Enhancing AI Text Detection with Frozen Pretrained Encoders and Ensemble Learning

Notebook for the PAN Lab at CLEF 2025

Shushanta Pudasaini[1], Luis Miralles-Pechuán[1], David Lillis[2] and Marisa Llorens Salvador[1]

[1]Technological University Dublin, Dublin, Ireland
[2]University College Dublin, Dublin, Ireland

Abstract

As AI systems become increasingly capable of generating text, distinguishing it from human-written content remains an ongoing research challenge. This paper proposes a simple yet effective ensemble-based approach for detecting AI-generated text using pre-trained encoders. Six different Large Language Models (LLMs) were fine-tuned with the PAN CLEF 2025 training set, and six ensemble learning approaches were applied on top of the five best-performing LLMs. These models were evaluated on the PAN CLEF validation dataset and a subset of the COLING 2025 dataset to ensure the model's performance across multiple datasets and domains. Experiments on benchmark datasets show that ensemble approaches significantly outperform individual models, achieving improved F1 scores and robustness across diverse AI-generated text samples. The best configuration (Bagging with support vector classifier on top of the results achieved from the top 5 performing individual LLMs) was able to achieve an F1 score of 0.9886 on the PAN CLEF 2025 benchmark compared to the F1 score of 0.9767 from the individual Deberta-v3-large model on the same benchmark dataset. Likewise, the preservation of pre-trained knowledge through frozen encoder layers consistently improved detection performance, demonstrated by the Deberta-v3-large model's 2.67% F1 scores improvement compared to its fully fine-tuned version. From this research, ensemble learning algorithms applied on top of LLMs were found to improve the performance of the AI-generated text detection task as experimented in the Voight-Kampff Generative AI Detection 2025 [1], which was a part of the PAN at CLEF 2025 [2] submission made through the TIRA platform [3]. The research is publicly available on GitHub under https://github.com/ShushantaTUD/Ensemble-Based-AI-Generated-Text-Detection.

Keywords

Large Language Models, AI Generated Text Detection, Ensemble Learning, Machine Learning, Encoders, Ensemble Learning,

1. Introduction

AI-generated text refers to content generated by Large Language Models (LLMs) like ChatGPT, which are trained on large datasets of human-written text. These LLMs can create essays, articles, and even research papers that mimic human writing styles, making it difficult to distinguish them from human-written content [4]. With the rapid development and availability of LLMs to the general public, their effects have reached across education, professional, and personal contexts, raising important questions about originality, authenticity, and intellectual integrity.

The educational sector faces challenges from AI-generated text, as it leads to AI-based plagiarism. Students submit partially or completely generated assignments using AI tools and use these AI tools to generate answers during online examinations, creating an unfair learning environment [4].

The ability of AI systems to generate convincing fake news articles, social media posts, and technical content is creating information disorder and reducing trust in legitimate sources. LLMs can bring inaccuracies, fabricated citations, or flawed reasoning that humans cannot detect [5]. As AI systems become increasingly capable of generating text, developing robust and adaptable detection systems

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain
*Corresponding author.
†These authors contributed equally.
D23129142@mytudublin.ie (S. Pudasaini); luis.miralles@TUDublin.ie (L. Miralles-Pechuán); david.lillis@ucd.ie (D. Lillis); marisa.llorens@TUDublin.ie (M. L. Salvador)
0000-0003-1612-4510 (S. Pudasaini)

is not just a technical challenge but a necessary step to maintain trust, fairness and the integrity of information in society.

Existing methods for detecting AI-generated text involve various strategies, such as supervised detection, zero-shot detection, retrieval-based detection, watermarking methods, and discriminating features [6]. Supervised detection involves models that are fine-tuned on AI-generated and human-written text. This approach typically requires large datasets, making it difficult to collect sufficiently large and diverse sample collections. Another approach to detecting AI-generated text is zero-shot detection, which uses pre-trained algorithms, eliminating the process of collecting a large dataset [7]. Retrieval-based detection is another method of detecting AI-generated text. This method compares the semantic similarities of the given text with predefined AI-generated text. It relies heavily on an extensive and up-to-date database of AI-generated texts.

