# Human or Not? Light-Weight and Interpretable Detection of AI-Generated Text

Notebook for the PAN Lab at CLEF 2025

Maximilian Seeliger[1,*,†], Patrick Styll[1,*,†], Moritz Staudinger[1] and Allan Hanbury[1]

[1]TU Wien Informatics, Favoritenstraße 9-11, 1040 Vienna, Austria

**Abstract**

Text generated by Large Language Models (LLMs) is becoming less distinguishable from their human-written counterparts. Reliable detection of the differences between the two is increasingly important to limit the spread of fake content, plagiarism and the manipulation of public opinion. We study the binary classification problem of distinguishing human-written from AI-generated text. We propose a two-step learning algorithm. In the first step, it calculates the correlation between the rows of the binary term-document matrix (TDM) and the binary labels associated with the documents. This step runs in $\mathcal{O}(nl_{\max} + nm)$ time, where $n$ is the number of texts, $l_{\max}$ is the maximum text length, and $m$ is the vocabulary size. In the second step, it uses these values to map any text to a sequence of correlations, which can be interpreted as a signal. This can be done in linear time $\mathcal{O}(l)$ where $l$ is the size of the text. Together with other statistical measurements, this signal serves as a feature for standard machine learning algorithms. Furthermore, we give a perspective on the interpretability of our proposed approach for global and local (instance-level) explanations. Our work demonstrates that while large language models like RoBERTa remain state-of-the-art in terms of raw accuracy for AI-text identification, our interpretable and computationally efficient approach offers a competitive alternative, particularly in scenarios where interpretability is important. We evaluate our approach within the Voight-Kampff Generative AI Detection task, which is part of the PAN lab at CLEF 2025.

**Keywords**

AI-Generated Text, Explainability, Signal Processing, PAN 2025

## 1. Introduction

With the advancement of Large Language Models (LLMs), generated texts are increasingly difficult to distinguish from their human counterparts [1]. This can pose risks to non-specialists readers, including but not limited to the spread of fake content, plagiarism, the publication of AI-written articles in scientific journals or the manipulation of public opinion [2, 3, 4]. As a consequence, there is an increased interest in automatic approaches capable of distinguishing machine-generated from human-written contents. Most of these approaches rely on computationally-expensive language model (LM) backbones, either using them directly [1] or exploiting certain statistical features, such as likelihood scores, that can be extracted from them [5, 6, 7, 8, 9]. Furthermore, since these methods depend on stochastic neural models, they are inherently non-interpretable. To address these pitfalls, we propose a new light-weight and interpretable method that relies only on statistical word-correlations rather than LM-backbones, but still achieves competitive performance.

To demonstrate the effectiveness of our approach, we use data from the Voight-Kampff Generative AI

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

*Corresponding author.

†These authors contributed equally.

maximilian.seeliger@tuwien.ac.at (M. Seeliger); patrick.styll@tuwien.ac.at (P. Styll); moritz.staudinger@tuwien.ac.at (M. Staudinger); allan.hanbury@tuwien.ac.at (A. Hanbury)

https://www.linkedin.com/in/maximilian-seeliger/ (M. Seeliger); https://www.linkedin.com/in/patrick-styll/ (P. Styll); https://informatics.tuwien.ac.at/people/moritz-staudinger (M. Staudinger); https://informatics.tuwien.ac.at/people/allan-hanbury (A. Hanbury)

0009-0000-8872-0624 (M. Seeliger); 0009-0009-6643-2512 (P. Styll); 0000-0002-5164-2690 (M. Staudinger); 0000-0002-7149-5843 (A. Hanbury)

Detection challenge, which is part of the PAN lab at CLEF 2025 [10, 11, 12]. The challenge is divided into two tasks: (1) binary classification of texts as either human- or AI-generated, and (2) multi-class classification estimating the degree of human or machine authorship in mixed-authorship texts. Each task consists of an individual dataset.

Our contributions include:

- A novel two-step learning algorithm that transforms text into a sequence of correlation values, interpretable as a signal.
- A collection of global and local interpretations based on the output of the learning algorithm.
- A simple approach to use hand-crafted linguistic features together with correlation signals, fed into a standard machine learning algorithm, to achieve competetive performance for distinguishing human-written from AI-generated text.

## 2. Main Method

We formally introduce the problem setting and propose the concept of *correlation signals* as well as a simple way to use them for classification. We use *binary term-document* matrices and the *Phi-coefficient* as fundamental building blocks to obtain a *word-correlation* value for each word. We map the words in a given text to their respective word-correlation and call this sequence a correlation signal.

