Unmasking AI-Written Content: A Novel Hybrid Kernel and Transformer-Based Detection Framework

Nazar Zaki
Department of Computer Science and Software Engineering,
College of Information Technology,
United Arab Emirates University,
Al Ain 15551, UAE
nzaki@uaeu.ac.ae

Reem Alderei
Department of Computer Science and Software Engineering,
College of Information Technology,
United Arab Emirates University,
Al Ain 15551, UAE
202102479@uaeu.ac.ae

Mahra Alketbi
Department of Information Systems and Security,
College of Information Technology,
United Arab Emirates University,
Al Ain 15551, UAE
202006536@uaeu.ac.ae

Alia Alkaabi
Department of Information Systems and Security,
College of Information Technology,
United Arab Emirates University,
Al Ain 15551, UAE
202108615@uaeu.ac.ae

Fatima Alneyadi
Department of Computer Science and Software Engineering,
College of Information Technology,
United Arab Emirates University,
Al Ain 15551, UAE
202302688@uaeu.ac.ae

Nadeen Zaki
Department of Computer Science and Software Engineering,
College of Information Technology,
United Arab Emirates University,
Al Ain 15551, UAE
700037065@uaeu.ac.ae

***Abstract*—The rapid evolution of large language models (LLMs) has made machine-generated text (MGT) increasingly indistinguishable from human writing, posing challenges in authenticity detection. In this study, we present a novel hybrid framework that integrates the structural sensitivity of string kernels with the semantic depth of transformer embeddings to detect AI-generated content. We propose four complementary methods including Attention-Augmented Kernels and a Custom Kernel Function to capture both linguistic structure and contextual nuance. Our evaluation across eight diverse datasets, featuring texts from GPT-3.5, GPT-4, DeepSeek, and Kimi, demonstrates that the Transformer-Guided N-gram Selection and Custom Kernel Function consistently outperform traditional baselines in accuracy and computational efficiency. This framework offers a scalable, interpretable, and robust solution for real-world MGT detection across varied generation styles.**

Keywords—AI-Generated Text Detection, Transformer Embeddings, String Kernels, Hybrid Models, Natural Language Processing

I. INTRODUCTION

The unprecedented advancements in large language models (LLMs), such as GPT-3.5, GPT-4, DeepSeek, and Kimi, have significantly enhanced the fluency, coherence, and contextual awareness of machine-generated text. These models can now generate content that closely mirrors human-authored writing, making it increasingly difficult to distinguish between the two. While this technological progress is transformative for many applications ranging from automated writing assistants to academic research support it also raises serious concerns about text authenticity, academic integrity, and the potential spread of misinformation. Consequently, developing reliable methods to detect machine-generated text (MGT) has become a pressing challenge in natural language processing (NLP).

Initial approaches to MGT detection were primarily based on stylometric analysis and shallow classifiers, using features such as word frequency, sentence length, and n-gram patterns. While interpretable and computationally efficient, these methods fall short when facing sophisticated LLM outputs that emulate human linguistic patterns. More recently, transformer-based models, especially those leveraging pre-trained architectures such as BERT and RoBERTa, have become the dominant paradigm for text classification and detection tasks due to their ability to capture deep contextual and semantic information [1], [2]. For instance, Bethany et al. [1] proposed T5LLMCipher, which utilizes subclustering of T5 embeddings to generalize across unseen generators, while Ben-Fares et al. [2] introduced a syntactically motivated model that aggregates token-level predictions from transformer layers for multilingual MGT detection.

Several studies have explored hybrid detection strategies that combine syntactic and semantic signals. Liu and Kao [3] proposed CopyCAT, which augments training data using saliency-aware masking to simulate varied input styles and improve robustness. Bafna et al. [4] introduced a RoBERTa-BiLSTM classifier that integrates contextual and sequential dependencies, improving detection across domains. Meanwhile, Abdalla et al. [5] emphasized explainability and generalization in detecting AI-generated scientific texts using transformer-based classifiers.

Robustness to paraphrased or obfuscated AI text is another core concern. Macko et al. [6] investigated the effects of authorship obfuscation techniques, such as paraphrasing and sentence fusion, on MGT detection performance, highlighting the trade-off between detection evasion and readability. Pu et al. [7] showed that models trained on medium-sized LLM outputs can generalize to more powerful models in zero-shot settings. Adversarial fine-tuning strategies have also been proposed to improve resilience, such as the work by Lee and Jang [8], which fine-tunes BERT to handle input variation in length and domain.

