MarSan at PAN: BinocularsLLM, Fusing Binoculars' Insight with the Proficiency of Large Language Models for Machine-Generated Text Detection

Notebook for PAN at CLEF 2024

Ehsan Tavan[1,†], Maryam Najafi[1,2,†]

[1]NLP Department, Part AI Research Center, Tehran, Iran
[2]Department of Computer Science and Information Systems, University of Limerick, Castletroy, V94 T9PX Limerick, Ireland

Abstract

Large Language Models have revolutionized natural language processing, exhibiting remarkable fluency and quality in generating human-like text. However, this advancement also brings challenges, particularly in distinguishing between human and machine-generated content. In this study, we propose an ensemble framework called **BinocularsLLM** for the PAN 2024 'Voight-Kampff' Generative AI Authorship Verification task. BinocularsLLM integrates supervised fine-tuning of LLMs with a classification head and the Binoculars framework, demonstrating promising results in detecting machine-generated text. Through extensive experimentation and evaluation, we showcase the effectiveness of our approach in addressing this critical task, achieving a perfect ROC-AUC score of 96.1%, a Brier score of 92.8%, a C@1 score of 91.2%, an F1 score of 88.4%, and an F0.5u score of 93.2% across all test datasets. BinocularsLLM outperforms all participants and baseline approaches, indicating its superior ability to generalize effectively and distinguish between human and machine-generated content. Our framework achieves the **first rank** among 30 teams participating in this competition.

Keywords

PAN 2024, Large Language Models, Machine-Generated Text Detection, Instruction Fine-Tuning

1. Introduction

In recent years, Large Language Models (LLMs) have made remarkable advancements, generating text that closely mimics human language with high fluency and quality. Models such as ChatGPT [1], GPT-3 [2], LLaMa [3], and Mistral [4] demonstrate impressive performance in a variety of tasks including question-answering, writing stories, and analyzing program code. These technologies offer significant potential to enhance efficiency and scalability across various domains, driving innovation and productivity [5, 6].

Machine-generated text is now used in a wide range of applications, from powerful chatbots [7] and real-time language translation [8] to analyzing and generating program code [9]. However, the sophistication of these models also introduces new challenges in distinguishing between human-generated and machine-generated content.

The ability to reliably detect machine-generated text is crucial. With the rapid expansion of information on the internet, there is an increased risk of misinformation spreading unchecked. The misuse of LLMs for generating fake news, fake product reviews, and propaganda pose substantial threats to the integrity of online communication. Furthermore, malicious activities such as spamming and fraud are intensified by the advanced capabilities of these models. Effective detection mechanisms are essential to protect against these risks, ensuring that digital content remains trustworthy and authentic. Developing tools and strategies to automatically detect machine-generated texts is essential to mitigate the threats posed by the misuse of LLMs.

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

[†]These authors contributed equally.

ehsan.tavan@partdp.ai (E. Tavan); maryam.najafi@ul.ie (M. Najafi)
https://github.com/Ehsan-Tavan (E. Tavan)
0000-0003-1262-8172 (E. Tavan); 0000-0001-5025-2044 (M. Najafi)

In the PAN'24 "Voight-Kampff" Generative AI Authorship Verification task [10, 11], participants are faced with an innovative challenge. Their task involves examining two texts: one authored by a human and the other by a machine. The goal is to identify the text authored by a human. This task highlights the ongoing need for robust methods to differentiate between human and machine-generated content, underscoring the importance of continued research and development in this area [12, 13, 10].

In this study, we explore innovative approaches to machine-generated text detection by investigating several key hypotheses. First, we examine whether leveraging LLMs with instruction fine-tuning can enhance the effectiveness of detecting machine-generated content. Second, we test the feasibility of training LLMs with a classification head that utilizes softmax to produce accurate output labels. Lastly, we investigate whether combining zero-shot techniques, which utilize metrics like perplexity and entropy, with fine-tuned models can significantly improve the accuracy of machine-generated text detection. These hypotheses aim to push the boundaries of our current detection capabilities, potentially leading to breakthroughs in ensuring the authenticity of digital content.

In Section 3, we introduce BinocularsLLM, our proposed ensemble framework, which integrates fine-tuned LLama2 [3] and Mistral models with a classification head, while also incorporating the Binoculars [14] model. This framework undergoes evaluation on both the main and nine additional test datasets, demonstrating notably promising results.