### 2.1. Problem Setting

We study the problem of distinguishing human-written from AI-generated text in a supervised binary classification setting. Let $\mathcal{X}$ be the instance space, containing all possible texts, and let $\mathcal{Y} = \{0, 1\}$ denote the binary label space, where label 0 represents human-written text and label 1 AI-generated text. For training, we get a set of $n$ labeled training instances $\{(T_i, y_i)\}_{i=1}^{n} \subseteq \mathcal{X} \times \mathcal{Y}$ and try to find a function $f : \mathcal{X} \to \mathcal{Y}$ that correctly classifies unseen instances.

Let $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$ be the set of texts from the training data. We consider each text $T_i$ as a sequence of word tokens $(w_1, w_2, \ldots, w_l)$, resulting from tokenization (cf. Section 4.3), and expand the notation of set inclusion to allow $w \in T_i$ to denote that the word $w$ is contained at any position in the text $T_i$. We define the vocabulary of $\mathcal{T}$ as $\text{Vocab}(\mathcal{T}) = \{w \mid w \in T_i \text{ for all } T_i \in \mathcal{T}\}$ and say that $m = |\text{Vocab}(\mathcal{T})|$ is the number of words in the text corpus.

### 2.2. Correlation Signals

We construct a binary term-document matrix $\mathbf{B}$ from $\mathcal{T}$, where a row represents for a specific word the inclusion relation to each text from the training dataset.

**Definition 1.** *A **binary term-document matrix** $\mathbf{B} \in \{0, 1\}^{m \times n}$ indicates at position $\mathbf{B}_{i,j}$ whether a word $w_i \in \text{Vocab}(\mathcal{T})$ is contained in document $T_j$ for $1 \leq i \leq m$ and $1 \leq j \leq n$:*

$$\mathbf{B}_{i,j} = \begin{cases} 1 & \text{if } w_i \in T_j \\ 0 & \text{otherwise} \end{cases}$$

Given the $i$'th row $\mathbf{B}_{i,\cdot} \in \{0, 1\}^n$ and the label vector $\mathbf{y} = (y_1, y_2, \ldots, y_n)$, we are interested in quantifying the predictive power that the occurrence of the word $w_i$ has (i.e. which label is more likely, after knowing that $w_i$ occurs in the text). For this, we calculate the correlation between these two vectors.

**Definition 2.** *The **Phi-coefficient** [13] (also known as Matthews correlation coefficient) is a special case of the Pearson correlation coefficient for binary vectors. Given two binary vectors* $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$ *it is defined as*

$$\varphi(\mathbf{x}, \mathbf{y}) = \frac{\frac{1}{n} \cdot \sum_{i=1}^{n} x_i y_i - \bar{x}\bar{y}}{\sqrt{\bar{x}(1-\bar{x}) \cdot \bar{y}(1-\bar{y})}}$$

*where*

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i.$$

This leads to the definition of word-correlations. For a word $w_i$, represented in the $i$'th row of the term-document matrix, we denote its word-correlation with the function $\varphi(w_i) = \varphi(\mathbf{B}_{i,\cdot}, \mathbf{y})$, where $\mathbf{y}$ is the label vector. We further extend this notation to texts and say that text $T = (w_1, w_2, \ldots, w_l)$ is mapped to its correlation signal with

$$\varphi(T) = \Big(\varphi(w_1), \varphi(w_2), \ldots, \varphi(w_l)\Big)$$

For the given corpus $\mathcal{T}$ of size $|\mathcal{T}| = n$ with a vocabulary of size $|\text{Vocab}| = m$, let $l_{\max}$ be the length of the longest text. We do preprocessing of the training corpus in $\mathcal{O}(nl_{\max} + nm)$ time. Constructing the binary term-document matrix takes $\mathcal{O}(nl_{\max})$ time by reading through each text in $\mathcal{O}(nl_{\max})$ time and updating entries in the matrix corresponding to occurring words in $\mathcal{O}(1)$ time. The subsequent calculation of the Phi-coefficient for each word individually takes $\mathcal{O}(n)$ time and is done in cumulative $\mathcal{O}(nm)$ time. The preprocessing results in a associative datastructure of size $\mathcal{O}(m)$, that maps each word to its word-correlation. Given constant lookup in this datastructure (e.g. hash table), we only need $\mathcal{O}(l)$ time to construct a correlation signal for a query text $T$ of size $|T| = l$.

### 2.3. Classifier

Given the mapping $\varphi$ from a text to its correlation signal, we define a classifier

$$f(T) = \begin{cases} 1 & \text{if } \frac{1}{|T|}\sum_{x \in \varphi(T)} x > \tau \\ 0 & \text{otherwise} \end{cases}$$

for a given parameter $\tau$. Intuitively, the average correlation signal acts as a soft decision boundary: if a text contains more words that tend to appear in AI-generated texts, its average correlation will be positive, and vice versa. The threshold $\tau$ determines the decision boundary in this latent correlation space. In practice the optimal decision threshold $\tau$ is chosen to minimize classification error for the given distribution of the training data (see Figure 1).