AI-generated texts can be embedded with a model signature that is invisible to the human eye, allowing it to be detected only by a computer. This method is known as watermarking. Another approach to detecting AI-generated text involves identifying distinctive traits that classify AI-generated texts and human-written texts, such as statistical features or linguistic features [6]. Despite these various detection methods, the evolving capabilities of AI language models continue to present challenges for reliable detection, highlighting the need for ongoing research and development of more robust identification techniques.

The difficulty in detecting AI-generated text stems from the basic architecture of LLMs, which is optimised to generate text that mimics human-written text. LLMs are trained on vast amounts of human-written text, making AI-generated texts almost indistinguishable from human-written texts [4]. Current AI detection tools show limited effectiveness. OpenAI's detector properly identifies only 26% of AI-generated texts, indicating the technical complexity of this task [8].

Detecting AI-generated text is more complicated as language models evolve rapidly, while detection tools rely on outdated methods and data [9]. Because detection methods cannot be tested until the new LLMs are launched, they always go one step behind. Simple techniques like paraphrasing AI-generated text can easily bypass many detectors [10]. Several challenges further complicate the identification of AI-generated text: the absence of standardized benchmarks for evaluating detection accuracy, the high computational cost of these tools, inherent biases that may unfairly flag texts written by non-native English speakers as AI-generated, the rapid advancement of large language models (LLMs) that outpaces detector development, and their susceptibility to adversarial attacks [9, 11].

To address the limitations of individual models, ensemble learning has become a powerful strategy for enhancing the detection of AI-generated content (AIGC). Ensemble methods combine the strengths of multiple models, each capable of identifying different patterns or compensating for the weaknesses of others. By aggregating predictions through voting, averaging, or weighted combinations, ensemble approaches help reduce errors caused by model bias (oversimplification) or variance (over-sensitivity to data). As a result, the detection of AI-generated text becomes more accurate, robust, and reliable [4].

Ensemble methods use techniques such as bagging and boosting. This collective approach is especially valuable for complex detection tasks where single models struggle to capture all relevant features, and research suggests that ensembles are more resilient to adversarial attacks and generalise better between different types of AI-generated content [4]. Thus, ensemble learning offers a practical path forward in addressing the technical and evolving challenges of detecting AI-generated text.

2. Literature Review

The challenge of distinguishing human-written text from machine-generated content has grown rapidly with the widespread use of large language models. However, this problem did not emerge suddenly; it evolved from earlier research in related areas such as plagiarism detection.

Early approaches to detecting machine-generated text were inspired by plagiarism detection methods. Techniques such as part-of-speech (POS) tag n-grams and perplexity-based measures were used to identify paraphrased or automatically rewritten content. These models performed well in texts with