In this paper, we conduct a comprehensive evaluation of Voight-Kampff Generative AI Authorship Verification tasks, comparing our proposed framework against both baseline models and state-of-the-art approaches. We have made our code and data publicly available on our GitHub repository[1] and our fine-tuned models are available on Hugging Face: Generative-AV-Mistral-v0.1-7b[2] and Generative-AV-LLaMA-2-7b[3]. Our contributions are organized as follows: Section 2 reviews the relevant background literature. Section 3 introduces BinocularsLLM. Section 4 details the evaluation metrics and presents the experimental results.

2. Background

The detection of machine-generated text has become a critical area of research, driven by the rapid advancement and widespread use of large language models (LLMs) such as GPT-4 [15], PaLM [16], and ChatGPT. This task is typically formulated as a classification problem. This section reviews existing methodologies categorized into supervised learning approaches, zero-shot detection models, and watermarking techniques.

Supervised Learning Approaches: Supervised learning methods train classifiers on labeled datasets [17, 18, 19]. Models like GPT2 Detector [20] and ChatGPT Detector [21] fine-tunes pre-trained models such as RoBERTa [22] on the output of GPT2 [23] and the HC3 [21] dataset. While these models demonstrate high accuracy within their training domains, they often struggle with generalization to out-of-domain texts [24, 25]. Techniques such as adversarial training [26] and abstention [27] have been explored to enhance robustness, but challenges remain, particularly in maintaining low false positive rates across diverse text distributions [28].

Zero-Shot Detection Models: Another approach to identifying machine-generated text involves zero-shot detection models, which leverage statistical features in texts without requiring explicit training on labeled datasets. These models, such as DetectGPT [29] and others [30, 31], analyze universal features inherent in machine-generated texts. They exploit concepts like entropy, perplexity, and n-gram frequencies to distinguish between human and machine-generated text. These models offer robustness across different types of text and languages, circumventing the domain-specific limitations of supervised classifiers [30]. However, the computational demands remain a significant challenge, particularly in methods relying on probability curvature and extensive perturbations [29, 31].

[1] https://github.com/MarSanTeam/BinocularsLLM
[2] https://huggingface.co/Ehsan-Tavan/Generative-AV-Mistral-v0.1-7b
[3] https://huggingface.co/Ehsan-Tavan/Generative-AV-LLaMA-2-7b

Watermarking Techniques: Watermarking involves embedding detectable patterns into the generated text that are imperceptible to humans but identifiable by algorithms. Grinbaum and Adomaitis [32] and Abdelnabi and Fritz [33] utilized syntax tree manipulation to embed watermarks, while Kirchenbauer et al. [34] required access to the LLM's logits to modify token probabilities. Although effective, these methods necessitate control over the text generation process, limiting their applicability to scenarios where such control is feasible.

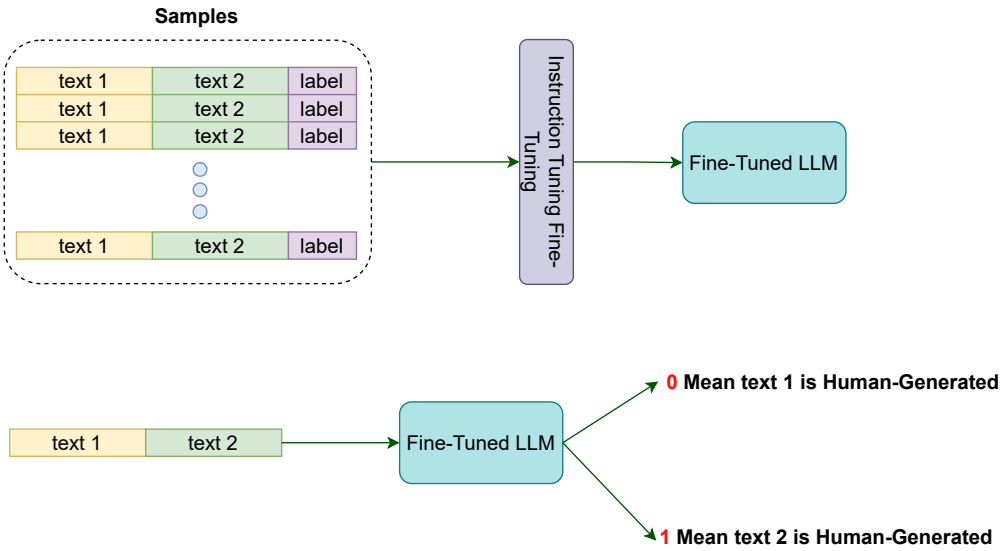

Figure 1: Instruction fine-tuning process.