## 3. Interpretability

This section gives a perspective on the interpretability of the proposed approach. Correlation signals are based on the word-correlations assigned to each individual word. This word level contribution offers ways to analyze the underlying model on a global and local (instance-level) scale to explain the final predictions.

### 3.1. Correlation Signals

Globally, we can look at the magnitude of the correlations and see that AI models appear to avoid specific words more (strong negative correlation, $\min_{w \in \text{Vocab}(\mathcal{T})} \varphi(w) = -0.4849$) than they seem to

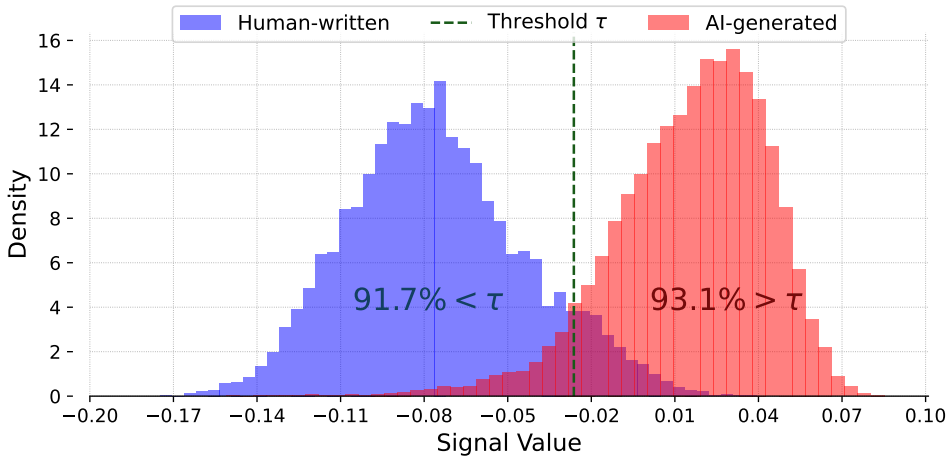


**Figure 1:** Distribution of the means of correlation signals for the two classes on instances of the training set of task 1 (cf. Section 4). The threshold $\tau = -0.0317$ is optimal with respect to the training data.

favor specific words (positive correlation, $\max_{w \in \text{Vocab}(\mathcal{T})} \varphi(w) = 0.3338$). A list of tokens with the largest/smallest correlation scores is given in Table 7 in Appendix B. Furthermore, interpreting text as a correlation signal opens the door to more advanced analyses, such as spectral methods to investigate global patterns and structural trends (see Appendix C).

On the local scale, these scores can be used to explain individual instances, as the final output sum can be traced back to the specific token-level contributions at each point in the sequence. Predictions are constructed sequentially from the individual word-correlations in a text. This allows to pinpoint exactly the word or sub-sentence structure that lead to either predicted class. Given an appropriate threshold $\tau$, we can see in Figure 2 how the models prediction changes from one class to the other as a result of words with an opposing word-correlation occurring.

