilabel Subtask B

6 Conclusions

This study showcased innovative techniques aimed at enhancing model robustness in the task of detecting machine-generated text, notably through the careful optimization of training sample size, the strategic assembly of diverse models into optimized ensembles, and the deployment of multimodal models. These methodologies collectively facilitated a system that adeptly navigates the challenges of monolingual and multilabel classifications.

The exploration of training sample sizes revealed a delicate balance between sufficient model training and the avoidance of overfitting, highlighting the importance of dataset optimization. The ensemble model's success, derived from combining models with varying strengths, emphasizes the value of diversity in model architecture for robust performance. Moreover, the integration of multimodal models, blending traditional NLP techniques with numerical text features, showcased a sophisticated approach to capturing the nuanced distinctions between human and machine-generated texts.

Looking ahead, the focus will be on refining these novel techniques to further bolster model robustness. Future work will explore more granular adjustments to training sample sizes and investigate the potential of dynamic ensemble configurations responsive to the nature of the text being analyzed. Additionally, the extension of multimodal model frameworks to incorporate emerging linguistic and semantic features presents a promising avenue for enhancing detection capabilities. Applying these advanced methodologies to other areas of model building could advance the landscape of machine learning, offering a blueprint for developing systems that are not only robust but also universally applicable across various NLP tasks and challenges.

References

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. 2020. Language models are few-shot learners.

Jason Brownlee. 2020. How to develop voting ensembles with python. https://machinelearningmastery.com/voting-ensembles-with-python/. Accessed: date.

Souradip Chakraborty, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2023. On the possibilities of ai-generated text detection.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Raghav Gaggar, Ashish Bhagchandani, and Harsh Oza. 2023. Machine-generated text detection using deep learning.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and nlp applications.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4171–4186.

Gilles Louppe. 2014. Understanding random forests: From theory to practice. *arXiv*, 1407.7502.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2023. Ghostbuster: Detecting text ghostwritten by large language models.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection.