LOG-AID: Logit-Based Statistical Features for AI Text Detection

Notebook for PAN at CLEF 2025

Sophie Titze[1,*], Oren Halvani[1]

[1]Fraunhofer Institute for Secure Information Technology SIT, Rheinstraße 75, 64295 Darmstadt, Germany

Abstract

This submission addresses Subtask 1 of the Voight-Kampff Generative AI Detection task, which is part of the PAN 2025 lab. The goal of the subtask is to distinguish AI-generated texts from human-written ones, even when the machine-generated texts have been intentionally obfuscated to appear more human-like. As Large Language Models (LLMs) continue to improve in fluency and coherence, this distinction becomes increasingly difficult and requires robust detection strategies. This submission introduces a zero-shot method based on token-level statistics, which are extracted from two pre-trained LLMs: a base model and an instruction-tuned model. This method LOG-AID computes five core features: mean surprisal under each model, Jensen-Shannon divergence between their predictive distributions, average entropy difference, the mean entropy of the base model and the average logarithmic rank of the ground-truth tokens. These features are combined into a fixed-size vector and classified using a logistic regression model. On the official test set, the proposed system achieved a mean score of 0.827 across five metrics, surpassing strong baselines such as Binoculars (0.818) and PPMd Compression (0.758). In particular, the combination of uncertainty-based measures (surprisal, entropy) and rank-based features (log-rank) enhances discriminative power. This contribution offers a simple, interpretable and self-contained classification approach that does not require any fine-tuning. The method relies solely on internal probability structures of pre-trained models and may serve as a lightweight baseline for future work in AI text detection.

Keywords

PAN 2025, Voight-Kampff Generative AI Detection, AI Text detection, GenAI Detection, LLM Text Detection

1. Introduction

LLMs offer considerable benefits across domains. For example, they can enhance productivity by accelerating content creation and software development [1], increase accessibility through simplified language and assistive integration [2] or downstream tasks like sentiment analysis, translation, and summarization [3]. However, their growing fluency also enables large-scale misuse. LLMs can be exploited to generate convincing misinformation [4], produce spam and phishing content or facilitate academic fraud [5].

The Voight-Kampff Generative AI Detection (VKGen) task [6], hosted as part of PAN 2025 [7], addresses this challenge by providing a controlled benchmark for identifying AI-generated texts that have undergone stylistic obfuscation. In Subtask 1, the goal is to classify individual texts as either human-written or machine-generated, using only the raw text as input. The difficulty is amplified by the fact that the AI-generated samples may be modified to mimic human style and genre conventions [6].

This submission presents a simple and robust approach that requires no fine-tuning. The method leverages two pretrained language models, a base model and its instruction-tuned counterpart, to compute token-level metrics that reflect model confidence and divergence. These include mean surprisal, Jensen-Shannon divergence between predictive distributions, entropy differences, the base model's average entropy, and the average logarithmic rank of ground-truth tokens.

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

*Corresponding author.

{FirstName.LastName}@SIT.Fraunhofer.de (S. Titze); {FirstName.LastName}@SIT.Fraunhofer.de (O. Halvani)

0009-0008-4034-7048 (S. Titze); 0000-0002-1460-9373 (O. Halvani)

These five features are aggregated and passed to a logistic regression classifier.

Unlike many existing systems, this method does not rely on text similarity, fine-tuned classifiers or data augmentation. Instead, it exploits model-intrinsic probabilities to characterize how expected a text appears from the perspective of different LLMs. This lightweight framework aims to provide a transparent and modular baseline for robust detection.

2. Related Work

The participating methods in the PAN 2024 'Voight-Kampff Generative AI' Competition showed a wide range of methods. Many models were based on BERT or DeBERTa, either fine-tuned or combined with additional components such as R-Drop, LSTM, CNN or contrastive learning [8]. At the same time, a more classical approach was surprising: the third-placed method from Lorenz et al. achieved a mean score of 0.886 using only TF-IDF term count features and linear classifiers such as an SVM, which demonstrated the current relevance of proven feature engineering strategies [8]. The winning solution from Tavan and Najafi [9] was convincing with an ensemble of fine-tuned LLMs (Mistral, Llama2) and the Binocular's method as a central component [8]. It is worth noting that Binoculars in conjunction with Falcon-7B was also the strongest baseline in the competition [8]. The Binoculars method compares token-level output distributions between two LLMs to detect asymmetries in confidence. Specifically, it computes a ratio of perplexity from a performer model and the cross-entropy measured by an observer model [10]. Binoculars is one of the so-called white-box approaches [11]. These require direct access to the underlying LLM. Logit-based methods use the raw output of the model, i. e., the values from the last linear layer before the softmax function is applied. White-box methods typically work zero-shot. They therefore do not require their own training, but analyze the probability distributions provided by the model directly. The logits can be used to calculate statistical variables such as token-surprisal, entropy, rank of a token or divergence from expected patterns [11].

The GLTR (*Giant Language model Test Room*) method, presented by Gehrmann et al., is a visualization tool for the recognition of AI-generated texts [12]. It is assuming that generative language models prefer to choose words from the upper range of the probability scale. GLTR uses the prediction distributions of an LLM (e. g., GPT-2) to calculate the probability, rank and entropy of the prediction distribution for each token. Here, the rank refers to the occurrence in the vocabulary when this is sorted according to the probability of a token in its previous text sequence. In a user study, the tool increased the recognition rate of generated texts from 54% to 72% [12].

In the publication on DetectLLM, two powerful zero-shot methods for recognizing machine-generated texts were presented [13]. The methods DetectLLM-LRR and DetectLLM-NPR use either the ratio of log likelihood to log rank or the change of the log rank value under specific text perturbations to identify generated texts based on their typical statistical instabilities. However, the average logarithmic rank position of the tokens (log(rank)) already proved to be a particularly strong baseline, achieving higher ROC values compared to the non-logarithmic rank [13].

In addition Venkatraman et al. showed that surprisal can be used for the AI detection task [14]. Surprisal describes how unexpected or surprising a token is in a given context. Their method GPT-who calculates its mean value, variance and the differences between consecutive tokens. Furthermore, texts were segmented in 20 token long subtexts. For all 20-token-segments the surprisal features were calculated using a sliding window. The most extreme surprisal variances were extracted as additional features. These features were classified using a logistic regression. The approach achieved an average F_1 score of 0.88 on the TuringBench dataset [14].

Overall, logit-based metrics, including surprisal or entropy, in combination with simple classifiers like logistic regression, can achieve remarkably strong performance in recognising machine-generated text.

3. Method

3.1. Task Definition and Evaluation Protocol

Subtask 1 of the PAN 2025 VKGen Challenge addresses the binary classification problem of determining whether a given text was authored by a human or generated by an AI system. In contrast to previous years, this year's task introduces additional challenges by incorporating adversarially obfuscated texts, designed to humanize the writing style [6]. The input consists of a single text $\mathcal{T}$ and the goal is to predict whether it is machine-generated $(y = 1)$ or human-authored $(y = 0)$. The expected system output is a score $s \in [0, 1]$, where [6]:

- $s < 0.5$: $\mathcal{T}$ is classified as human-written,
- $s > 0.5$: $\mathcal{T}$ is classified as AI-generated,
- $s = 0.5$: the system abstains from making a prediction.

It should be emphasized that the LOG-AID method always outputs predictions without defining an uncertainty range for which $s = 0.5$ applies. In other words, we do not have any non-predictions, which have an effect on performance metrics such as $c@1$. In this specific case, $c@1$ equals the standard accuracy metric, according to Stamatatos et al. [15]. The participants were provided with two data sets: a training set with 17,730 texts (9,101 human, 8,629 machine) and a validation set with 3,589 texts (1,277 human, 2,312 machine). The texts come from three genres (fiction, news, essays) and exhibit a wide variety of styles and models. A total of 29 different AI models were used, with GPT models dominating. Each entry contains the label, the genre, the text and, where applicable, the generating model. A separate test set was retained and used exclusively for the final evaluation [6].