3. System Overview

In this section, we present BinocularsLLM, our ensemble framework to address the PAN'24 "Voight-Kampff" Generative AI Authorship Verification task, with a focus on detecting machine-generated text. Our goals are twofold: to compare the effectiveness of classification-head fine-tuning with instruction fine-tuning and to integrate the power of the Binoculars technique with fine-tuned LLMs. Both approaches utilize QLoRA, ensuring that only the QLoRA and the classification head weights are trained, not all the parameters of the LLM.

The Binoculars model[4] employs observer and performer models to evaluate perplexity and entropy, critical metrics for identifying machine-generated text. By integrating these evaluations with the advanced capabilities of supervised fine-tuning, our ensemble is designed to be capable of distinguishing between human and machine-generated text.

Based on our experiments, we observed that LLM models employing a classification head performed more effectively in detecting machine-generated texts compared to instruction fine-tuning. Consequently, BinocularsLLM integrates two fine-tuned LLMs, LLaMA2 and Mistral (selected based on the results in Table 1), alongside the Binoculars approach. This comprehensive approach leverages the capabilities of statistical metrics and LLM fine-tuning, ensuring robust and accurate detection of machine-generated text.

3.1. Instruction Fine-Tuning for Machine-Generated Text Detection

Instruction Fine-Tuning (IT) involves further training LLMs with specific input-output pairs and accompanying instructions in a supervised manner. This approach has proven effective in enhancing an

[4]https://github.com/ahans30/Binoculars

LLM's ability to generalize to new, unseen tasks [35] and is considered a viable strategy for improving LLM alignment [36, 37].

In our study on Voight-Kampff Generative AI Authorship Verification, we examine the efficacy of the IT method. Specifically, we evaluate various LLMs' performance when fine-tuned with a specific set of instructions. This process involve creating an instruction dataset, V, comprising instruction pairs $s =$ (INSTRUCTION, OUTPUT). Each instruction s is generated using a fixed template and samples x from the training dataset R. These samples are labeled x_l based on their corresponding labels in dataset R. Figure 1 illustrates our instruction fine-tuning process.

The resulting instruction text detection dataset V consists of instruction pairs along with their source labels. A label of 0 indicates the first text is human-generated, while a label of 1 indicates the second text is human-generated. Thus, the instruction text detection dataset V includes pairs along with their corresponding source labels, formally represented as $V = \{(\text{instruction}, x, x_l) \mid x \in R\}$.

Here's an illustration of the instruction format:

> **Instruction:** I provide two texts and ask you to determine which one is authored by humans and which one is authored by machines. Your output is simply a 0 or 1; do not generate any additional text. 0 indicates Text1 is authored by the machine, and 1 indicates Text2 is authored by the machine.
> **Text1:** $[x\_text1]$
> **Text2:** $[x\_text2]$
> **Response:** $[x\_r]$

Given an LLM with parameters θ as the initial model for instruction tuning, training the model on the constructed instruction dataset V results in adapting the LLM's parameters from θ to θ_v, referred to as the LLM-Detector. Specifically, θ_v is obtained by maximizing the probability of predicting the next tokens in the OUTPUT component of each instruction sample s, conditioned on the INSTRUCTION. This process is formulated as follows:

$$\theta_v = \arg\max_{\theta} \sum_{s \in V} \log P(OUTPUT \mid INSTRUCTION; \theta, s) \tag{1}$$

3.2. Supervised Fine-Tuning LLMs

Fine-tuning LLMs involves adjusting model weights using a labeled dataset to enhance performance on specific tasks. This process can be computationally intensive, requiring significant memory resources, particularly when dealing with full LLM fine-tuning due to its substantial memory demands. To address these challenges, Parameter-Efficient Fine-Tuning (PEFT)[38] techniques such as LoRA [39] and QLoRA [40] are employed.

LoRA fine-tunes only two smaller matrices that approximate the larger weight matrix, reducing memory requirements and preserving the original LLM weights. Taking a step further, QLoRA enhances memory efficiency by quantizing these smaller matrices to a lower precision, such as 4-bit, without compromising effectiveness. Employing these fine-tuning techniques for both the classification head fine-tuning and instruction fine-tuning augments the LLM's capacity to accurately distinguish between machine-generated and human-generated text.

