# Team RADAR at PAN 2026: Robust Adversarial-Resistant AI Text Detection

Notebook for the PAN Lab at CLEF 2026

Yusr Alqarni*, Fahad AlGhamdi

*Al-Baha University, Al-Baha, Saudi Arabia*

**Abstract**

We present RADAR (Robust Adversarial-Resistant Detection with Adaptive Reasoning), a hybrid neural system for binary AI-generated text detection submitted to the PAN 2026 Voight-Kampff shared task [1, 2]. The system fuses deep semantic representations extracted by a large pretrained encoder with handcrafted stylometric signals through a cross-attention-style fusion mechanism, yielding a unified representation for classification. RADAR was trained in two stages: first with the encoder frozen to stabilize the fusion components, then with all weights jointly fine-tuned. Training combined the official PAN-2026 task data with a curated subset of the RAID benchmark to improve robustness against adversarial obfuscations. On the official PAN-2026 challenge test set, the encoder-frozen variant (RADAR-EF) achieves the highest mean score of 0.788 across all five evaluation metrics (ROC-AUC, Brier complement, C@1, F1, F0.5u), outperforming the end-to-end fine-tuned model. This finding suggests that preserving pre-trained encoder representations improves generalization to unseen generators and obfuscation strategies present in real evaluation conditions.

**Keywords**

PAN 2026, Voight-Kampff Generative AI Detection, Stylometric features, Forensic linguistics

## 1. Introduction

Authorship verification and AI-generated text detection have become increasingly important research problems in Natural Language Processing (NLP), especially with the rapid evolution of large language models (LLMs). Modern generative systems are capable of producing highly fluent and semantically coherent text that closely resembles human writing, making the distinction between human-written and machine-generated content considerably more difficult [3, 4, 5]. Concerns about the abuse of generative AI in areas such as academic dishonesty, misinformation, automated spam generation, and synthetic-content production have increased alongside this capability [6, 7].

To address this challenge, several detection approaches have been proposed, including statistical methods based on perplexity and token probability distributions [8], zero-shot detection techniques exploiting probability curvature properties [9], and supervised transformer-based classifiers trained on labelled corpora [10]. While existing approaches often achieve strong performance under controlled experimental conditions, their robustness is still limited when applied to unseen domains, previously unseen generators, or obfuscated texts [11]. In addition, many current detectors operate as opaque classification systems, offering limited interpretability and little linguistic justification for their predictions; this opacity can limit practical trust in forensic and institutional settings where high-stakes decisions require transparent evidence [12].

These challenges have increased interest in stylometry and forensic linguistics as complementary approaches for AI-generated text detection. Compared to purely neural detectors, stylometric methods focus on quantifiable linguistic features indicative of writing style, including lexical diversity, syntactic variation, punctuation usage, and readability characteristics [13, 14]. Such linguistic features are typically more interpretable and often remain effective even when superficial textual patterns are obscured. The

*CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany*

*Corresponding author.

yusralqarni@gmail.com (Y. Alqarni); fghamdi@bu.edu.sa (F. AlGhamdi)

0000-0002-4161-2658 (F. AlGhamdi)

PAN 2026 Voight-Kampff Generative AI Detection shared task especially targets these challenges by evaluating detection systems under real-world conditions involving style imitation, obfuscation, and previously unseen generators [1, 2]. In this paper, we present RADAR (Robust Adversarial-Resistant Detection with Adaptive Reasoning), a hybrid framework that combines transformer-based semantic representations with handcrafted stylometric features to improve robustness and generalization across challenging evaluation conditions. To further enhance generalization, the proposed framework is trained using both the official PAN-2026 dataset and a curated subset of the RAID benchmark dataset [15]. To support replication, we publish the source code of the RADAR approach[1].