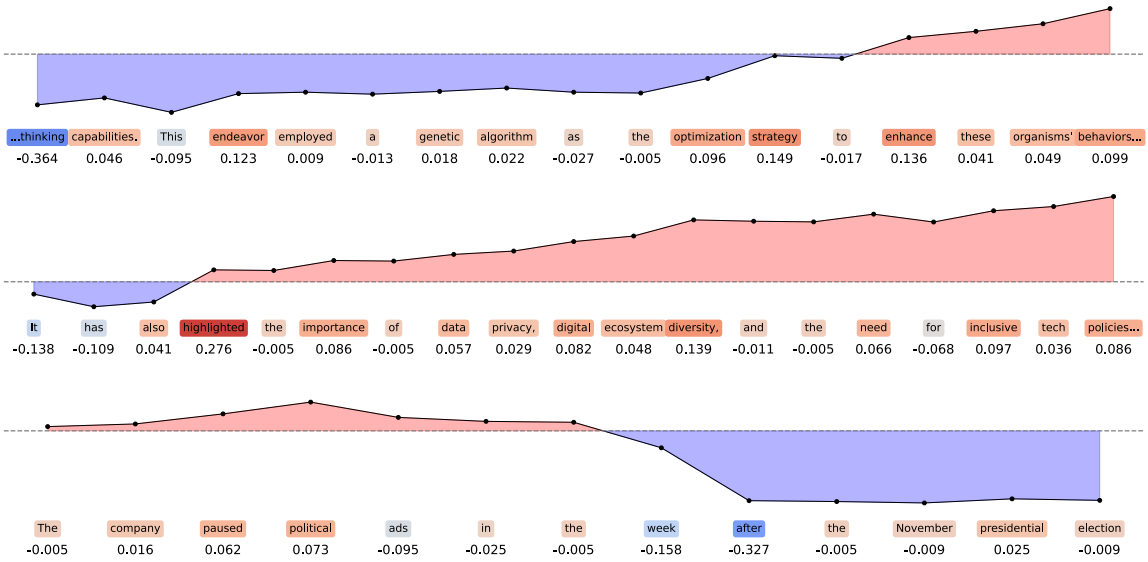


**Figure 2:** Example of how the cumulative sum (top) of individual words/signals (bottom) determine the final prediction. Positive values indicate higher correlation to machine-generated texts, while negative values are indicative of human-written texts. Excerpts were taken from the validation set and correctly classified.

### 3.2. $n$-gram extension

We generalize our approach to $n$-grams by treating them the same as simple word tokens. We calculate an $n$-gram-correlation score analogous to word-correlations and build the final correlation signal as a sequence of such $n$-gram-correlations.

Intuitively, we can capture more nuanced language interactions from the text by using $n$-grams as they capture local contextual dependencies. However, $n$-grams for $n > 1$ are sparse. There is a total of $56987$ tokens contained in the text corpus of the training data. Only $0.3\%$ of the tokens in the validation set are not present during training. However, about $34\%$ of the 2-grams and $84\%$ of the 3-grams in the validation set have not been seen during training. This leads to poor generalization to unseen data, while the ability to find $n$-gram-correlations that fit the training dataset improves with larger $n$. (This effect explains the reduced performance of the `corsig-2gram` and `corsig-3gram` runs in Table 3.)

## 4. Experimental Evaluation

We evaluate the performance of correlation signal classifiers. There are two main objectives in our experiments: (1) Determine the ability of our approach to generalize to new instances and (2) identify if correlation signals contain additional predictive information, not contained in simple linguistic measures. We will evaluate our approach on the dataset provided in the PAN Lab's Voight-Kampff Generative AI Detection challenge [10]. This challenge is split into two tasks. Task 1 consists of training and validation data for the binary classification setting presented in Section 2.1. Task 2 is a variation with 6 classes for different human-AI collaboration schemes (cf. Table 1).

The experiments are implemented in Python and the code is available on GitHub[1].

### 4.1. Exploratory Data Analysis

For both tasks of the Voight-Kampff Generative AI Detection challenge, separate datasets are provided. As shown in Table 1, the class distributions in the training and validation sets are relatively balanced for task 1. In contrast, task 2 shows significant imbalances, both across individual classes and between the training and validation splits. Specifically, in the training set, classes 3–5 together account for less than $10\%$ of the data. This is even more prominent in the validation set, where classes 4–5 collectively represent only $1.01\%$ of samples. The most significant inconsistency appears in class 3: while it comprises just $3.72\%$ of the training data, it dominates the validation set with $51.16\%$. Such inconsistencies between training and validation distributions can severely impair controlled evaluation of model performance, as they lead to incorrect representations of the target data distribution during training.

### 4.2. Baselines

We introduce a simple baseline classifier that takes several hand-crafted features into account. For task 1, simple classification based on the respective optimal threshold $\tau$ of said features already achieves a high performance that translates well from training to validation data (see Table 2). The features are calculated separately for the train and validation set and then fed into any standard machine learning algorithm (Random Forest, RF, in our case) to serve as a baseline. Additionally, we employ Facebook's RoBERTa base model [14] (`roberta-base`[2] via Hugging Face) as a Language Model (LM) baseline classifier, which has proven beneficial in previous studies [1]. We fine-tune RoBERTa using a maximum input sequence length of 500 tokens, running for three epochs on a T4 GPU provided by Google Colab.

[1] https://github.com/max-seeli/steely
[2] https://huggingface.co/FacebookAI/roberta-base