The Mistral and Llama2 models are fine-tuned exclusively using the provided bootstrap dataset and the QLoRA technique. Each input example consists of a text string ⟨TEXT⟩ and a corresponding label ⟨LABEL⟩ that indicates the source of the text. The input format can be represented as:

$$\langle \text{TEXT} \rangle : \langle \text{LABEL} \rangle, \quad \text{where} \quad \text{LABEL} = \begin{cases} 1 & \text{for human-generated text} \\ 0 & \text{for machine-generated text} \end{cases}$$

3.3. Inference Time

During the inference phase, the process initiates by receiving two texts as input. Each text is processed separately via the fine-tuned LLama2 and Mistral models to predict the probability of being human-written. If the probability assigned to the first text surpasses that of the second text, the score for the input sample is calculated by subtracting the score of the first text from that of the second. Conversely, if the probability of the second text is greater, the input text is labeled as 0.

Additionally, the input is also processed with the binoculars model, which generates a score for each text using its specialized algorithm. If the binoculars score of the first text exceeds that of the second text, the input score is assigned as 0; otherwise, it is assigned as 1. Figure 2 illustrates BinocularsLLM.

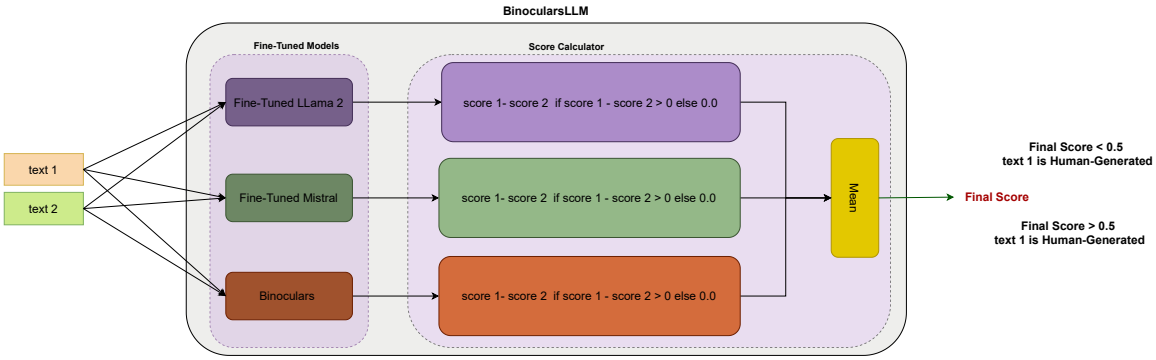

Figure 2: Overview of BinocularsLLM framework

4. Results

In this section, we present the implementation details, evaluation metrics, and provide a comprehensive analysis of the results. We utilize the TIRA [41] platform to evaluate our framework using test datasets.

4.1. Implementation Details

In this research, the framework was implemented in PyTorch and executed on Nvidia V100 GPUs. The training process was conducted for 5 epochs, utilizing the AdamW optimizer with a learning rate of 2e-5. The training batch size was set to 2, with gradient accumulation set to 8. For QLoRa, we configured LoRA's rank to 64 and its alpha to 16, employing 4-bit quantization. To evaluate fine-tuned models, we used 20% of the given dataset as a development dataset.

4.2. Evaluation Metrics

To evaluate the performance of our proposed model, we used the evaluation metrics provided by PAN, which include the following metrics:

- $ROC - AUC$: The conventional area under the curve score.
- $c@1$: Rewards systems that leave complicated problems unanswered.
- $F_{0.5u}$: Focus on deciding same-author cases correctly.
- $F1 - score$: A harmonic way of combining the precision and recall of the model.
- $Brier$: Evaluates the accuracy of probabilistic predictions.

4.3. Result Analysis on Development Dataset

As mentioned earlier, we compare two fine-tuning approaches for detecting machine-generated text: instruction fine-tuning and classification-head fine-tuning. The performance of various LLMs under

these methodologies is illustrated in Table 1 using the development dataset. Based on the results from Table 1, we select the two top-performing LLMs to integrate into our ensemble framework.

Table 1
Performance of different LLMs under classification head fine-tuning and instruction tuning using the development dataset.

Model	Classification Head						Instruction Fine-Tuning					
	roc	brier	c@1	f1	f05u	mean	roc	brier	c@1	f1	f05u	mean
llama3-7B	1	0.997	0.995	0.995	0.998	0.997						
llama2-7B	1	1	1	1	1	1	0.528	0	0.532	0.568	0.586	0.443
Mistral-7B	1	0.995	0.995	0.995	0.998	0.997	0.835	0.853	0.853	0.882	0.838	0.852
SOLAR-7B	1	0.986	0.986	0.984	0.993	0.99	0.623	0.624	0.624	0.655	0.672	0.64
zephyr-7B	1	0.986	0.986	0.984	0.993	0.99	0.526	0.532	0.532	0.582	0.588	0.552

In analyzing the results presented in Table 1, it becomes evident that both the LLama2-7B and Mistral-7B models, fine-tuned with a classification head, demonstrate promising performance across various evaluation metrics on our development dataset. LLama2-7B demonstrates exceptional scores across all metrics using the classification head fine-tuning approach, showcasing its robustness in distinguishing between human and machine-generated text. Meanwhile, Mistral-7B also has notable performance, indicating its efficacy in authorship verification tasks. These findings show the effectiveness of employing classification head fine-tuning for both LLama2-7B and Mistral-7B within the BinocularsLLM framework.

Comparing classification head fine-tuning with instruction tuning, we observe that classification head fine-tuning yields superior performance. These findings indicate that classification head fine-tuning is more effective than instruction tuning for enhancing the performance of LLMs in distinguishing between human and machine-generated text.

Table 2
Overview of the accuracy in detecting if a text is written by a human in task 4 on PAN 2024 (Voight-Kampff Generative AI Authorship Verification). We report ROC-AUC, Brier, C@1, F_1, $F_{0.5u}$ and their mean.

Approach	ROC-AUC	Brier	C@1	F_1	$F_{0.5u}$	Mean
BinocularsLLM	**1.0**	**0.995**	**0.997**	**0.997**	**0.999**	**0.998**
Binoculars	0.972	0.957	0.966	0.964	0.965	0.965
Fast-DetectGPT (Mistral)	0.876	0.8	0.886	0.883	0.883	0.866
PPMd	0.795	0.798	0.754	0.753	0.749	0.77
Unmasking	0.697	0.774	0.691	0.658	0.666	0.697
Fast-DetectGPT	0.668	0.776	0.695	0.69	0.691	0.704

4.4. Results on Blinded Test Dataset

As Table 2 shows, BinocularsLLM achieved outstanding performance across multiple evaluation metrics on the PAN 2024 Task 4 (Voight-Kampff Generative AI Authorship Verification) main test dataset, demonstrating its effectiveness in detecting machine-generated text. With a perfect ROC-AUC score of **1.0** and a Brier score close to **1.0**, **BinocularsLLM** exhibits high discriminative ability and excellent calibration. Additionally, **BinocularsLLM** outperforms all baseline approaches in terms of C@1, F1, and F0.5u scores. The mean evaluation score further underscores the robustness and reliability of the **BinocularsLLM** framework in distinguishing between human and machine-generated text.

Table 3 presents the analysis of BinocularsLLM across nine variants of the test set. The mean accuracy over these variants provides insights into the generalization capability of different approaches across diverse datasets. Among the approaches evaluated, the BinocularsLLM framework achieved the highest mean accuracy, with a median score of 0.990, indicating strong performance across various test variants.

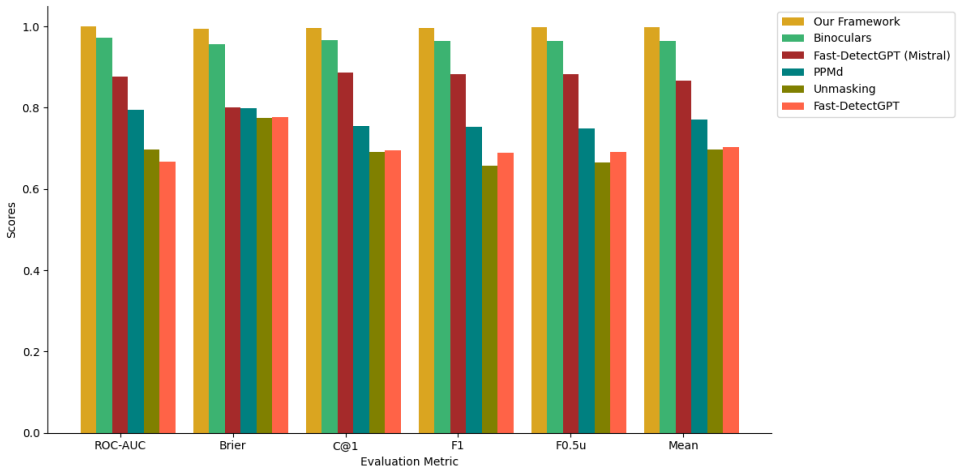

Figure 3: Comparison of model accuracy across different quantiles for various approaches.

Table 3
Overview of the mean accuracy over 9 variants of the test set. We report the minimum, median, maximum, the 25-th, and the 75-th quantile, of the mean per the 9 datasets.