3.2. System Workflow

The proposed detection method LOG-AID for the PAN25 Voight-Kampff Challenge adopts a two-stage architecture. In the first stage, each input text is analyzed independently using two pre-trained causal language models: Falcon-7B[1] and its instruction-tuned variant Falcon-7B-Instruct[2]. These models operate in a zero-shot, autoregressive setting to compute token-level output distributions (e. g., logits). From these, six interpretable statistical features are derived, including mean surprisal, entropy difference and Jensen-Shannon divergence for both models. As well as entropy and the mean logit rank for the base model. To ensure efficiency, models are loaded sequentially. In the second stage, the feature vector is standardized using a z-score normalization and passed to a logistic regression classifier trained to distinguish between human and AI-generated texts.

3.3. Metric Descriptions

Surprisal quantifies how surprising a token w_t is for a language model in the given context $w_{<t}$ and corresponds to the negative logarithm of the associated prediction probability [14]:

$$\text{Surprisal}\,(w_t) = -\log p\,(w_t \mid w_{<t}) \tag{1}$$

In our approach, this metric is averaged over all tokens of a text. This reflects the average model uncertainty in predicting the words that actually occur. A lower mean surprisal value indicates a higher predictability of the text [14]. With **Shanon entropy**, the entropy of the prediction probability distribution $p\,(w \mid x_{<t})$ is calculated for each token x_t:

[1] https://huggingface.co/tiiuae/falcon-7b
[2] https://huggingface.co/tiiuae/falcon-7b-instruct

$$H(p) = -\sum_{i=1}^{V} p_i \log p_i \tag{2}$$

Here V is the size of the vocabulary and p_i is the predicted probability for the i^{th} token. In this implementation, entropy is computed per token using the scipy.stats.entropy [3] function applied to the softmax-normalized logit distributions. The average entropy is then calculated as the average of all token-specific entropy values over the entire text:

$$\text{Mean Entropy} = \frac{1}{T}\sum_{t=1}^{T} H\left(p_t\right) \tag{3}$$

In addition, the difference between the mean entropies of two models is calculated to determine divergent uncertainties between a base model and an instruct model:

$$\Delta H = \frac{1}{T}\sum_{t=1}^{T} \left| H^{(A)}\left(p_t\right) - H^{(B)}\left(p_t\right) \right| \tag{4}$$

To quantify the typicality of a token under the model's predictive distribution, we compute the **logarithmic rank** of each observed token within the vocabulary. The rank measures how highly the actual token x_t is located in the model's predicted probability distribution $p\left(w \mid x_{<t}\right)$, where $x_{<t}$ denotes the preceding context [12]. Formally, let $\operatorname{rank}\left(x_t\right) \in \mathbb{N}$ denote the position of the token x_t in the list of all vocabulary items $w \in V$, sorted by descending probability $p\left(w \mid x_{<t}\right)$. The average log-rank over a sequence of T tokens provides a robust scalar feature that reflects the model's overall perception of how typical the observed sequence is:

$$\text{Mean} \log\text{-}\operatorname{Rank}(x) = \frac{1}{T}\sum_{t=1}^{T} \log\left(\operatorname{rank}(x_t)\right) \tag{5}$$

where x_t is the token at position t and $\operatorname{rank}\left(x_t\right)$ is the index (starting at 1) of x_t in the sorted distribution. Low log-rank values indicate that tokens consistently appear near the top of the model's predicted distribution-suggesting a more stereotyped or expected sequence-while high values reflect less typical or more surprising lexical choices [12]. The **Jensen-Shannon divergence** (JSD) is used to quantify the divergence between the conditional probability distributions of two language models at each token position. It compares two probability distributions P and Q (in this case the logit probabilities of the base-model and the instruct-model) over the vocabulary V. The JSD is defined as [16]:

$$\operatorname{JSD}(P\|Q) = \frac{1}{2}D_{\mathrm{KL}}(P\|M) + \frac{1}{2}D_{\mathrm{KL}}(Q\|M), \quad \text{with} \quad M = \frac{1}{2}(P+Q) \tag{6}$$

Here, D_{KL} denotes the Kullback-Leibler divergence and M the mixed distribution. The Kullback-Leibler divergence between two discrete probability distributions P and Q over the vocabulary V is defined as [16]:

$$D_{\mathrm{KL}}(P\|Q) = \sum_{i=1}^{V} p_i \log \frac{p_i}{q_i} \tag{7}$$

where p_i and q_i denote the predicted probabilities of token i under P and Q, respectively. Since JSD is symmetric and restricted to the interval $[0, \log 2]$, it is particularly suitable for comparing probabilistic model [16]. In this method, the JSD is calculated for each token x_t based on the previous tokens $x_{<t}$. The mean JSD across all tokens serves as a measure of the average prediction deviation between the base and instruct models.

[3] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.entropy.html

3.4. Configurations and Development Environment

The final prediction is generated using a logistic regression classifier trained on the six-dimensional feature-vectors. Prior to classification, all feature values are standardized using z-score normalization (StandardScaler[4]). The logistic regression model[5] is configured with L2 regularization (default penalty), a regularization strength parameter C=1.0, and a maximum iteration limit of 1000. To account for class imbalance, the class_weight parameter is set to 'balanced'. The texts are processed in batches of the size 8. Training was conducted on a single NVIDIA H100 GPU (80GB HBM3) with CUDA 12.7, using approximately 60 GB of GPU memory. The program was then made accessible to the standardized evaluation procedures on the TIRA platform during the course of the competition and tested there. TIRA is a sandbox-based evaluation platform that enables the reproducible, isolated and fair execution of participant solutions [17].

4. Results

To evaluate the effectiveness of the proposed detection method, we report results on the official PAN25 test set. Evaluation was carried out by the competition organizers on data unknown to participants (see Tab. 1). All scores were computed using the PAN evaluation toolkit and include the following metrics: AUC, Brier score complement, C@1, F1 score, F0.5u, and the arithmetic mean of these five values [6]. The metrics used examine different aspects to evaluate model performance. AUC measures the probability that a positive example receives a higher score than a negative one [18]. The F_1-score measures the harmonic mean of precision and recall. The value is influenced by the selected positive class [19]. The brier score is the mean square error (MSE) between the predicted probabilities and the true labels [20]. The evaluation metrics all have a range of $[0, 1]$. The classic Brier score shows better performance at low values, which would make averaging with the other metrics inaccurate. To counteract this, its complement was formed within the competition. Another special feature of this year's competition was to give participants the opportunity to define a range of uncertainty in which no statement needs to be made. The c@1 value therefore also deals with non-predictions and is defined as follows [15]:

$$c@1 = \frac{1}{n}\left(n_c + \frac{n_c \cdot n_u}{n}\right) \tag{8}$$

Where n is the total number of instances, n_c is the number of correctly classified cases and n_u is the number of non-predicted cases. Since LOG-AID always makes a prediction, $n_u = 0$, which is why $\frac{n_c \cdot n_u}{n}$ disappears. The accuracy can be derived from the c@1 term as follows:

$$\frac{n_c}{n} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} = \mathrm{Accuracy} \tag{9}$$

As this is an unbalanced data set, the larger class is more important in the evaluation. For this reason, an additional metric, the balanced accuracy curve (BAC), is listed in Table 1. This is defined as follows [21]:

$$\mathrm{BAC} = \frac{\mathrm{TPR} + \mathrm{TNR}}{2} = \frac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}}\right) \tag{10}$$

Table 1 shows that the presented approach performs better than two of the three baselines. LOG-AID outperforms both the PPMd compression-based system and the Binoculars zero-shot baseline across all metrics, whereas the TF-IDF baseline performs better overall.

[4] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
[5] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Table 1
The final outcomes of the submitted solutions on the test data are shown. The two rightmost columns present the balanced accuracy and its rank, which was computed based on the reported FPR and FNR values.

rk	Team	AUC	Brier	$c@1$	F_1	$F_{0.5u}$	**Mean**	FPR	FNR	BAC	rk_{BAC}
1	mdok	0.853	0.896	0.894	0.898	0.903	**0.899**	0.108	0.094	0.899	1
2	steely	0.842	0.879	0.877	0.865	0.881	**0.880**	0.151	0.100	0.875	8
3	nexus-interrogators	0.865	0.874	0.870	0.860	0.881	**0.879**	0.159	0.083	0.879	6
4	yangjlg	0.845	0.878	0.871	0.856	0.881	**0.877**	0.172	0.062	0.883	4
5	cnlp-nits-pp	0.825	0.873	0.873	0.854	0.882	**0.874**	0.176	0.050	0.887	2
6	unibuc-nlp	0.828	0.885	0.864	0.845	0.876	**0.872**	0.187	0.052	0.881	5
7	moadmoad	0.822	0.866	0.865	0.855	0.882	**0.871**	0.175	0.058	0.884	3
8	iimasnlp	0.838	0.868	0.856	0.851	0.877	**0.869**	0.171	0.077	0.876	7
9	bohan-li	0.848	0.858	0.852	0.847	0.870	**0.866**	0.174	0.092	0.867	12
10	advacheck	0.802	0.855	0.855	0.854	0.879	**0.863**	0.169	0.084	0.874	9
11	hello-world	0.838	0.871	0.836	0.827	0.862	**0.856**	0.153	0.128	0.860	13
x	Baseline TF-IDF	0.838	0.871	0.836	0.827	0.862	**0.856**	0.153	0.128	0.856	x
12	xlbniu	0.794	0.847	0.847	0.840	0.869	**0.854**	0.188	0.077	0.868	10
13	shushantatud	0.823	0.850	0.840	0.831	0.862	**0.852**	0.203	0.093	0.852	14
14	ds-gt-pan	0.803	0.844	0.844	0.835	0.867	**0.851**	0.195	0.070	0.868	11
15	styloch	0.793	0.866	0.821	0.823	0.853	**0.844**	0.201	0.131	0.834	19
16	felix-volpel	0.815	0.854	0.816	0.822	0.855	**0.843**	0.212	0.115	0.837	18
17	sinai-inta	0.811	0.841	0.807	0.818	0.860	**0.838**	0.222	0.079	0.850	15
18	pindrop	0.782	0.854	0.814	0.815	0.853	**0.835**	0.211	0.115	0.837	17
19	diveye	0.786	0.828	0.806	0.823	0.862	**0.831**	0.211	0.104	0.843	16
20	s-titze	0.797	0.848	0.798	0.807	0.849	**0.827**	0.243	0.131	0.813	20
x	Baseline Binoculars	0.760	0.835	0.793	0.802	0.831	**0.818**	0.206	0.200	0.80	x
21	iunlp	0.734	0.799	0.799	0.829	0.850	**0.814**	0.178	0.210	0.806	21
22	hiwiy	0.765	0.806	0.771	0.830	0.791	**0.807**	0.000	0.636	0.682	24
23	team-a	0.603	0.783	0.783	0.824	0.801	**0.788**	0.049	0.457	0.747	23
x	Baseline PPMd	0.636	0.795	0.735	0.763	0.771	**0.758**	0.129	0.499	0.686	x
24	asdkklkk	0.718	0.739	0.739	0.726	0.781	**0.753**	0.308	0.110	0.791	22