Benchmarking frameworks like SemEval, PAN, and IberLEF have standardized evaluation protocols for MGT detection. Yang et al. [9] introduced DNA-GPT, a training-free approach using n-gram divergence, while Bao et al. [10] presented Fast-DetectGPT, a zero-shot method leveraging conditional probability curvature to detect LLM outputs efficiently. However, many existing methods are tested only on narrow datasets or outdated models, limiting their generalizability and real-world applicability.

Human performance in MGT detection has also been studied. Ippolito et al. [11] found that human evaluators often fail to detect AI-generated content when it is grammatically correct and topically relevant. Lyu et al. [12] and Giudice et al. [13] identified common weaknesses in AI-generated texts, such as incoherent topic transitions and unnatural length distributions. Tools like RoFT [14] and Binoculars [15]have been used to evaluate and compare human vs. machine performance, consistently showing that even simple lexical patterns can mislead both.

Another limitation of many transformer-based approaches is their high computational cost and dependence on retraining or fine-tuning. As LLMs continue to evolve rapidly, models tied to specific generators may struggle to remain effective without constant updates. In contrast, simpler methods, while efficient, often lack the semantic depth needed to detect polished AI content. The need for scalable, adaptable, and interpretable detection models that balance performance and efficiency remains largely unmet.

In response to these challenges, we propose a novel hybrid detection framework that integrates the structural sensitivity of string kernels [16], [17] with the semantic richness of transformer embeddings. Our approach introduces four complementary methods:

- **Attention-Augmented Kernel**: Enhances traditional string kernel classification by incorporating attention scores from transformers to emphasize semantically important tokens.
- **Error Pattern Analysis**: Utilizes masked language model (MLM) loss to approximate text fluency and identify statistical anomalies characteristic of AI-generated content.
- **Transformer-Guided N-gram Selection**: Refines feature extraction by aligning n-gram boundaries with transformer tokenization, improving semantic coherence.
- **Custom Kernel Function**: Dynamically fuses semantic embeddings and structural n-gram patterns via a tunable weight, enabling adaptability across text styles and LLM variants.

We evaluate these methods on a comprehensive benchmark composed of eight datasets, including both AI-enhanced and fully AI-generated texts derived from GPT-3.5, GPT-4, DeepSeek, and Kimi. The datasets include both refined human-authored abstracts edited by LLMs and texts generated entirely from prompts, providing a rigorous testbed for detection. Notably, the proposed Transformer-Guided N-gram Selection and Custom Kernel Function consistently achieve high classification performance with significantly reduced computational overhead compared to pure transformer-based baselines.

Our main contributions are summarized as follows:

- We propose a novel hybrid framework for MGT detection that effectively combines semantic insights from transformers with the interpretability and efficiency of string kernels.
- We introduce a custom kernel function that balances structural and contextual features, enhancing adaptability to previously unseen AI-generated texts without retraining.
- We construct and evaluate on a diverse, publicly available benchmark dataset featuring multiple modern LLMs and text generation styles.
- We demonstrate that our methods outperform existing baselines in accuracy, robustness, and computational efficiency, offering a scalable and generalizable solution for real-world applications.

This study lays the foundation for a new class of interpretable and efficient AI-text detection systems that are not only accurate but also adaptable to the fast-changing landscape of generative language models.

II. METHOD

A. Datasets

The methodology involved constructing a diverse and robust dataset to effectively evaluate the detection of machine-generated text (MGT). The dataset includes eight subsets (D1–D8), each containing both human-written and AI-generated text samples. Human-written samples, sourced from Scopus scientific abstracts published between 2010 and 2015, predate modern AI models and thus provide an unbiased benchmark. AI-generated samples were produced using advanced language models such as GPT-3.5, GPT-4, DeepSeek, and KIMI, and labeled accordingly: AI-generated texts marked as "1" and human-written texts as "0".

The AI-generated samples are categorized into two groups: Refined AI-Generated Texts (D1–D4), involving AI enhancements or paraphrasing of original human-written abstracts, posing a challenging detection scenario; and Fully AI-Generated Abstracts (D5–D8), consisting entirely of AI-generated content based on given keywords, allowing assessment of the model's ability to distinguish purely synthetic text. The dataset is designed to rigorously evaluate detection model performance across various AI text-generation styles. Detailed descriptions of the datasets and the data itself are publicly available on GitHub (https://github.com/nzaki02/SK-LLM) as a resource for further research.

B. Method Overview

The proposed methodology integrates transformer-based deep contextual representations and structural text similarities captured by string kernels to robustly detect machine-generated text (MGT). Specifically, we introduce four novel kernel-based models designed to enhance the discriminative capabilities of MGT detection:

- **Attention-Augmented Kernel:** This model integrates transformer attention scores into traditional string kernels, effectively weighting structural features by their contextual importance.
- **Error Pattern Analysis Kernel:** This method leverages perplexity-based error scores derived

from masked language modeling (MLM) to capture subtle linguistic inconsistencies prevalent in AI-generated content.

- **Transformer-Guided N-gram Selection:** This approach aligns n-gram extraction directly with transformer tokenization, ensuring extracted features represent semantically coherent token sequences, thus improving model generalizability.
- **Custom Kernel Function:** A hybrid kernel is proposed, dynamically balancing deep semantic embeddings and structural n-grams via a tunable weighting parameter α, thus enabling tailored emphasis on either semantic depth or structural fidelity.

C. *Data Preprocessing*

Raw text data undergoes comprehensive preprocessing, including removal of extraneous symbols, special characters, and redundant whitespace. Subsequently, texts are tokenized using a pretrained BERT tokenizer to ensure uniform input representation. Structural linguistic patterns are extracted through character-level n-gram modeling (with n ranging from 2 to 5), and the dataset is randomly partitioned into training (80%) and testing (20%) subsets for rigorous model evaluation.

D. *Feature Extraction*

Multiple complementary feature types are extracted to optimize detection performance. We utilize BERT-derived contextual embeddings (CLS token) as deep semantic representations. To capture structural nuances, character-level n-gram features are extracted. Additionally, we quantify linguistic significance through mean attention scores derived from the transformer's final layer, measuring the overall token-level importance, and we approximate perplexity using MLM-based error scores to evaluate textual fluency and coherence.

E. *Mathematical Formulation*

- **Mean Attention Score:** $\boldsymbol{A_i = \frac{1}{HT}\sum_{h=1}^{H}\sum_{t=1}^{T} A_{h,t}}$ where H represents attention heads, T is the sequence length, and $A_{h,t}$ denotes the attention assigned by head h to token t.
- **Perplexity Approximation (Error Score):** $\boldsymbol{P_i = \frac{1}{N}\sum_{j=1}^{N} L_{MLM}(X_{i,j})}$ where N indicates token count, and L_{MLM} is the MLM loss.
- **Basic String Kernel:** $K_{string}(X_i, X_j) = V_i \cdot V_j^T$ where V_i, V_j are the respective n-gram vectors.
- **Attention-Augmented Kernel:** $K_{attention}(X_i, X_j) = (A_i V_i) \cdot (A_i V_i)^T$ incorporating attention-weighted structural vectors.
- **Error Pattern Analysis Kernel:** $\boldsymbol{K_{error}(X_i, X_j) = (P_i V_i) \cdot (P_j V_J)^T}$ incorporating perplexity-based error weighting.
- **Transformer-Guided N-gram Kernel:** $K_{guided}\,(\boldsymbol{X_i, X_j}) = \boldsymbol{V_{BERT-tokenized,i}} \cdot V_{BERT-tokenized,j}^T$ using transformer-aligned tokenization.
- **Custom Kernel Function (Proposed Model):** $K_{custom}(\boldsymbol{X_i, X_j}) = \boldsymbol{\alpha K_{semantic}\,(X_i, X_j) + (1 - \alpha) K_{structural}\,(X_i, X_j)}$ where semantic and structural kernels are defined as: $K_{custom}(\boldsymbol{X_i, X_j}) = E_i \cdot E_j^T, \boldsymbol{K_{structural}\,(X_i, X_j)} = V_i \cdot V_j^T$ and $\alpha \in [0,1]$ controls their relative importance.

F. *Baseline Model*

For baseline comparison, we implement a transformer-only classifier using solely deep contextual embeddings: $\hat{y} = f_{LLM}(E_i)$ where E_i represents BERT-derived embeddings, and f_{LLM} is the classification function.