Approach	Minimum	25-th Quantile	Median	75-th Quantile	Max
BinocularsLLM	**0.887**	**0.976**	**0.990**	**0.998**	**1.000**
Binoculars	0.342	0.818	0.844	0.965	0.996
Fast-DetectGPT (Mistral)	0.095	0.793	0.842	0.931	0.958
PPMd	0.270	0.546	0.750	0.770	0.863
Unmasking	0.250	0.662	0.696	0.697	0.762
Fast-DetectGPT	0.159	0.579	0.704	0.719	0.982
95-th quantile	0.863	0.971	0.978	0.990	1.000
75-th quantile	0.758	0.865	0.933	0.959	0.991
Median	0.605	0.645	0.875	0.889	0.936
25-th quantile	0.353	0.496	0.658	0.675	0.711
Min	0.015	0.038	0.231	0.244	0.252

However, when compared to baseline approaches, BinocularsLLM consistently outperforms them, showcasing its superior ability to generalize effectively. The performance of baseline approaches varies significantly across different datasets, as evidenced by the wide range between the minimum and maximum scores. This suggests that while some approaches exhibit consistent performance across diverse datasets, others may struggle to generalize effectively. Further analysis of the quantile values elucidates the distribution of performance scores, highlighting the variability and potential challenges in achieving consistent accuracy across different test variants.

PPMd and **Unmasking** display moderate performance, with median accuracies of **0.750** and **0.696**, respectively. However, their lower quantiles, particularly the minimum and 25th quantile, indicate significant variability and potential instability in their performance.

Fast-DetectGPT shows the most variability among the baselines, with a minimum accuracy of **0.159** and a maximum of **0.982**. This wide range suggests inconsistency and unreliability in different test scenarios.

The comparative analysis present in Figure 4 illustrates the discernible impact of training on the Mistral and Llama2 model. Before training, both models exhibited limited discriminatory capability on our **development dataset** between AI-generated and human-written text, as evidenced by the overlapping distribution of data points in their respective scatter plots. However, post-training, a

noticeable refinement emerges, with the models demonstrating enhanced proficiency in distinguishing between the two text categories. The scatter plots after training reveal a clearer separation between AI-generated and human-written text samples, indicating an improvement in the model's ability to capture distinguishing features inherent to each text type.

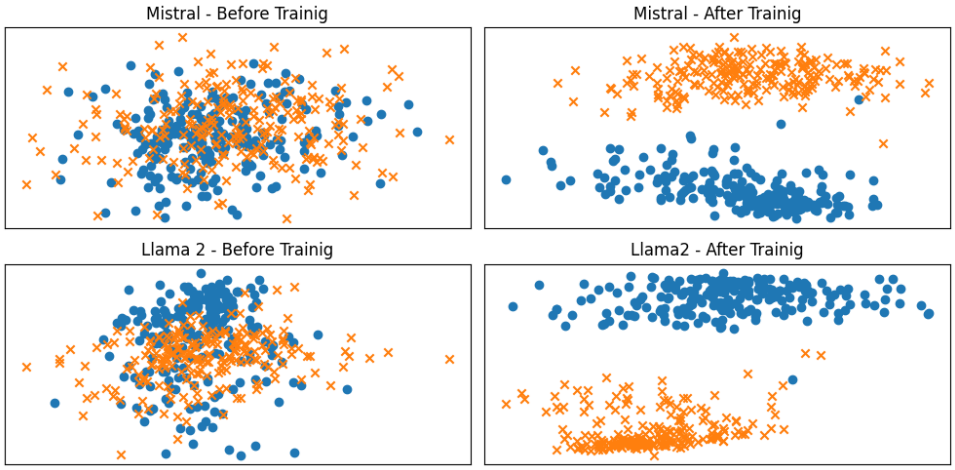

Figure 4: comparison: Mistral and Llama2 models before and after Training, specifically focusing on their ability to distinguish between AI-generated and human-written text.

4.5. Leaderboard on Test Datasets

Our team, **MarSan**, achieves the top position in the task leaderboard among 30 teams with our BinocularsLLM framework and demonstrates strong performance across various metrics. Table 4 outlines the performance metrics of the top 10 teams in the competition.