**Table 1**
Class Distributions in Train and Validation Datasets for Task 1 and Task 2.

| Task | Dataset | Label | Count | Ratio |
|---|---|---|---|---|
| Task 1 | Train | Human (0) | 9,101 | 38.39% |
| | | AI (1) | 14,606 | 61.61% |
| | Validation | Human (0) | 1,277 | 35.58% |
| | | AI (1) | 2,312 | 64.42% |
| Task 2 | Train | Fully human-written (0) | 75,270 | 26.05% |
| | | Human-written, machine-polished (1) | 95,398 | 33.02% |
| | | Machine-written, humanized (2) | 91,232 | 31.58% |
| | | Human-initiated, machine-continued (3) | 10,740 | 3.72% |
| | | Deeply-mixed (human + machine parts) (4) | 14,910 | 5.16% |
| | | Machine-written, human-edited (5) | 1,368 | 0.47% |
| | Validation | Fully human-written (0) | 12,330 | 16.97% |
| | | Human-written, machine-polished (1) | 12,289 | 16.91% |
| | | Machine-written, humanized (2) | 10,137 | 13.95% |
| | | Human-initiated, machine-continued (3) | 37,170 | 51.16% |
| | | Deeply-mixed (human + machine parts) (4) | 225 | 0.31% |
| | | Machine-written, human-edited (5) | 510 | 0.70% |

The selected hyperparameters are based on default values and were chosen to establish a reasonable initial baseline for comparison.

**Table 2**
Extracted baseline features and their classification performance on task 1 when thresholding via their respective $\tau$.

| Feature | $\text{ACC}_{\text{train}}$ | $\text{ACC}_{\text{val}}$ |
|---|---|---|
| `document length` | 65.08% | 65.51% |
| `average sentence length`[a] | 59.70% | 61.27% |
| `average word length`[b] | 78.60% | 79.66% |
| `type-token ratio (TTR)` | 64.87% | 61.30% |
| `stopword ratio` | 77.09% | 76.76% |
| `punctuation density` | 62.93% | 66.15% |
| `inverse document frequency (IDF)` | 71.40% | 73.67% |

[a] We refer to a sentence as a dot-delimited sequence of characters.
[b] We refer to a word as a space-delimited sequence of characters.

## 4.3. Data Preprocessing

To prepare the input data for processing into correlation signals, we first use a word-tokenizer that is sensitive to punctuation for the English language. Subsequently, we employ the Porter stemming algorithm [15] and remove English stopwords.

For the RoBERTa baseline, we use the model specific tokenizer and do not further preprocess the inputs.

## 4.4. Task 1: Binary Classification

For task 1, we analyze six systems and present the evaluation metrics in Table 3. We run the statistical baseline with the name `stats` and the RoBERTa baseline as `roberta`. The systems `corsig-<n>gram` for $n \in \{1, 2, 3\}$ uses our main approach as presented in Section 2 as well as the extension to $n$-grams from Section 3.2. Finally, the system `stats-corsig` is an adaptation to the statistical baseline, that

uses the correlation signal $\frac{1}{|T|}\sum_{x\in T}\varphi(x)$ for each text $T \in \mathcal{T}$ as an additional feature.

We can clearly see the negative effect $n$-grams with $n > 1$ have on the discriminative power of correlation signals, as we witness a slight decline in the performance metrics from `corsig-1gram` to `corsig-2gram` and a significantly more pronounced drop in performance when looking at `corsig-3gram`. The reason for this behaviour is the sparsity of $n$-grams as explained in Section 3.2.

Furthermore, system `stats-corsig` displays a substantial increase over the `stats` baseline. This indicates that correlation signals contain statistical information, not available from simple linguistic features. `stats-corsig` also shows that combined with correlation signals, a simple statistical baseline is sufficient for competetive performance levels to the `roberta` baseline.

**Table 3**
Validation-Set performance for Task 1 (higher is better).

| Run | Roc-Auc | Brier | C@1 | F1 | F05U | Mean |
|---|---|---|---|---|---|---|
| `corsig-1gram (ours)` | 0.902 | 0.916 | 0.916 | 0.935 | 0.927 | 0.919 |
| `corsig-2gram (ours)` | 0.895 | 0.920 | 0.920 | 0.940 | 0.917 | 0.918 |
| `corsig-3gram (ours)` | 0.510 | 0.651 | 0.651 | 0.787 | 0.698 | 0.659 |
| `stats-corsig (ours)` | 0.992 | 0.969 | 0.957 | 0.967 | 0.962 | 0.969 |
| `stats (baseline)` | 0.945 | 0.918 | 0.894 | 0.920 | 0.905 | 0.916 |
| `roberta (baseline)` | 0.996 | 0.984 | 0.984 | 0.988 | 0.983 | 0.987 |

### 4.5. Task 2: Multi-Class Classification

For task 2, it is important to note that we are no longer dealing with binary classification, but rather a multi-class setting with six distinct classes. Consequently, our approach for creating correlation signals via a binary label vector $\mathbf{y}$ and classifying the summed up signals via $\tau$, as introduced in Sections 2.2 and 2.3, no longer works. We define $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \{0, 1, 2, 3, 4, 5\}^n$ and build the correlation signals according to $\varphi(w_i) = \varphi(\mathbf{B}_{i,\cdot}, \mathbf{y})$. Instead of using a threshold $\tau$ for classification, we use the RF classifier as described in Section 4.2, both with and without normalization $\frac{1}{|T|}\sum_{x\in T}\varphi(x)$ for each $T \in \mathcal{T}$.