## 2. Task and Data

### 2.1. PAN 2026 task setting

The PAN 2026 task asks systems to classify each input text independently as human-written or AI-written. The input file contains JSON lines with at least an identifier and text, and the output must contain one JSON line per input with the same identifier and a confidence score. This isolation constraint is important: a system must not exploit distributional information from other items in the same test file. RADAR satisfies this requirement because each text is preprocessed, featurized, encoded, and scored independently, with batching used only for computational efficiency.

The official PAN 2026 training and validation data include the fields `id`, `text`, `model`, `label`, and `genre`. Labels are binary: 0 for human and 1 for machine, and the genres are essays, news, and fiction. Only the text field was used as model input, while the label field was used for supervised training and evaluation. The `model` and `genre` metadata were retained for split description and diagnostic analysis only; they were not passed to the classifier as predictive features. In our local preparation notebooks, the original official training split contained 23,707 rows and the validation split contained 3,589 rows. After preprocessing and exact-text deduplication, 23,682 PAN training examples were retained, while the 3,589 validation examples were kept as the official validation reference.

### 2.2. External RAID data

To reduce overfitting to the official generators and domains, we augmented training with RAID, a large benchmark for robust machine-generated text detection that includes multiple generators, domains, decoding settings, and adversarial attacks [15]. We did not use RAID wholesale. Instead, we filtered it to match the PAN 2026 setting as closely as possible. The retained domains were `news`, `books`, and `poetry`, which are the nearest available equivalents of PAN's news, fiction, and essay-like writing. The retained attack types were `none`, `homoglyph`, `synonym`, `paraphrase`, and `alternative_spelling`. These attacks were selected because they represent clean generation, character-level obfuscation, lexical substitution, semantic rewriting, and spelling variation, respectively.

The raw RAID loading step produced 5,615,820 rows. Domain filtering reduced this pool to 2,239,440 rows, and attack filtering reduced it to 933,100 rows. We then created stratified RAID development and test pools and finally sampled a balanced 30,000-row RAID development subset and a balanced 10,000-row RAID held-out test subset. The down-sampling was intentional: the goal was to use RAID as a robustness regularizer without allowing it to dominate the smaller and task-specific PAN distribution.

### 2.3. Preprocessing

Preprocessing was deliberately conservative. Email addresses, user mentions, and phone numbers were replaced with placeholder tokens; text was lowercased and stripped; exact duplicates were removed; and very short texts below 50 characters were filtered. Exact deduplication refers to string-level duplicate removal after preprocessing and does not claim removal of semantic near-duplicates. We avoided aggressive normalization that could erase stylistic signals. For example, punctuation, paragraphing,

[1] https://github.com/yusrpro9/radar

digits, quotations, and capitalization-derived features remain accessible to the stylometric extractor before or during feature computation, while the transformer branch receives normalized text suitable for stable tokenization.

**Table 1**
Final data splits used in the RADAR experiments. The training mixture combines the official PAN 2026 training data with a balanced RAID development subset, increasing the effective training set size from 23,682 to 53,682 examples. The official validation split (3,589 examples) was retained unchanged for evaluation.

| Split | Total | Human | Machine | Machine % | Source |
|---|---|---|---|---|---|
| PAN train | 23,682 | 9,076 | 14,606 | 61.7 | Official PAN 2026 |
| PAN validation | 3,589 | 1,277 | 2,312 | 64.4 | Official PAN 2026 |
| RAID development sample | 30,000 | 15,000 | 15,000 | 50.0 | Filtered RAID |
| RAID held-out test sample | 10,000 | 5,000 | 5,000 | 50.0 | Filtered RAID |
| Training mixture | 53,682 | 24,076 | 29,606 | 55.2 | PAN train + RAID dev. |

## 3. System Overview

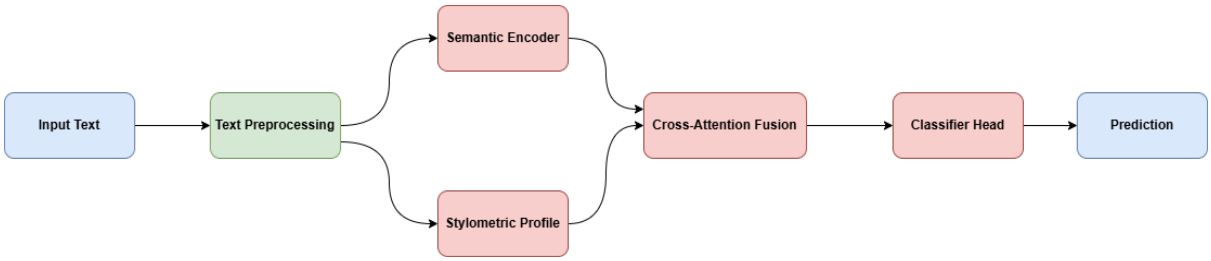


**Figure 1:** RADAR architecture. The system combines ModernBERT semantic encoding with a 38-dimensional stylometric profile. The two representations are fused through cross-attention-style fusion and scored with a sigmoid classifier.

### 3.1. Semantic encoder

The semantic branch uses `answerdotai/ModernBERT-large`. ModernBERT updates the encoder-only transformer family with architectural and training improvements for efficient long-context fine-tuning and inference [16, 17]. In our experiments, inputs were tokenized with a maximum length of 512 tokens. The first-token hidden state of the encoder is used as the semantic representation. Although ModernBERT supports longer contexts, the 512-token limit was chosen to control training cost and because the PAN texts in our processed splits were generally short enough for this setting to capture the main evidence. This limit should therefore be interpreted as a resource-control choice rather than a claim that longer-context evidence is unnecessary.

### 3.2. Stylometric profile

The stylometric extractor computes 38 normalized features grouped into five categories. Vocabulary richness features include type-token ratio, hapax ratio, Brunet's W, Honore's R, moving-average type-token ratio, vocabulary richness slope, rare-word ratio, and function-word density. Syntactic complexity features approximate sentence length, sentence-length variance, clause density, subordination, passive voice, noun-phrase-like capitalization, verb density, coordination, relative word position, and sentence-boundary rhythm. Discourse coherence features measure lexical cohesion, pronoun density, transition-word use, paragraph coherence, topic consistency, information-flow variance, semantic repetition, and

connective diversity. Style fingerprinting features capture punctuation density, capitalization, numeric tokens, quotation usage, parentheticals, dash/semicolon habits, and paragraph length variability. Finally, lightweight perplexity-signal features approximate compression ratio, character entropy, word-length entropy, repeated character patterns, and formulaic phrase usage.

These features are not intended to replace neural semantic modeling. Their role is to provide a compact and interpretable view of authorial form. They are particularly useful for adversarial settings because paraphrases may preserve or distort different layers of writing style. A detector that sees only token sequences may learn strong but brittle lexical correlations, whereas a detector that also receives normalized stylometric evidence can learn more stable distinctions.

The 38-dimensional feature vector is projected through a linear layer, layer normalization, GELU activation, and dropout into a 128-dimensional style representation. This projection allows the model to learn which handcrafted features are most useful for the task rather than treating all features as equally important.

### 3.3. Cross-attention fusion

RADAR fuses semantic and stylometric evidence using a lightweight cross-attention-style fusion module. The projected style vector is first mapped to the semantic hidden dimension. The semantic representation acts as the query, and the style representation acts as the key and value. The resulting attended semantic vector is added back to the original semantic vector through a residual connection and layer normalization. This vector is then concatenated with the projected style representation and passed through a fusion MLP to obtain a 512-dimensional fused representation.

Because the stylometric branch is represented as a compact projected vector rather than as a sequence of feature-level tokens, this component is used as a learned conditional fusion layer rather than as a source of feature-level attention explanations. The motivation for this design, compared with simple concatenation, is that the usefulness of a stylometric cue depends on semantic context. For instance, high repetition may be normal in poetry, suspicious in a news report, and ambiguous in a literary analysis. By conditioning the semantic representation on the style projection before final fusion, RADAR can learn interactions between genre, content, and linguistic form.