5. Conclusion

This work presents a lightweight approach to detecting machine-generated text, based on a compact set of statistical features extracted from the output distributions of two pre-trained language models. Without relying on fine-tuning or large-scale training, the system achieves competitive results in the PAN25 Voight-Kampff Challenge, surpassing several strong baselines. By combining token-level metrics such as surprisal, entropy, log-rank, and Jensen-Shannon divergence into a logistic regression classifier, the system captures robust signals of artificiality and human-like variation. Empirical results demonstrate high accuracy on the test set. The method could be supplemented with additional features in future work. For example, not only the mean of entropy and surprisal could be computed, but also their standard deviations or burstiness. Moreover, it would be worthwhile to explore topic-masking techniques such as POSNoise [22] to guarantee a detection robustness against topic-related biases.

6. Acknowledgments

This research work was supported by the National Research Center for Applied Cybersecurity ATHENE. ATHENE is funded jointly by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research and the Arts.

Declaration on Generative AI

During the preparation of this paper, the authors used ChatGPT and DeepL for rewording, grammar and spelling checks. All content was subsequently reviewed and edited by the authors, who take full responsibility for the final version of the publication.

References

[1] T. Weber, M. Brandmaier, A. Schmidt, S. Mayer, Significant Productivity Gains through Programming with Large Language Models, Proc. ACM Hum.-Comput. Interact. 8 (2024). doi:10.1145/3661145.

[2] P. Martínez, A. Ramos, L. Moreno, Exploring Large Language Models To Generate Easy To Read Content, Frontiers in Computer Science 6 (2024). doi:10.3389/fcomp.2024.1394705.

[3] T. Shahzad, T. Mazhar, M. U. Tariq, W. Ahmad, K. Ouahada, H. Hamam, A Comprehensive Review of Large Language Models: Issues and Solutions in Learning Environments, Discover Sustainability 6 (2025). doi:10.1007/s43621-025-00815-8.

[4] L. De Angelis, F. Baglivo, G. Arzilli, G. P. Privitera, P. Ferragina, A. E. Tozzi, C. Rizzo, ChatGPT and the Rise of Large Language Models: The New AI-Driven Infodemic Threat in Public Health, Frontiers in Public Health 11 (2023). doi:10.3389/fpubh.2023.1166120.

[5] 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, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[6] J. Bevendorff, Y. Wang, J. Karlgren, M. Wiegmann, M. Fröbe, A. Tsivgun, J. Su, Z. Xie, M. Abassy, J. Mansurov, R. Xing, M. N. Ta, K. A. Elozeiri, T. Gu, R. V. Tomar, J. Geng, E. Artemova, A. Shelmanov, N. Habash, E. Stamatatos, I. Gurevych, P. Nakov, M. Potthast, B. Stein, Overview of the "Voight-Kampff" Generative AI Authorship Verification Task at PAN and ELOQUENT 2025, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2025.

[7] J. Bevendorff, D. Dementieva, M. Fröbe, B. Gipp, A. Greiner-Petter, J. Karlgren, M. Mayerl, P. Nakov, A. Panchenko, M. Potthast, A. Shelmanov, E. Stamatatos, B. Stein, Y. Wang, M. Wiegmann, E. Zangerle, Overview of PAN 2025: Voight-Kampff Generative AI Detection, Multilingual Text Detoxification, Multi-Author Writing Style Analysis, and Generative Plagiarism Detection, in: J. C. de Albornoz, J. Gonzalo, L. Plaza, A. G. S. de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2025.

[8] 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, G. M. D. Nunzio, L. Soulier, P. Galuscakova, A. G. S. Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. 15th International Conference of the CLEF Association (CLEF 2024), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2024.