The results of our experiments on the validation-set can be seen in Table 4. The RoBERTa baseline (`roberta`) clearly outperformed the RF classifiers, both with (`stats-corsig`) and without (`stats`) the correlation signal, which just slightly outperform guessing levels.

We hypothesize that the lack of performance can be attributed to an inconsistent class distribution between the training and validation sets, as described in Section 4.1. To verify this, we combined the original training and validation data and performed a new stratified split. The results on the new validation set confirm our assumptions, as we receive an F1-score of $96\%$ via the RoBERTa baseline (`roberta-strat`). Additionally, we can now we observe a clear performance gain when using the correlation signal as a feature in the RF classifier (`stats-corsig-strat`) compared to using baseline features alone (`stats-strat`). Nonetheless, the RF classifier still underperforms relative to the LM baseline, suggesting that our feature-based approach may be less effective for multi-class classification tasks.

## 5. Conclusion

In this work, we presented a lightweight and interpretable approach for distinguishing human-written from AI-generated text. Our method leverages the statistical correlation between individual words and class labels, encoding texts as correlation signals that can be processed efficiently and explained

**Table 4**
Validation-Set (top) and stratified Validation-Set (bottom) performance for Task 2 (higher is better).

| Run | Accuracy | Macro F1 | Macro Recall | Mean |
|---|---|---|---|---|
| `roberta (baseline)` | 0.57 | 0.61 | 0.67 | 0.616 |
| `stats (baseline)` | 0.29 | 0.21 | 0.32 | 0.273 |
| `stats-corsig (ours)` | 0.31 | 0.22 | 0.33 | 0.287 |
| `roberta-strat (baseline)` | 0.97 | 0.96 | 0.96 | 0.963 |
| `stats-strat (baseline)` | 0.55 | 0.45 | 0.43 | 0.477 |
| `stats-corsig-strat (ours)` | 0.68 | 0.66 | 0.63 | 0.657 |

both globally and locally. We demonstrated that this signal-based representation achieves strong performance in the binary classification setting and adds complementary value when combined with standard statistical features.

In the multi-class classification setting, we observed that correlation signals alone may not capture the full complexity of mixed-authorship scenarios. However, they still offer predictive gains when incorporated into classical models, provided that the data distribution is properly balanced. While language models like RoBERTa remain state-of-the-art in terms of raw accuracy, our findings show that interpretable, transparent, and computationally efficient methods can provide competitive alternatives—particularly when interpretability is a key concern.

In future work, we plan to introduce a relevance weight (e.g. tf-idf) for each word to calculate a weighted correlation signal, ensuring that more significant words impact the overall signal more. When removing stopwords, we already saw a performance improvement, which indicates that less relevant terms primarily add noise, hindering the prediction. Future work also includes extending correlation-based features to more fine-grained signals over richer linguistic representations (e.g., syntactic or semantic structures), and exploring hybrid models that combine the interpretability of correlation signals with the expressiveness of neural networks.

## 6. Declaration on Generative AI

During the preparation of this work, we used ChatGPT to paraphrase and reword. After using this service, we reviewed and edited the content as needed and take full responsibility for the publication's content.

## References


[1] A. M. Sarvazyan, J. Ángel González, M. Franco-Salvador, F. Rangel, B. Chulvi, P. Rosso, Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains, 2023. URL: https://arxiv.org/abs/2309.11285. arXiv:2309.11285.

[2] G. Cabanac, C. Labbé, Prevalence of nonsensical algorithmically generated papers in the scientific literature, Journal of the Association for Information Science and Technology 72 (2021) 1461–1476. doi:10.1002/asi.24495.

[3] J. D. Rodriguez, T. Hay, D. Gros, Z. Shamsi, R. Srinivasan, Cross-domain detection of GPT-2-generated technical text, in: M. Carpuat, M.-C. de Marneffe, I. V. Meza Ruiz (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 1213–1233. URL: https://aclanthology.org/2022.naacl-main.88/. doi:10.18653/v1/2022.naacl-main.88.

[4] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, H. Hajishirzi, Fact or fiction: Verifying scientific claims, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 7534–7550. URL: https://aclanthology.org/2020.emnlp-main.609/. doi:10.18653/v1/2020.emnlp-main.609.

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

[6] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, C. Finn, Detectgpt: Zero-shot machine-generated text detection using probability curvature, 2023. URL: https://arxiv.org/abs/2301.11305. arXiv:2301.11305.