### 3.4. Classifier and inference

The fused representation is passed to a single linear classifier, producing one logit. The final score is obtained with the sigmoid function and interpreted as the probability that the text is machine-authored.

## 4. Training Protocol

All experiments used binary cross-entropy with logits. When the training split was imbalanced, we used a positive-class weight equal to the number of negative examples divided by the number of positive examples. Optimization used AdamW, a learning rate of $2 \times 10^{-5}$, cosine scheduling, batch size 32, evaluation batch size 64, seed 42, and 10 epochs. Evaluation was performed with the official PAN metric implementation exposed by the local evaluator package.

### 4.1. Main two-stage training

The primary RADAR model was trained in two stages. In the first stage, the ModernBERT encoder was frozen and only the stylometric projection, fusion module, and classifier were trained. This reduced the risk that randomly initialized fusion layers would immediately destabilize the pretrained encoder. The frozen-encoder stage also provided a diagnostic: if the model performs well with only the top layers trained, then the stylometric and fusion components are learning meaningful task information.

In the second stage, training resumed from the frozen-encoder checkpoint and the entire model was unfrozen. This end-to-end stage allowed ModernBERT representations to adapt to the combined

PAN+RAID distribution and to the stylometric fusion objective. We selected the best checkpoint according to the arithmetic mean of the PAN metrics.

### 4.2. Controlled variants

We trained two additional variants to analyze data-source effects. The PAN-only frozen-encoder model (RADAR-EF-PAN26) used the official PAN data without RAID augmentation, with the encoder kept frozen throughout. The RAID-only frozen-encoder model (RADAR-EF-RAID) used the filtered RAID development data with the encoder frozen, and was evaluated on the RAID held-out sample with an additional cross-domain check on the PAN validation split. These variants were not intended as final submissions; they were used to understand how much robustness came from the architecture and how much came from source diversity.

It is important to distinguish these frozen-encoder variants from Stage 1 of the main two-stage training. Stage 1 is a preliminary pass in which the encoder is frozen only to stabilize the newly initialized fusion and projection layers before end-to-end fine-tuning begins in Stage 2; it is therefore not an independently evaluated or submitted model. By contrast, RADAR-EF, RADAR-EF-PAN26, and RADAR-EF-RAID are models trained to convergence with the encoder permanently frozen—they constitute fully independent configurations evaluated and submitted on their own. The "EF" label (encoder-frozen) is used throughout the result tables to mark any variant that was trained entirely in this mode, while "RADAR" (without the EF suffix) denotes the full two-stage model that undergoes end-to-end fine-tuning in Stage 2.

## 5. Results and Discussion

All models were evaluated using the official PAN-2026 evaluation script, which reports five complementary metrics: ROC-AUC, Brier score complement, C@1, F1, and F0.5u. The arithmetic mean of these five metrics serves as the primary ranking criterion. The Brier column below reports the PAN Brier-score complement, so all metric columns are higher-is-better. Evaluations were conducted on two splits: the official PAN-2026 validation set (3,589 samples, in-domain) and a held-out RAID test set (10,000 balanced samples, out-of-domain).

### 5.1. Baseline

For reference, the TF-IDF SVM baseline provided by the task organizers was evaluated in our notebooks on the combined validation set of 18,423 samples. It achieved a mean score of 0.746. The Binoculars zero-shot approach [18] reached a mean of 0.877 on a separate split, while the supervised TF-IDF SVM on the official PAN split reached 0.978. Because these baseline figures were produced under different validation compositions, they are used only as orientation points rather than strictly paired comparisons with Table 2.

### 5.2. Main Results

Table 2 summarizes the evaluation results for all experimental configurations.

The single-source variants highlight a clear domain-mixing effect. RADAR-EF-PAN26 performs strongly on the official PAN validation split but drops on the RAID held-out test set, whereas RADAR-EF-RAID performs well on RAID but transfers less effectively to PAN. The combined-data variants reduce this cross-domain gap, and the full two-stage RADAR model achieves the highest scores on both internal sets. This pattern supports the use of RAID as a robustness regularizer while preserving the task-specific PAN distribution as the central target.