Table 4
Leaderboard on Test Datasets

Ranking	Team	ROC-AUC	Brier	C@1	F1	F0.5u	Mean
1	**MarSan (our)**	**0.961**	**0.928**	**0.912**	**0.884**	**0.932**	**0.924**
2	you-shun-you-de	0.931	0.926	0.928	0.905	0.913	0.921
3	baselineavengers	0.925	0.869	0.882	0.875	0.869	0.886
4	g-fosunlpteam	0.889	0.875	0.887	0.884	0.884	0.884
5	lam	0.851	0.850	0.850	0.852	0.849	0.851
6	drocks	0.866	0.863	0.834	0.825	0.820	0.843
7	aida	0.831	0.825	0.795	0.788	0.782	0.806
8	cnlp-nits-pp	0.844	0.793	0.805	0.789	0.792	0.806
9	fosu-stu	0.833	0.867	0.799	0.748	0.767	0.804
10	ap-team	0.853	0.862	0.795	0.718	0.742	0.796

5. Conclusion

In conclusion, the BinocularsLLM framework for the PAN 2024 "Voight-Kampff" Generative AI Authorship Verification task demonstrates significant advancements in detecting machine-generated text. Through the integration of supervised fine-tuning of LLMs with a classification head and the Binoculars model, we have achieved outstanding performance, as evidenced by a perfect ROC-AUC score of 1.0 and a Brier score close to 1.0 on the main test dataset. BinocularsLLM framework outperforms all baseline approaches in crucial evaluation metrics, highlighting its robustness and effectiveness in distinguishing between human and machine-generated content. Looking ahead, the success of our approach opens up exciting avenues for future research, including exploring more sophisticated ensemble techniques,

investigating the impact of different fine-tuning strategies, and addressing challenges related to scalability and computational efficiency. By continuing to innovate in this critical area, we can further advance the field of machine-generated text detection and contribute to enhancing the trustworthiness and authenticity of digital content.

References

[1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, Advances in neural information processing systems 35 (2022) 27730–27744.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020. arXiv:2005.14165.

[3] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open foundation and fine-tuned chat models, 2023. arXiv:2307.09288.

[4] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7b, 2023. arXiv:2310.06825.

[5] G. Jawahar, M. Abdul-Mageed, L. V. Lakshmanan, Automatic detection of machine generated text: A critical survey, arXiv preprint arXiv:2011.01314 (2020).

[6] N. Lu, S. Liu, R. He, Q. Wang, Y.-S. Ong, K. Tang, Large language models can be guided to evade ai-generated text detection, arXiv preprint arXiv:2305.10847 (2023).

[7] D. Bill, T. Eriksson, Fine-tuning a llm using reinforcement learning from human feedback for a therapy chatbot application, 2023.

[8] Y. Moslem, R. Haque, J. D. Kelleher, A. Way, Adaptive machine translation with large language models, 2023. arXiv:2301.13294.

[9] M. Nejjar, L. Zacharias, F. Stiehle, I. Weber, Llms for science: Usage for code generation and data analysis, arXiv preprint arXiv:2311.16733 (2023).

[10] J. Bevendorff, X. B. Casals, B. Chulvi, D. Dementieva, A. Elnagar, D. Freitag, M. Fröbe, D. Korenčić, M. Mayerl, A. Mukherjee, A. Panchenko, M. Potthast, F. Rangel, P. Rosso, A. Smirnova, E. Stamatatos, B. Stein, M. Taulé, D. Ustalov, M. Wiegmann, E. Zangerle, Overview of PAN 2024: Multi-Author Writing Style Analysis, Multilingual Text Detoxification, Oppositional Thinking Analysis, and Generative AI Authorship Verification, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2024), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2024.

[11] A. A. Ayele, N. Babakov, J. Bevendorff, X. B. Casals, B. Chulvi, D. Dementieva, A. Elnagar, D. Freitag, M. Fröbe, D. Korenčić, M. Mayerl, D. Moskovskiy, A. Mukherjee, A. Panchenko, M. Potthast, F. Rangel, N. Rizwan, P. Rosso, F. Schneider, A. Smirnova, E. Stamatatos, E. Stakovskii, B. Stein, M. Taulé, D. Ustalov, X. Wang, M. Wiegmann, S. M. Yimam, E. Zangerle, Overview of PAN 2024: Multi-Author Writing Style Analysis, Multilingual Text Detoxification, Oppositional Thinking Analysis, and Generative AI Authorship Verification, in: L. Goeuriot, P. Mulhem, G. Quénot,

D. Schwab, L. Soulier, G. M. D. Nunzio, P. Galuščáková, A. G. S. de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2024.

[12] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. URL: https://link.springer.com/chapter/10.1007/978-3-031-28241-6_20. doi:10.1007/978-3-031-28241-6_20.