[9] E. Tavan, M. Najafi, MarSan at PAN: BinocularsLLM , fusing Binoculars' Insight with the Proficiency

of Large Language Models for Machine-Generated Text Detection, in: G. Faggioli, N. Ferro, P. Galuščáková, A. G. S. Herrera (Eds.), Working Notes Papers of the CLEF 2024 Evaluation Labs, CEUR-WS.org, 2024, pp. 2901–2912. URL: http://ceur-ws.org/Vol-3740/paper-281.pdf.

[10] 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, in: Proceedings of the 41st International Conference on Machine Learning, ICML'24, JMLR.org, 2024, pp. 17519 – 17537. doi:10.5555/3692070.3692768.

[11] J. Wu, S. Yang, R. Zhan, Y. Yuan, L. S. Chao, D. F. Wong, A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions, Computational Linguistics 51 (2025) 275–338. URL: https://aclanthology.org/2025.cl-1.8/. doi:10.1162/coli_a_00549.

[12] 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. doi:10.18653/v1/P19-3019.

[13] J. Su, T. Zhuo, D. Wang, P. Nakov, DetectLLM: Leveraging Log Rank Information for Zero-Shot Detection of Machine-Generated Text, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 12395–12412. doi:10.18653/v1/2023.findings-emnlp.827.

[14] S. Venkatraman, A. Uchendu, D. Lee, GPT-who: An Information Density-based Machine-Generated Text Detector, in: K. Duh, H. Gomez, S. Bethard (Eds.), Findings of the Association for Computational Linguistics: NAACL 2024, Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 103–115. doi:10.18653/v1/2024.findings-naacl.8.

[15] E. Stamatatos, W. Daelemans, B. Verhoeven, M. Potthast, B. Stein, P. Juola, M. Sanchez-Perez, A. Barrón-Cedeño, Overview of the Author Identification Task at PAN 2014, in: L. Cappellato, N. Ferro, M. Halvey, W. Kraaij (Eds.), Working Notes Papers of the CLEF 2014 Evaluation Labs, volume 1180 of *Lecture Notes in Computer Science*, 2014. URL: https://ceur-ws.org/Vol-1180/CLEF2014wn-Pan-StamatosEt2014.pdf.

[16] F. Nielsen, On the Jensen–Shannon Symmetrization of Distances Relying on Abstract Means, Entropy 21 (2019). doi:10.3390/e21050485.

[17] 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.

[18] C. Cortes, M. Mohri, Auc optimization vs. error rate minimization, in: S. Thrun, L. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems, volume 16, MIT Press, 2003. URL: https://proceedings.neurips.cc/paper_files/paper/2003/file/6ef80bb237adf4b6f77d0700e1255907-Paper.pdf.

[19] S. A. Hicks, I. Strümke, V. Thambawita, M. Hammou, M. A. Riegler, P. Halvorsen, S. Parasa, On evaluation metrics for medical applications of artificial intelligence, Scientific Reports 12 (2022) 5979. URL: https://doi.org/10.1038/s41598-022-09954-8. doi:10.1038/s41598-022-09954-8.

[20] L. Hoessly, On misconceptions about the brier score in binary prediction models, 2025. URL: https://arxiv.org/abs/2504.04906. arXiv:2504.04906.

[21] K. H. Brodersen, C. S. Ong, K. E. Stephan, J. M. Buhmann, The balanced accuracy and its posterior distribution, in: 2010 20th International Conference on Pattern Recognition, 2010, pp. 3121–3124. doi:10.1109/ICPR.2010.764.

[22] O. Halvani, L. Graner, POSNoise: An Effective Countermeasure Against Topic Biases in Authorship Analysis, in: Proceedings of the 16th International Conference on Availability, Reliability and Security, ARES '21, Association for Computing Machinery, New York, NY, USA, 2021. URL: https://doi.org/10.1145/3465481.3470050. doi:10.1145/3465481.3470050.