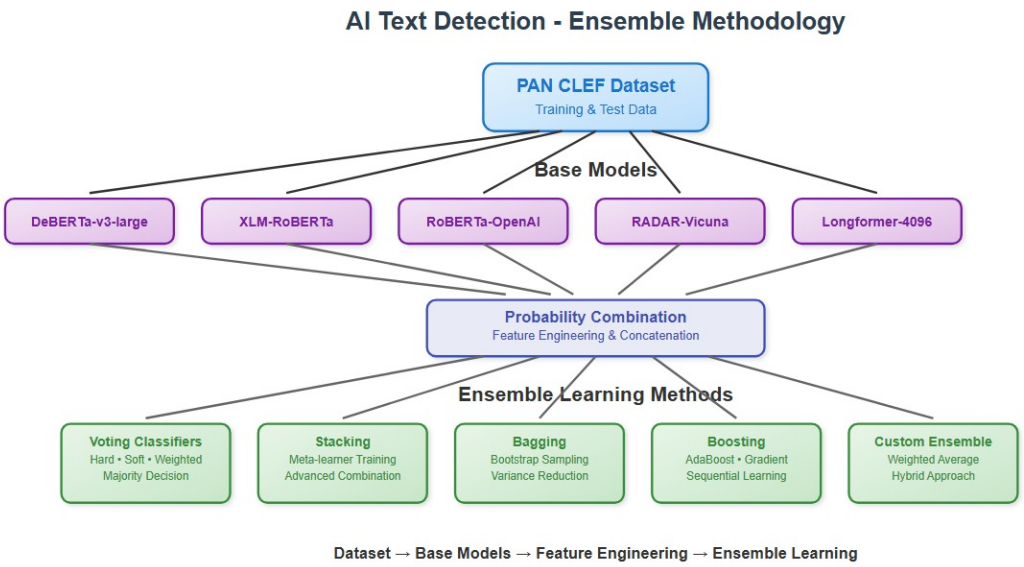

Figure 1: Overall Methodology block diagram including the base models and ensemble algorithms applied on the base models in the experiment

high and low levels of obfuscation and achieved competitive results using the Plagdet evaluation metric [12].

As research progressed, more efficient models were proposed. One such model was the Weighted Neural Bag-of-n-grams (WNB-ngram), a lightweight neural network designed for text classification tasks. It performed well on datasets like Yelp Reviews and IMDB, demonstrating that even small models could capture meaningful linguistic patterns [12].

With the rise of deep learning, researchers began framing AI text detection as a classification problem. Large pre-trained language models such as RoBERTa and DeBERTa were fine-tuned on specially curated datasets to detect subtle patterns in text that distinguish between human and AI-written content [13].

In a different approach, Harika Abburi et al. [14] applied classical machine learning techniques like Gradient Boosting, Stacking, and Voting. Instead of raw text, these models used the probability outputs from various pre-trained LLMs as input features. Their system achieved high performance in the AuTexTification shared task, ranking first in model attribution for both English and Spanish texts.

In parallel, real-world tools for AI text detection started to appear. OpenAI released its classifier in 2023, but it was later discontinued due to poor accuracy on short or factual inputs. Other tools, such as GLTR, GPTZero, and DetectGPT, experimented with analysing token-level likelihoods and distribution shifts to identify AI-generated text [13].

Together, these developments show how the field has moved from early rule-based techniques to classical machine learning, and now to advanced fine-tuned language models. Despite these advances, reliable AI text detection, especially in open-domain settings, remains an ongoing research challenge.

3. Methodology

Our study proposes a simple yet effective ensemble-based approach for detecting AI generated text using large language models. The methodology consists of four main components: dataset preparation, base model selection, feature engineering and ensemble learning. The overall methodology of the experiment is represented in Fig. 1.

Datasets for Training and Evaluation

The benchmark datasets used contain both human-written and AI-generated texts. These include data sets from COLING-2025 and PAN CLEF, which provide labelled samples in English, and each data

set is preprocessed using tokenisers from different models. We split the training dataset into training (80%), validation (20%), and made predictions on the test set.

Table 1
Training and Testing Datasets with Observation Counts

Dataset	Split	Number of Observations
PAN CLEF 2025	Training Set	23,707
PAN CLEF 2025	Testing Set	3,589
COLING 2025	Testing Set	3,000

Base LLMs

The following LLMs were used in the experiment:

- microsoft/deberta-v3-large [15]
- FacebookAI/XLM roberta-large [16]
- openai-community/roberta-large-openai-detector [17]
- lmsys/vicuna-7b-v1.5 (RADAR-vicuna)[18]
- google-bert/bert-base-multilingual-cased [19]
- allenai/longformer-base-4096

The selection of models is based on the results of the experiment conducted by Harika Abburi et al. [4]. In addition, the choice to include models like Deberta-v3-large and RADAR-Vicuna-7B is due to their strong performance in the classification task.