[7] G. Bao, Y. Zhao, Z. Teng, L. Yang, Y. Zhang, Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature, 2024. URL: https://arxiv.org/abs/2310.05130. arXiv:2310.05130.

[8] Y. Xu, Y. Wang, H. An, Z. Liu, Y. Li, Detecting subtle differences between human and model languages using spectrum of relative likelihood, 2024. URL: https://arxiv.org/abs/2406.19874. arXiv:2406.19874.

[9] Z. Yang, Y. Yuan, Y. Xu, S. Zhan, H. Bai, K. Chen, Face: Evaluating natural language generation with fourier analysis of cross-entropy, 2023. URL: https://arxiv.org/abs/2305.10307. arXiv:2305.10307.

[10] 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.

[11] 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.

[12] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: 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.

[13] B. W. Matthews, Comparison of the predicted and observed secondary structure of t4 phage lysozyme, Biochimica et Biophysica Acta (BBA)-Protein Structure 405 (1975) 442–451.

[14] 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.

[15] M. F. Porter, An algorithm for suffix stripping, Program 14 (1980) 130–137.

[16] D. Dickey, W. Fuller, Distribution of the estimators for autoregressive time series with a unit root, JASA. Journal of the American Statistical Association 74 (1979). doi:10.2307/2286348.

# A. Further Results

The PAN Lab's challenge organizers evaluated the submitted models from Task 1 on additional datasets. The test-set is a previously unknown part of the original dataset for competition purposes and the Eloquent dataset comes from a related competition, where participants are asked to generate text, such that it is indistinguishable from human text. We present the results in Tables 5 and 6.

**Table 5**
Test-Set performance for Task 1 (higher is better).

| Run | Roc-Auc | Brier | C@1 | F1 | F05U | Mean |
|---|---|---|---|---|---|---|
| `corsig-1gram (ours)` | 0.823 | 0.823 | 0.823 | 0.867 | 0.895 | 0.846 |
| `corsig-2gram (ours)` | 0.826 | 0.837 | 0.837 | 0.880 | 0.896 | 0.855 |
| `corsig-3gram (ours)` | 0.518 | 0.709 | 0.709 | 0.827 | 0.749 | 0.702 |
| `stats-corsig (ours)` | 0.972 | 0.924 | 0.886 | 0.914 | 0.944 | 0.928 |
| `stats (baseline)` | 0.921 | 0.898 | 0.851 | 0.892 | 0.895 | 0.891 |
| `roberta (baseline)` | 0.966 | 0.927 | 0.925 | 0.945 | 0.965 | 0.945 |

**Table 6**
Eloquent dataset performance for Task 1 (higher is better).

| Run | Roc-Auc | Brier | C@1 | F1 | F05U | Mean |
|---|---|---|---|---|---|---|
| `corsig-1gram (ours)` | 0.613 | 0.632 | 0.632 | 0.761 | 0.863 | 0.700 |
| `corsig-2gram (ours)` | 0.698 | 0.674 | 0.674 | 0.791 | 0.888 | 0.745 |
| `corsig-3gram (ours)` | 0.500 | 0.923 | 0.923 | 0.960 | 0.937 | 0.849 |
| `stats-corsig (ours)` | 0.918 | 0.917 | 0.916 | 0.953 | 0.967 | 0.934 |
| `stats (baseline)` | 0.835 | 0.930 | 0.933 | 0.964 | 0.965 | 0.925 |
| `roberta (baseline)` | 0.724 | 0.579 | 0.575 | 0.703 | 0.852 | 0.687 |

## B. Significant Word-Correlations

**Table 7**
Tokens most correlated with machine-generated texts (positive scores, left) and human-written texts (negative scores, right). Tokens are stemmed, leading to truncated word forms.