[13] J. Bevendorff, M. Wiegmann, E. Stamatatos, M. Potthast, B. Stein, Overview of the Voight-Kampff Generative AI Authorship Verification Task at PAN 2024, in: G. F. N. Ferro, P. Galuščáková, A. G. S. de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2024.

[14] A. Hans, A. Schwarzschild, V. Cherepanova, H. Kazemi, A. Saha, M. Goldblum, J. Geiping, T. Goldstein, Spotting llms with binoculars: Zero-shot detection of machine-generated text, 2024. arXiv:2401.12070.

[15] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, B. Zoph, Gpt-4 technical report, 2024. arXiv:2303.08774.

[16] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay,

N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, N. Fiedel, Palm: Scaling language modeling with pathways, 2022. arXiv:2204.02311.

[17] M. Najafi, S. Sadidpur, Paa: Persian author attribution using dense and recursive connection (2024).

[18] E. Tavan, M. Najafi, R. Moradi, Identifying ironic content spreaders on twitter using psychometrics, contextual and ironic features with gradient boosting classifier., in: CLEF (Working Notes), 2022, pp. 2687–2697.

[19] M. Najafi, E. Tavan, Text-to-text transformer in authorship verification via stylistic and semantical analysis., 2022.

[20] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, et al., Release strategies and the social impacts of language models, arXiv preprint arXiv:1908.09203 (2019).

[21] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, Y. Wu, How close is chatgpt to human experts? comparison corpus, evaluation, and detection, arXiv preprint arXiv:2301.07597 (2023).

[22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.

[24] A. Bakhtin, S. Gross, M. Ott, Y. Deng, M. Ranzato, A. Szlam, Real or fake? learning to discriminate machine from human generated text, 2019. arXiv:1906.03351.

[25] A. Uchendu, T. Le, K. Shu, D. Lee, Authorship attribution for neural text generation, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 8384–8395. URL: https://aclanthology.org/2020.emnlp-main.673. doi:10.18653/v1/2020.emnlp-main.673.

[26] X. Hu, P.-Y. Chen, T.-Y. Ho, Radar: Robust ai-text detection via adversarial learning, 2023. arXiv:2307.03838.

[27] Y. Tian, H. Chen, X. Wang, Z. Bai, Q. Zhang, R. Li, C. Xu, Y. Wang, Multiscale positive-unlabeled detection of ai-generated texts, arXiv preprint arXiv:2305.18149 (2023).

[28] W. Liang, M. Yuksekgonul, Y. Mao, E. Wu, J. Zou, Gpt detectors are biased against non-native english writers, 2023. arXiv:2304.02819.

[29] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, C. Finn, Detectgpt: Zero-shot machine-generated text detection using probability curvature, 2023. arXiv:2301.11305.

[30] S. Gehrmann, H. Strobelt, A. Rush, GLTR: Statistical detection and visualization of generated text, in: M. R. Costa-jussà, E. Alfonseca (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Florence, Italy, 2019, pp. 111–116. URL: https://aclanthology.org/P19-3019. doi:10.18653/v1/P19-3019.

[31] J. Su, T. Y. Zhuo, D. Wang, P. Nakov, Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text, arXiv preprint arXiv:2306.05540 (2023).

[32] A. Grinbaum, L. Adomaitis, The ethical need for watermarks in machine-generated language, 2022. arXiv:2209.03118.

[33] S. Abdelnabi, M. Fritz, Adversarial watermarking transformer: Towards tracing text provenance with data hiding, 2021. arXiv:2009.03015.

[34] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, T. Goldstein, A watermark for large language models, 2024. arXiv:2301.10226.

[35] S. Longpre, L. Hou, T. Vu, A. Webson, H. W. Chung, Y. Tay, D. Zhou, Q. V. Le, B. Zoph, J. Wei, et al., The flan collection: Designing data and methods for effective instruction tuning, in: International

Conference on Machine Learning, PMLR, 2023, pp. 22631–22648.
[36] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T. B. Hashimoto, Stanford alpaca: An instruction-following llama model, https://github.com/tatsu-lab/stanford_alpaca, 2023.
[37] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, S. Zhang, G. Ghosh, M. Lewis, L. Zettlemoyer, O. Levy, Lima: Less is more for alignment, 2023. arXiv:2305.11206.
[38] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, B. Bossan, Peft: State-of-the-art parameter-efficient fine-tuning methods, https://github.com/huggingface/peft, 2022.
[39] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, 2021. arXiv:2106.09685.
[40] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms, 2023. arXiv:2305.14314.
[41] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. doi:10.1007/978-3-031-28241-6_20.