Ensemble Techniques

Six different ensemble techniques were implemented to improve the performance of AI-generated text detection. These included a Voting Classifier, a Stacking Classifier, and a Gradient Boosting Classifier. Instead of using raw text features, these classifiers were trained on the class probability scores generated by large language models for each text sample. The six ensemble approaches used in this paper are:

1. Custom Ensemble
   The custom ensemble first trains multiple models (Random Forest, XGBoost, LightGBM) and evaluates their performance using cross-validation. It then assigns weights to each model based on their cross-validation scores - the better-performing models get higher weights. The final prediction is made by taking a weighted average of all model predictions.
2. Bagging (Decision Tree Classifier)
   This technique works by leveraging the principle of Bootstrap Aggregating to reduce the variance of a base model, which is the Decision Tree in this case. It generates multiple versions of the training dataset by sampling with replacement (bootstrap), ensuring that each base estimator is trained on a slightly different subset of the data.
3. Bagging (SVC)
   By bagging with SVC, multiple SVC models are trained on different bootstrap samples of the data. The final prediction combines all SVC predictions through majority voting, reducing the variance while maintaining SVC's strong classification boundaries.
4. Voting (Soft)
   In soft voting, the final prediction is made by averaging the predicted probabilities of all models and choosing the class with the highest average probability.
5. Gradient Boosting Classifier
   Gradient boosting builds models sequentially, where each new model learns to correct the errors made by the previous models. It starts with a simple model and iteratively adds new models that focus on the misclassified examples from previous iterations. Each subsequent model is trained on the residual errors.

6. Stacking (Random Forest, XGBoost, LightGBM, CatBoost; Final Model: Gradient Boosting Classifier)
   Stacking uses a two-level approach where the first level trains multiple diverse models (Random Forest, XGBoost, LightGBM, CatBoost) on the training data. A meta-learner (Gradient Boosting Classifier) is then trained on the predictions of these first-level models to learn how to best combine their outputs.

The models were evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score.

4. Results

In this section, we present and analyse the experimental findings of our AI-generated content detection research. Our evaluation demonstrates the effectiveness of individual transformer-based models and ensemble methods across multiple benchmarks. The results highlight significant performance differences between individual models and ensemble strategies, providing valuable insights for developing robust detection systems for AI-generated text.

The following results are organised to provide clear performance comparisons between different detection approaches. First, we analyse the performance metrics of standalone transformer-based architectures to establish baseline capabilities. Then, we explore how combining these models through various ensemble techniques affects detection performance. The analysis includes both standard ensemble methods applied to all models and specialised ensembling of only top-performing models to determine optimal integration strategies.

4.1. Results from Individual Models

Initially, experiments were performed to evaluate whether the fully fine-tuned LLM or fine-tuning while keeping the first five layers frozen, preserving the pre-trained knowledge, would achieve good results in the PAN 2025 dataset. The Deberta-v3-large model with frozen first five encoder layers (0.8347) outperformed its fully fine-tuned model (0.8080), suggesting that preserving pre-trained knowledge improves detection performance. The comparison of the performance of fine-tuning on the full model and fine-tuning keeping some layers frozen, preserving pre-trained knowledge, is shown in Table 2.

Table 2
Model Accuracy Comparison with and without Frozen Encoder Layers on the COLING 2025 dataset

Model	Fine-tuning Strategy	Accuracy
DeBERTa-v3-large	Full fine-tuning	0.8080
RoBERTa (OpenAI Detector)	Full fine-tuning	0.7960
XLM-RoBERTa-large	Full fine-tuning	0.7293
DeBERTa-v3-large	First 5 layers frozen	0.8347
RoBERTa (OpenAI Detector)	First 5 layers frozen	0.7480
XLM-RoBERTa-large	First 5 layers frozen	0.7707

Following this the COLING 2025 benchmark was used to test the models. On this dataset, the longformer model achieved the highest F1 score of 0.8377, showing excellent detection capabilities. The Roberta-large model achieved the lowest performance among the transformer models with an F1 score of 0.7293. The average F1 score across all individual models on this benchmark was 0.748, achieved by Xlm-roberta (frozen first five encoder layers) and Roberta-openai-detector (frozen first five encoder layers).