All implementations and datasets used in this study are publicly accessible on GitHub (https://github.com/nzaki02/-SK-LLM), facilitating reproducibility and further research.

The evaluation of models employs several standard metrics to ensure comprehensive assessment: Precision, measuring correctness of positive predictions; Recall, measuring coverage of actual positive instances; and F1 Score, which balances precision and recall. Accuracy provides the overall proportion of correctly classified instances.

Computational efficiency is assessed by measuring total processing time during training and testing, including feature extraction, model fitting, and prediction phases. The processing time is recorded using Python's time module, allowing a comparative analysis of performance versus computational cost across different methods.

III. EXPERIMENTAL WORK AND RESULTS

Extensive experiments were conducted using eight diverse datasets (D1–D8) to evaluate the effectiveness of the proposed machine-generated text (MGT) detection methods. Models were implemented in Python 3.9, utilizing scikit-learn and Hugging Face Transformers, and executed on Google Colab with an NVIDIA A100 GPU for accelerated processing. Data was partitioned into 80% training and 20% testing subsets, and features were extracted using character-based n-grams ranging from 2 to 5. Performance was evaluated using multiple metrics, including precision, recall, F1-score, accuracy, and computational processing time.

The study compared two baseline models, the Basic String Kernel, which employs character-level n-gram similarity, and the LLM Only model, which utilizes BERT embeddings with four proposed enhanced methods: Attention-Augmented Kernel, Error Pattern Analysis, Transformer-Guided N-gram Selection, and the Custom Kernel Function. All methods utilized polynomial Support Vector Machines (SVM) with default hyperparameters, a regularization parameter (C=10), a convergence tolerance of 10^{-4}, and a maximum of 1,000 iterations. In the Custom Kernel Function, a balance between semantic (BERT embeddings) and structural (n-gram) features was maintained via the hyperparameter α set at 0.5.

Table I summarizes performance across all methods, demonstrating clear variations among datasets. Specifically, the Custom Kernel Function achieved the highest accuracy (0.7344) and F1-score (0.6383) on GPT-3.5-enhanced texts (Dataset D1). Conversely, the Transformer-Guided N-gram Selection method exhibited superior performance for GPT-4-

enhanced texts (Dataset D2), achieving an accuracy of 0.9016 and an F1-score of 0.8500. These results highlight the effectiveness of integrating structural and semantic textual features in MGT classification.

TABLE I. Comparative performance (Precision, Recall, F1-score, Accuracy) of baseline and proposed methods on eight datasets (D1–D8).

	Method	Precision	Recall	F1	Accuracy
Text enhanced using GPT-3.5 (Dataset D1)	Basic String Kernel	0.6190	0.5000	0.5532	0.6719
	Attention-Augmented Kernel	0.6087	0.5385	0.5714	0.6719
	Error Pattern Analysis	0.5769	0.5769	0.5769	0.6563
	Transformer-Guided N-gram Selection	0.5417	0.5000	0.5200	0.6250
	Custom Kernel Function	**0.7143**	**0.5769**	**0.6383**	**0.7344**
	LLM	0.3478	0.3077	0.3265	0.4844
Text enhanced using GPT-4 (Dataset D2)	Basic String Kernel	0.7500	0.7143	0.7317	0.8197
	Attention-Augmented Kernel	0.7778	0.6667	0.7179	0.8197
	Error Pattern Analysis	0.7368	0.6667	0.7000	0.8033
	Transformer-Guided N-gram Selection	**0.8947**	**0.8095**	**0.8500**	**0.9016**
	Custom Kernel Function	0.7619	0.7619	0.7619	0.8361
	LLM	0.7222	0.6190	0.6667	0.7869
Text enhanced using DeepSeek (Dataset D3)	Basic String Kernel	**1**	0.555556	0.714286	0.916667
	Attention-Augmented Kernel	**1**	0.555556	0.714286	0.916667
	Error Pattern Analysis	**1**	0.555556	0.714286	0.916667
	Transformer-Guided N-gram Selection	**1**	**1**	**1**	**1**
	Custom Kernel Function	**1**	0.777778	0.875	0.958333
	LLM	**1**	**1**	**1**	**1**
Text enhanced using Kimi (Dataset D4)	Basic String Kernel	0.96296	**1**	0.981132	0.984848
	Attention-Augmented Kernel	0.962963	**1**	0.981132	0.984848
	Error Pattern Analysis	0.962963	**1**	0.981132	0.984848
	Transformer-Guided N-gram Selection	**1**	**1**	**1**	**1**