**Table 2**
Evaluation results for all RADAR variants on two test sets. **Bold** values indicate the best result per column. *PAN26-Val*: official PAN-2026 validation split (3,589 samples). *RAID-Test*: balanced out-of-domain test set (10,000 samples). Brier denotes the PAN Brier-score complement.

| Model | Test Set | ROC-AUC | Brier comp. | C@1 | F1 | F0.5u | Mean |
|---|---|---|---|---|---|---|---|
| RADAR-EF-RAID | PAN26-Val | 0.795 | 0.651 | 0.644 | 0.784 | 0.694 | 0.714 |
| | RAID-Test | 0.974 | 0.941 | 0.920 | 0.918 | 0.935 | 0.938 |
| RADAR-EF-PAN26 | PAN26-Val | 0.997 | 0.983 | 0.979 | 0.984 | 0.982 | 0.985 |
| | RAID-Test | 0.716 | 0.731 | 0.650 | 0.659 | 0.649 | 0.681 |
| RADAR-EF | PAN26-Val | 0.997 | 0.982 | 0.976 | 0.982 | 0.978 | 0.983 |
| | RAID-Test | 0.976 | 0.940 | 0.919 | 0.921 | 0.941 | 0.933 |
| **RADAR** | PAN26-Val | **1.000** | **0.997** | **0.996** | **0.997** | **0.996** | **0.997** |
| | RAID-Test | **0.990** | **0.967** | **0.964** | **0.964** | **0.974** | **0.972** |

*Note:* The in-domain PAN-2026 validation performance (mean = 0.997, ROC-AUC = 1.000) requires cautious interpretation due to the distributional similarity between the training and validation data. The out-of-domain RAID result (mean = 0.972) offers a more realistic estimate of model robustness.

## 5.3. Official Platform Evaluation

Table 3 reports the evaluation scores produced by the TIRA platform on the official PAN-2026 challenge test set (`pan26-generative-ai-detection-20260507-test`), which was withheld from participants and contains unseen generators and novel obfuscation strategies [2].

**Table 3**
Official TIRA evaluation on `pan26-generative-ai-detection-20260507-test`. All metrics are higher-is-better. Brier denotes the PAN Brier-score complement.

| Variant | ROC-AUC | Brier | C@1 | F1 | F0.5u | Mean |
|---|---|---|---|---|---|---|
| **RADAR-EF** | **0.772** | **0.805** | **0.722** | **0.800** | **0.839** | **0.788** |
| RADAR-EF-RAID | 0.701 | 0.792 | 0.694 | 0.783 | 0.812 | 0.756 |
| RADAR-EF-PAN26 | 0.673 | 0.658 | 0.545 | 0.608 | 0.736 | 0.644 |
| RADAR (E2E) | 0.790 | 0.541 | 0.530 | 0.555 | 0.741 | 0.632 |

The official results reveal a notable reversal compared to the internal validation findings: RADAR-EF, the encoder-frozen model trained on the combined PAN+RAID data, achieves the highest mean score of 0.788, while the end-to-end fine-tuned RADAR model—which dominated on the internal validation split—falls to a mean of only 0.632. Several factors jointly explain this outcome.

**Generalization versus overfitting.** Full fine-tuning allows ModernBERT's representations to shift substantially toward the training distribution. While this improves in-sample accuracy, it also makes the model sensitive to the statistical fingerprints of the generators and obfuscation types seen during training. The official test set deliberately contains novel generators and previously unseen attack strategies [2], which produce text that lies outside the distribution RADAR was optimized for. The frozen encoder, by contrast, retains the broad general-purpose representations learned during large-scale pre-training—representations that encode diverse writing patterns across many domains and styles. This pre-trained knowledge acts as a form of implicit regularization, preventing the model from collapsing onto shortcuts that are specific to the training generators.