Deberta-v3-large achieved the top performance with an F1 score of 0.9767 for the PAN CLEF benchmark. The Vicuna-7b model and the Roberta-large model also displayed robust capabilities with F1

Table 3
Evaluation of individual fine-tuned LLMs on COLING 2025 Test set and PAN CLEF 2025 Validation set

Model	F1 (COLING 2025)	F1 (PAN CLEF 2025)
Deberta-v3-large	0.8080	**0.9767**
roberta-openai-detector	0.796	0.9630
roberta-large	0.7293	0.9654
bert-base-multilingual-cased	0.7794	0.9582
longformer	**0.8377**	0.9580
vicuna-7b	0.8216	0.9751

Table 4
Evaluation of Ensemble Learning Algorithms on COLING 2025 Test Set and PAN CLEF 2025 Validation Set

Model	F1 (COLING 2025)	F1 (PAN CLEF 2025)
Bagging (Decision Tree Classifier)	**0.8399**	0.9876
Bagging (SVC)	0.8324	**0.9886**
Voting (Soft)	0.8324	0.9876
Gradient Boosting Classifier	0.8249	0.9876
Stacking (RF, XGB, LGB, CatBoost; Final: GBC)	0.8199	0.9866

scores of 0.9751 and 0.9654, respectively. For the PAN CLEF dataset, the longerformer model showed the lowest performance with a score of 0.9580. The average F1 score across all models for this benchmark was 0.9582, achieved by bert-base-multilingual-cased. Table 3 shows the results obtained using COLING 2025 and the PAN CLEF 2025 validation datasets.

4.2. Results after the Ensemble Approach

After applying ensemble methods to the COLING 2025 Test Set, Bagging with Decision Tree Classifier achieved the highest F1 score of 0.8399. Both Bagging with SVC and soft Voting methods achieved similar scores of 0.8324, while Stacking achieved the lowest performance with an F1 score of 0.8199. The average F1 score across all ensemble methods was 0.832449.

When evaluating ensemble techniques using only the top 4 performing models on the PAN CLEF validation set, Bagging with SVC presented the best performance with an F1 score of 0.9886. Gradient Boosting Classifier, soft voting and bagging with Decision Tree Classifier showed similar performance with F1 scores of 0.9876. Stacking again produced the lowest results with an F1 score of 0.9866. The average performance across these top-model ensemble approaches was 0.9876, showing a clear improvement over ensembles using all models. The results for the ensemble learning algorithms can be seen in Table 4

Our results highlight that the optimal approach for AIGC detection involves combining strategically frozen transformer models with ensemble methods, particularly those using support vector classifiers with bagging. The performance gain achieved through the ensemble method indicates that different model architectures capture complementary linguistic features of AI-generated content, enabling more robust detection when combined.

The proposed ensemble methodology can be effectively deployed in various applications requiring reliable AI content detection, including academic integrity systems, news verification platforms, and social media content moderation. Its high performance on diverse benchmarks suggests strong generalisation capability across different types of AI-generated text, making it particularly valuable for educational institutions and media organisations needing to distinguish between human and machine-written content.

4.3. Ensemble Learning Full Result on PAN CLEF Dataset

The full metric results of the ensemble learning approaches applied on the PAN CLEF 2025 dataset have been represented in the Table 5.