	Custom Kernel Function	**1**	**1**	**1**	**1**
	LLM	**1**	**1**	**1**	**1**
Text generated using GPT-3.5 (Dataset D5)	Basic String Kernel	0.88	**1**	0.93617	0.95
	Attention-Augmented Kernel	0.88	**1**	0.93617	0.95
	Error Pattern Analysis	0.814815	**1**	0.897959	0.916667
	Transformer-Guided N-gram Selection	**0.956522**	**1**	**0.977778**	**0.983333**
	Custom Kernel Function	0.88	**1**	0.93617	0.95
	LLM	**0.956522**	**1**	**0.977778**	**0.983333**
Text generated using GPT-4 (Dataset D6)	Basic String Kernel	0.88	**1**	0.93617	0.95
	Attention-Augmented Kernel	0.88	**1**	0.93617	0.95
	Error Pattern Analysis	0.814815	**1**	0.897959	0.916667
	Transformer-Guided N-gram Selection	**0.956522**	**1**	**0.977778**	**0.983333**
	Custom Kernel Function	0.88	**1**	0.93617	0.95
	LLM	**0.956522**	**1**	**0.977778**	**0.983333**
Text generated using DeepSeek (Dataset D7)	Basic String Kernel	**1**	**1**	**1**	**1**
	Attention-Augmented Kernel	**1**	**1**	**1**	**1**
	Error Pattern Analysis	**1**	**1**	**1**	**1**
	Transformer-Guided N-gram Selection	**1**	**1**	**1**	**1**
	Custom Kernel Function	**1**	**1**	**1**	**1**
	LLM	**1**	**1**	**1**	**1**
Text generated using Kimi (Dataset D8)	Basic String Kernel	0.979167	**1**	0.989474	0.989362
	Attention-Augmented Kernel	0.959184	**1**	0.979167	0.978723
	Error Pattern Analysis	0.979167	**1**	0.989474	0.989362
	Transformer-Guided N-gram Selection	**1**	**1**	**1**	**1**
	Custom Kernel Function	**1**	**1**	**1**	**1**

	LLM	**1**	**1**	**1**	**1**

Datasets enhanced by DeepSeek (D3) and Kimi (D4) showed exceptionally high performance across all evaluated methods. Notably, Transformer-Guided N-gram Selection and LLM methods achieved perfect scores in precision, recall, F1-score, and accuracy, demonstrating strong capability in distinguishing AI-enhanced content. The Custom Kernel Function also exhibited nearly flawless results.

For datasets containing fully AI-generated texts (D5–D8), detection performance remained consistently high across all methods. Transformer-Guided N-gram Selection and LLM methods particularly excelled, delivering perfect or near-perfect scores, especially for GPT-generated texts (D5, D6). These findings underscore the effectiveness of integrating semantic transformer embeddings with structural n-gram features. The Custom Kernel Function maintained robust, balanced performance, highlighting the benefits of its hybrid semantic-structural design.

Transformer-Guided N-gram Selection emerged as particularly effective, emphasizing the significance of tailored, transformer-based strategies in accurately identifying machine-generated text.

A. *Hyperparameter Optimization*

To ensure optimal classification performance, we conducted systematic hyperparameter tuning using an extensive grid search. Our classification pipeline included character-level n-gram feature extraction (via CountVectorizer), normalization (using StandardScaler), and a polynomial-kernel SVM. We explored various n-gram ranges, polynomial degrees, and regularization values (C), selecting the best configuration based on mean F1-score through five-fold cross-validation.

The optimal configuration was identified as an n-gram range of (2,5), a polynomial kernel of degree 2, and a regularization parameter of C = 10. This setup effectively captured underlying structural patterns while preserving both generalization capability and computational efficiency. The findings highlight the critical role of meticulous hyperparameter tuning in improving the accuracy and robustness of models designed for detecting machine-generated text.

IV. Discussion

The experimental results provide clear evidence supporting the effectiveness of hybrid methods integrating semantic transformer embeddings with structural n-gram features in detecting various forms of AI-generated and AI-enhanced texts. Specifically, the Custom Kernel Function excelled on GPT-3.5-enhanced texts (Dataset D1), effectively capturing nuanced linguistic alterations introduced by AI enhancements. Similarly, Transformer-Guided N-gram Selection significantly outperformed other methods on GPT-4-enhanced texts (Dataset D2), benefiting from aligning structural extraction with transformer tokenization, thus improving semantic representation and detection accuracy.