**Calibration under distributional shift.** The Brier complement for RADAR on the official test drops sharply to 0.541, a marked deterioration from 0.997 on the internal PAN-2026 validation split. This

collapse indicates that the end-to-end model assigns high-confidence predictions that are frequently incorrect when the test distribution shifts—a common symptom of overconfident fine-tuning. RADAR-EF, by contrast, retains a Brier complement of 0.805, demonstrating substantially better-calibrated scores under distributional shift.

**Role of the stylometric branch.** Because the stylometric features are explicitly designed as generator-agnostic linguistic indicators, they capture properties of writing style that remain meaningful across generator families. By training only the stylometric projection, fusion module, and classifier, RADAR-EF learns a compact, task-relevant adapter on top of robust frozen encoder representations. This narrower parameter space is less prone to catastrophic forgetting of general language knowledge and provides a more stable foundation when facing text from generators not represented during training.

These findings underscore a fundamental tension in transfer learning for detection tasks: more end-to-end fine-tuning improves fit to the training distribution but can reduce robustness to distributional shift. For real-world deployment—where new generators and obfuscation strategies emerge continuously—the frozen-encoder approach offers a favorable robustness–accuracy trade-off.

## 6. Conclusion

This paper presented RADAR, a hybrid AI text detection system that combines a large pretrained transformer encoder with handcrafted stylometric features through a learned cross-attention-style fusion mechanism. Three design principles shaped the system: the use of complementary signal types (neural semantics and symbolic stylometry), a two-stage training curriculum that stabilizes learning before committing to full fine-tuning, and a diverse training corpus spanning multiple genres, attack types, and generator families.

The experiments show that domain mixing is a critical factor for robustness: models trained on a single data source generalize poorly to out-of-distribution examples, whereas the combined-dataset model achieves strong results across both in-domain and adversarially obfuscated evaluation sets. On the official PAN-2026 challenge test set, the encoder-frozen variant RADAR-EF achieved the highest mean score of 0.788, outperforming the end-to-end fine-tuned model (mean 0.632). This outcome highlights a key practical insight: preserving pre-trained encoder representations through encoder freezing improves generalization to unseen generators and obfuscation strategies, whereas full end-to-end fine-tuning increases the risk of overfitting to the training distribution under the conditions of the PAN 2026 task.

Future work could explore replacing the rule-based stylometric extractor with a lightweight trained component to improve feature quality on short texts, and could investigate curriculum learning with progressively harder obfuscation examples to further narrow the gap between in-domain and out-of-domain performance. Parameter-efficient adaptation techniques such as LoRA could also be explored as an intermediate approach between fully frozen and fully fine-tuned encoder training, potentially offering better generalization than end-to-end optimization while still adapting the encoder to the target task.

## Acknowledgments

The authors thank the PAN 2026 organizers for preparing the Voight-Kampff Generative AI Detection task [1, 2] and the maintainers of the open-source libraries used in this work, including PyTorch, Transformers, Datasets, and the PAN evaluation package.[19, 20]

## Declaration on Generative AI

During the preparation of this work, generative AI assistance was used for drafting, language polishing, and LaTeX editing. The authors reviewed, revised, and approved the final content and take full responsibility for the publication's content.

## References


[1] J. Bevendorff, M. Fröbe, A. Greiner-Petter, A. Jakoby, M. Mayerl, P. Nakov, H. Plutz, M. Potthast, B. Stein, M. N. Ta, Y. Wang, E. Zangerle, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, A. Barrón-Cedeño, A. G. S. de Herrera, E. S. Salido, S. MacAvaney, J. M. Struß (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2026.