Table 5
Full metrics results of the ensemble learning algorithms applied on the PAN CLEF dataset

Model	Accuracy	ROC AUC	1 - Brier	C@1	F1	F0.5u	Mean Metric
Custom ensemble	0.9847	0.9974	0.9860	0.9847	0.9887	0.9893	0.9892
Soft voting	0.9833	0.9974	0.9864	0.9833	0.9877	0.9889	0.9887
Gradient boosting classifier	0.9833	0.9967	0.9862	0.9833	0.9877	0.9889	0.9885
Bagging DT	0.9833	0.9966	0.9866	0.9833	0.9877	0.9889	0.9886
Advanced stacking (GB Tree)	0.9805	0.9974	0.9851	0.9805	0.9856	0.9868	0.9871
Bagging SVC	0.9847	0.9800	0.9860	0.9847	0.9886	0.9917	0.9862

5. Discussion and Analysis of the Results

The experimental results reveal several key patterns in model performance across different test datasets. The longformer model demonstrated superior effectiveness with the highest F1 score (0.8377) on the COLING 2025 benchmark, establishing it as the most reliable detector in our evaluation. Notably, the Deberta-v3-large model with frozen first five encoder layers (0.8347) significantly outperformed its fully fine-tuned counterpart (0.8080), suggesting that preserving pre-trained knowledge structures enhances detection capabilities.

Performance consistency varied considerably across models when tested on different datasets. While some models maintained relatively stable performance metrics, other models showed noticeable changes, indicating sensitivity to dataset-specific characteristics. This variability shows the importance of comprehensive evaluation across diverse test conditions when deploying AI-generated text detection systems in real-world applications.

Based on the experiments, this paper concludes the following insights:

- **Freezing encoder layers improved detection performance.** Models with the first five encoder layers frozen consistently outperformed their fully fine-tuned counterparts across multiple architectures. For instance, DeBERTa-v3-large demonstrated a performance gain of approximately 2.67%. This suggests that retaining the linguistic knowledge embedded during pre-training—while allowing higher layers to adapt to the detection task—results in a more effective framework for distinguishing between human and AI-generated content.
- **Domain-specific performance gaps reveal detection challenges.** Model performance varied significantly across texts from two datasets: PAN CLEF 2025 and COLING 2025. This dataset and domain sensitivity highlight the need for either domain-specific fine-tuning or ensemble methods that incorporate specialised detectors tailored to different content types.
- **Ensemble approaches improved classification performance.** Both bagging and stacking ensemble techniques were evaluated across two benchmark datasets—COLING 2025 and PAN CLEF 2025. On the COLING 2025 test set, individual fine-tuned LLMs achieved an average F1 score of 0.7953, with the best-performing model being Longformer (F1 = 0.8377). In comparison, ensemble models achieved a higher average F1 score of 0.8299, with the best ensemble method—Bagging (Decision Tree Classifier)—reaching 0.8399. This marks an approximate relative improvement of 4.3% over the average individual model and a marginal gain over the top single model.
  On the PAN CLEF 2025 validation set, the advantage of ensembles was even more pronounced. The average F1 score of individual fine-tuned models was 0.9660, while ensemble methods achieved an average of 0.9878. The best ensemble—Bagging (SVC)—achieved an F1 of 0.9886, outperforming the top individual model (DeBERTa-v3-large, F1 = 0.9767) by 1.2 percentage points. These results

demonstrate that ensemble methods not only improve generalisation but also help reduce the variance and overfitting tendencies of individual models, especially in high-stakes classification tasks.

6. Conclusion

This research demonstrates the effectiveness of ensemble learning approaches for AI-generated text detection. Our findings show that strategically organised ensemble methods significantly outperform individual models, with the best configuration (Bagging with SVC using top 4 models) achieving an F1 score of 0.87248 on the COLING 2025 benchmark, 3.5% improvement over the best single model. The preserved pre-trained knowledge through frozen encoder layers consistently enhanced detection performance, demonstrated by the Deberta-v3-large model's 2.67% F1 score improvement compared to its fully fine-tuned version.