For texts enhanced using DeepSeek (D3) and Kimi (D4), both Transformer-Guided N-gram Selection and Custom Kernel Function methods achieved exceptionally high performance, underscoring their reliability in identifying AI-enhanced content. The fully AI-generated texts (Datasets D5–D8) consistently yielded near-perfect detection results across methods, particularly with Transformer-Guided N-gram Selection and the LLM-based baseline, indicating distinct identifiable features in purely synthetic texts.

Computational analyses revealed significant efficiency advantages of the Transformer-Guided N-gram Selection and Custom Kernel Function methods, both demonstrating linear computational complexity (O(Nd) and O(Nd)+O(ND), respectively). In contrast, purely structural methods exhibited quadratic complexity and slower execution speeds, highlighting the computational benefits of the proposed hybrid methods. These results emphasize the practical advantages of tailored transformer-guided and hybrid kernel-based approaches, which effectively balance accuracy and computational efficiency, making them ideal for real-time and large-scale detection scenarios.

V. Conclusion

This study introduces and rigorously evaluates a novel hybrid framework that combines the strengths of traditional string kernels with modern transformer-based embeddings to enhance the detection of AI-generated text. Central to our approach are two key innovations: Transformer-Guided N-gram Selection, which leverages contextual embeddings to inform the selection of meaningful textual patterns, and a Custom Kernel Function designed to capture nuanced syntactic and semantic differences between human and machine-generated text. These components work synergistically to improve both classification accuracy and computational efficiency when compared to baseline and state-of-the-art methods.

Our experimental design encompassed a broad spectrum of generative AI systems, including GPT-3.5, GPT-4, DeepSeek, and Kimi, to ensure a comprehensive evaluation across diverse model architectures and output styles. The results consistently demonstrated the robustness, scalability, and generalizability of our proposed framework. In addition, we conducted extensive hyperparameter optimization tuning kernel parameters, regularization strengths, and embedding configurations to underscore the practical importance of model calibration in real-world detection tasks. Our findings confirm that the interplay between symbolic and neural representations can yield significant gains in both detection precision and interpretability.

This study offers a promising direction for hybrid approaches to text authenticity detection. By bridging symbolic pattern recognition and contextual deep learning, our method sets the stage for more accurate, scalable, and explainable solutions in the ongoing challenge of distinguishing human-authored content from machine-generated text.

Despite these contributions, the study is not without limitations. The primary corpus consisted of scientific abstracts, which while structurally rich, may not fully represent the stylistic diversity found in other textual domains such as creative writing, informal discourse, or multilingual social media posts. This potentially restricts the cross-domain applicability of our model. Moreover, the human-written texts used as baselines were largely drawn from pre-transformer-era documents, which may differ stylistically and syntactically from current human writing that often mimics or responds to AI-generated content. Another limitation lies in the dependency on BERT embeddings; while effective, they may not capture the full spectrum of linguistic nuance available in more recent or specialized transformer architectures. Our exploration of the hyperparameter space, though systematic, also remained bounded by computational constraints.

Looking ahead, future research should aim to overcome these limitations through several avenues. First, expanding the dataset to include a richer variety of genres such as news articles, user-generated content, academic essays, and dialogue transcripts will enhance the ecological validity and generalization capability of detection models. Second, incorporating alternative or ensemble transformer models, such as RoBERTa, DeBERTa, and multilingual transformers like XLM-R, could further improve performance across different linguistic contexts. Third, ensemble strategies that combine the outputs of multiple classifiers may increase robustness against evolving AI-generated text styles and adversarial examples. Importantly, future efforts should also focus on enhancing model interpretability. The integration of explainable AI (XAI) methodologies such as attention visualization, saliency maps, and rule-based post hoc explanations will be crucial for fostering user trust and ensuring ethical deployment in sensitive domains like education, journalism, and policy.

VI. References

[1] Bethany, M. and Wherry, B. and Bethany, E. and Vishwamitra, N. and Rios, A. and Najafirad, P., "Deciphering Textual Authenticity: A Generalized Strategy through the Lens of Large Language Semantics for Detecting Human vs. Machine-Generated Text," in 33rd USENIX Security Symposium, 2024.