[2] J. Bevendorff, R. R. Gunti, J. Karlgren, M. Fröbe, M. Potthast, B. Stein, Overview of the third "Voight-Kampff" Generative AI / LLM Detection Task at PAN and ELOQUENT 2026, in: A. Barrón-Cedeño, A. G. S. de Herrera, E. S. Salido, S. MacAvaney, J. M. Struß (Eds.), Working Notes of CLEF 2026 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[3] Gemini Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al., Gemini: A Family of Highly Capable Multimodal Models, arXiv preprint arXiv:2312.11805 (2023). URL: https://arxiv.org/abs/2312.11805.

[4] OpenAI, GPT-4 Technical Report, Technical Report, OpenAI, 2023. URL: https://arxiv.org/abs/2303.08774.

[5] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open Foundation and Fine-Tuned Chat Models, arXiv preprint arXiv:2307.09288 (2023). URL: https://arxiv.org/abs/2307.09288.

[6] E. Clark, T. August, S. Serrano, N. Haduong, S. Gururangan, N. A. Smith, All That's "Human" Is Not Gold: Evaluating Human Evaluation of Generated Text, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, 2021, pp. 7282–7296. URL: https://aclanthology.org/2021.acl-long.565. doi:10.18653/v1/2021.acl-long.565.

[7] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, M. McCain, A. Newhouse, J. Blazakis, K. McGuffie, J. Wang, Release Strategies and the Social Impacts of Language Models, arXiv preprint arXiv:1908.09203 (2019). URL: https://arxiv.org/abs/1908.09203.

[8] S. Gehrmann, H. Strobelt, A. M. Rush, GLTR: Statistical Detection and Visualization of Generated Text, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, 2019, pp. 111–116. URL: https://aclanthology.org/P19-3019. doi:10.18653/v1/P19-3019.

[9] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, C. Finn, DetectGPT: Zero-Shot Machine-Generated Text Detection Using Probability Curvature, in: Proceedings of the 40th International Conference on Machine Learning, Proceedings of Machine Learning Research, PMLR, 2023. URL: https://proceedings.mlr.press/v202/mitchell23a.html.

[10] G. Jawahar, M. Abdul-Mageed, L. V. S. Lakshmanan, Automatic Detection of Machine Generated Text: A Critical Survey, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, 2020, pp. 2296–2309. URL: https://aclanthology.org/2020.coling-main.208. doi:10.18653/v1/2020.coling-main.208.

[11] K. Krishna, Y. Song, M. Karpinska, J. Wieting, M. Iyyer, Paraphrasing Evades Detectors of AI-Generated Text, But Retrieval Is an Effective Defense, in: Advances in Neural Information Processing Systems, 2023. URL: https://proceedings.neurips.cc/paper_files/paper/2023/hash/575c450013d0e99e4b0ecf82bd1afaa4-Abstract-Conference.html.

[12] C. Rudin, Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead, Nature Machine Intelligence 1 (2019) 206–215. URL: https://doi.org/10.1038/s42256-019-0048-x. doi:10.1038/s42256-019-0048-x.

[13] P. Juola, Authorship Attribution, Foundations and Trends in Information Retrieval 1 (2006) 233–334. URL: https://doi.org/10.1561/1500000005. doi:10.1561/1500000005.

[14] M. Coulthard, A. Johnson, D. Wright, An Introduction to Forensic Linguistics: Language in Evidence, 2nd ed., Routledge, London, 2017.

[15] L. Dugan, A. Hwang, F. Trhlik, J. M. Ludan, A. Zhu, H. Xu, D. Ippolito, C. Callison-Burch, RAID: A Shared Benchmark for Robust Evaluation of Machine-Generated Text Detectors, arXiv preprint arXiv:2405.07940 (2024). URL: https://arxiv.org/abs/2405.07940.

[16] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, et al., Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference, arXiv preprint arXiv:2412.13663 (2024). URL: https://arxiv.org/abs/2412.13663.

[17] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[18] 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 2024), PMLR, 2024. URL: https://arxiv.org/abs/2401.12070.

[19] 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: Generative AI Authorship Verification, Multi-Author Writing Style Analysis, Multilingual Text Detoxification, and Generative Plagiarism Detection, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2025), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2025.

[20] 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.