The best performance of ensembles, particularly when combining only top-performing models, confirms that different architectures capture complementary linguistic patterns distinguishing AI-generated from human-written text. Despite these achievements, our experiment identifies important challenges, including cross-domain performance variability and the need for continuous adaptation to evolving language models. Future work should focus on developing adaptive ensemble approaches, exploring domain-specific detection modules, and investigating interpretability methods to enhance trust in these systems for educational and professional applications.

7. Acknowledgments

This publication has emanated from research conducted with the financial support of/supported in part by a grant from Science Foundation Ireland under Grant number 18/CRT/6183. For Open Access, the author has applied a CC by public copyright licence to any Author Accepted Manuscript version arising from this submission.

8. Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT and Grammarly to: Grammar and spelling check. Further, the author(s) used Claude for Figure 1 to generate images. After using these tools (s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] 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.

[2] 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.

[3] 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: 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.

[4] K. C. Fraser, H. Dawkins, S. Kiritchenko, Detecting ai-generated text: Factors influencing detectability with current methods, Journal of Artificial Intelligence Research 82 (2025) 2233–2278.

[5] T. Karakose, M. Demirkol, R. Yirci, H. Polat, T. Y. Ozdemir, T. Tülübaş, A conversation with chatgpt about digital leadership and technology integration: Comparative analysis based on human–ai collaboration, Administrative Sciences (2023). URL: https://api.semanticscholar.org/CorpusID:259737796.

[6] S. Abdali, R. Anarfi, C. Barberan, J. He, Decoding the ai pen: Techniques and challenges in detecting ai-generated text, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 6428–6436.

[7] Hugging Face, Zero-shot object detection, 2025. URL: https://huggingface.co/docs/transformers/en/tasks/zero_shot_object_detection, accessed: 2025-05-30.

[8] M. Heikkilä, Why detecting ai-generated text is so difficult—and what to do about it, 2023. URL: https://www.technologyreview.com/2023/02/07/1067928/why-detecting-ai-generated-text-is-so-difficult-and-what-to-do-about-it/, accessed: 2025-05-30.

[9] Grammarly, Are ai detectors accurate? understanding their limitations, 2025. URL: https://www.grammarly.com/blog/ai/ai-detectors-accuracy/, accessed: 2025-05-30.

[10] M. P. et al., Simple techniques to bypass GenAI text detectors: implications for inclusive education - International Journal of Educational Technology in Higher Education — educationaltechnologyjournal.springeropen.com, https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-024-00487-w, 2024. [Accessed 01-07-2025].

[11] A. K. Kadhim, L. Jiao, R. Shafik, O.-C. Granmo, Adversarial attacks on ai-generated text detection models: A token probability-based approach using embeddings, https://arxiv.org/abs/2501.18998, 2025. ArXiv preprint arXiv:2501.18998.

[12] B. Li, Z. Zhao, T. Liu, P. Wang, X. Du, Weighted neural bag-of-n-grams model: New baselines for text classification, in: Y. Matsumoto, R. Prasad (Eds.), Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, The COLING 2016 Organizing Committee, Osaka, Japan, 2016, pp. 1591–1600. URL: https://aclanthology.org/C16-1150/.

[13] H. Abburi, M. Suesserman, N. Pudota, B. Veeramani, E. Bowen, S. Bhattacharya, Generative ai text classification using ensemble llm approaches, arXiv preprint arXiv:2309.07755 (2023).

[14] K. Yalcin, I. Cicekli, G. Ercan, An external plagiarism detection system based on part-of-speech (pos) tag n-grams and word embedding, Expert Systems with Applications 197 (2022) 116677.

[15] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021. arXiv:2111.09543.

[16] 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, CoRR abs/1907.11692 (2019). URL: http://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[17] 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).

[18] L. M. S. Organization, lmsys/vicuna-7b-v1.5 · Hugging Face — huggingface.co, https://huggingface.co/lmsys/vicuna-7b-v1.5, 2024. [Accessed 01-07-2025].

[19] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.