[2] Ben-Fares, M. and Zaratiana, U. and Hernandez, S.D. and Holgado, P., "FI Group at SemEval-2024 Task 8: A Syntactically Motivated Architecture for Multilingual Machine-Generated Text Detection," in 18th International Workshop on Semantic Evaluation, Proceedings of the Workshop, 2024.

[3] Liu, C.-L. and Kao, H.-Y., "CopyCAT: Masking Strategy Conscious Augmented Text for Machine Generated Text Detection," Lecture Notes in Computer Science, vol. 13935 LNCS, pp. 367-379, 2023.

[4] Bafna, J.S. and Mittal, H. and Sethia, S. and Shrivastava, M. and Mamidi, R., "Mast Kalandar at SemEval-2024 Task 8: On the Trail of Textual Origins: RoBERTa-BiLSTM Approach to Detect AI-Generated Text," in SemEval 2024 - 18th International Workshop on Semantic Evaluation, Proceedings of the Workshop, 2024.

[5] Abdalla, M.H.I. and Malberg, S. and Dementieva, D. and Mosca, E. and Groh, G., "A Benchmark Dataset to Distinguish Human-Written and Machine-Generated Scientific Papers," Information, vol. 14, no. 10, p. 522, 2023.

[6] Macko, D. and Moro, R. and Uchendu, A. and Srba, I. and Lucas, J.S. and Yamashita, M. and Tripto, N.I. and Lee, D. and Simko, J. and Bielikova, M., "Authorship Obfuscation in Multilingual Machine-Generated Text Detection," in Findings of the Association for Computational Linguistics: EMNLP 2024, 2024.

[7] Pu, X. and Zhang, J. and Han, X. and Tsvetkov, Y. and He, T., "On the Zero-Shot Generalization of Machine-Generated Text Detectors," Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 4799-4808, 2023.

[8] Lee, D.H. and Jang, B., "Enhancing Machine-Generated Text Detection: Adversarial Fine-Tuning of Pre-Trained Language Models," IEEE Access, vol. 12, pp. 65333-65340, 2024.

[9] Yang, X. and Cheng, W. and Wu, Y. and Petzold, L.R. and Wang, W.Y. and Chen, H., "DNA-GPT: Divergent N-Gram Analysis for Training-free Detection of GPT-Generated Text," in 12th International Conference on Learning Representations, ICLR 2024, 2024.

[10] Bao, G. and Zhao, Y. and Teng, Z. and Yang, L. and Zhang, Y., "Fast-DetectGPT: Efficient Zero-Shot Detection of Machine-Generated Text via Conditional Probability Curvature," in 12th International Conference on Learning Representations, ICLR 2024, 2024.

[11] Ippolito, D. and Duckworth, D. and Callison-Burch, C. and Eck, D., "Automatic detection of generated text is easiest when humans are fooled," in Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2020.

[12] Lyu, M. and Bao, C. and Tang, J. and Wang, T. and Liu, P., "Automatic Detection for Machine-generated Texts is Easy," IEEE Access, vol. 11, pp. 1379-1386, 2022.

[13] Giudice, O. and Maggi, A. and Nardelli, M., "Exploring Naive Approaches to Tell Apart LLMs Productions from Human-written Text," ACM International Conference Proceeding Series, 2023.

[14] Dugan, L. and Ippolito, D. and Kirubarajan, A. and Shi, S. and Callison-Burch, C., "Real or Fake Text?: Investigating Human Ability to Detect Boundaries between Human-Written and Machine-Generated Text," in Proceedings of the 37th AAAI Conference on Artificial Intelligence, 2023.

[15] Hans, A. and Schwarzschild, A. and Cherepanova, V. and Kazemi, H. and Saha, A. and Goldblum, M. and Geiping, J. and Goldstein, T., "Spotting LLMs With Binoculars: Zero-Shot Detection of Machine-Generated Text," in Proceedings of Machine Learning Research, 2024.

[16] Lodhi, Huma; Saunders, Craig; Shawe-Taylor, John; Cristianini, Nello; Watkins, Chris , "Text classification using string kernels," Journal of Machine Learning Research, p. 419–444, 2002.

[17] Zaki NM, Deris S, Illias R. , "Application of string kernels in protein sequence classification," Appl Bioinformatics, vol. 4, no. 1, pp. 45-52, 2005.