| Machine-generated (Positive) | | | | Human-written (Negative) | | | |
|---|---|---|---|---|---|---|---|
| **Token** | **Score** | **Token** | **Score** | **Token** | **Score** | **Token** | **Score** |
| echo | 0.334 | landscap | 0.215 | go | -0.485 | three | -0.297 |
| challeng | 0.318 | reson | 0.212 | went | -0.425 | talk | -0.294 |
| shadow | 0.293 | impact | 0.211 | littl | -0.417 | ye | -0.293 |
| despit | 0.286 | reveal | 0.211 | look | -0.416 | till | -0.293 |
| within | 0.283 | approach | 0.207 | say | -0.400 | half | -0.292 |
| highlight | 0.276 | resolv | 0.207 | said | -0.399 | come | -0.291 |
| complex | 0.274 | path | 0.206 | get | -0.382 | pretti | -0.291 |
| cast | 0.273 | gentl | 0.204 | put | -0.377 | made | -0.289 |
| reflect | 0.271 | beneath | 0.204 | came | -0.371 | make | -0.284 |
| whisper | 0.269 | solac | 0.204 | never | -0.369 | round | -0.283 |
| amidst | 0.265 | profound | 0.203 | think | -0.364 | gave | -0.283 |
| weight | 0.265 | danc | 0.201 | thing | -0.362 | anyth | -0.282 |
| signific | 0.263 | air | 0.201 | good | -0.357 | realli | -0.282 |
| remind | 0.262 | unspoken | 0.200 | two | -0.350 | first | -0.278 |
| underscor | 0.260 | measur | 0.199 | much | -0.348 | quit | -0.277 |
| potenti | 0.253 | resili | 0.199 | know | -0.348 | tell | -0.274 |
| testament | 0.247 | transform | 0.198 | got | -0.344 | answer | -0.267 |
| emphas | 0.242 | warmth | 0.197 | well | -0.344 | done | -0.265 |
| flicker | 0.242 | concern | 0.197 | great | -0.344 | enough | -0.257 |
| navig | 0.236 | intric | 0.196 | give | -0.340 | peopl | -0.256 |
| gaze | 0.230 | crucial | 0.196 | would | -0.333 | oh | -0.254 |
| role | 0.229 | tension | 0.195 | want | -0.332 | whole | -0.253 |
| linger | 0.228 | spark | 0.193 | ask | -0.318 | money | -0.249 |
| tapestri | 0.227 | serv | 0.193 | noth | -0.316 | man | -0.247 |
| share | 0.227 | remain | 0.192 | old | -0.315 | take | -0.246 |
| ensur | 0.225 | narr | 0.191 | one | -0.314 | last | -0.246 |
| stark | 0.223 | emerg | 0.187 | told | -0.307 | hous | -0.243 |
| embrac | 0.221 | spirit | 0.187 | poor | -0.303 | year | -0.240 |
| unfold | 0.219 | sens | 0.186 | see | -0.301 | better | -0.236 |
| shift | 0.216 | scent | 0.186 | girl | -0.298 | heard | -0.235 |

## C. Spectral Analysis of Correlation-Signals

Since we are looking at texts in the form of signals (see Section 2.2), we hypothesize that there are certain structural differences between human-written and AI-generated texts that can be uncovered by analyzing their frequency components. Specifically, let $\varphi(T)_j$ denote the real-valued correlation signal of the word at position $j$ of text $T$. We interpret $\varphi(T)$ as a discrete-time process, which encodes some sort of evidence towards AI- or human-authorship. Our goal is to examine the power spectral density (PSD) for a text $T$ via the periodogram $P_T$, which serves as a basic estimator for the PSD. $P_T$ is defined as

$$P_T(f) = \left| \sum_{n=0}^{l-1} \varphi(T)_j \cdot e^{-i2\pi \frac{f}{l} j} \right|^2$$

where $f \in \{0, 1, \ldots, l-1\}$ is the discrete frequency index and $l$ is the length of document $T$.

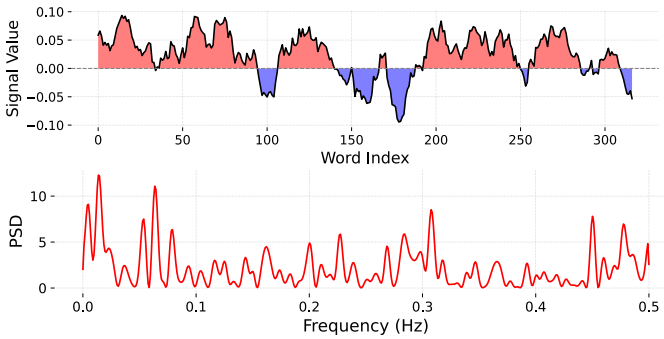


**Figure 3:** Example of how an AI-generated text $T$ can be seen as a signal $\varphi(T)$ (top) and how this signal can be used for spectral analysis $P_T$ (bottom).

To conduct spectral analysis, will will use Welch's method, which segments the signal into overlapping windows, applies a tapering function and finally averages the resulting periodograms. This method, however, assumes *stationarity* of the signal, which means that the mean and variance do not change over time; this is non-trivial for natural language. Similarly to [9], we applied the Augmented Dickey-Fuller (ADF) test [16] to examine this property. Our null hypothesis $H_0$ of the ADF test is *non-stationarity*, meaning that $p < .05$ test results would reject $H_0$ and hence accept the alternative hypothesis of *stationarity* in the signals. For the training set of task 1, we see that $99.92\%$ of texts accept $H_1$, which is also why we assume that Welch's method can be applied to this kind of correlation signal. An example of a resulting PSD for an AI-generated text can be seen in Figure 3.

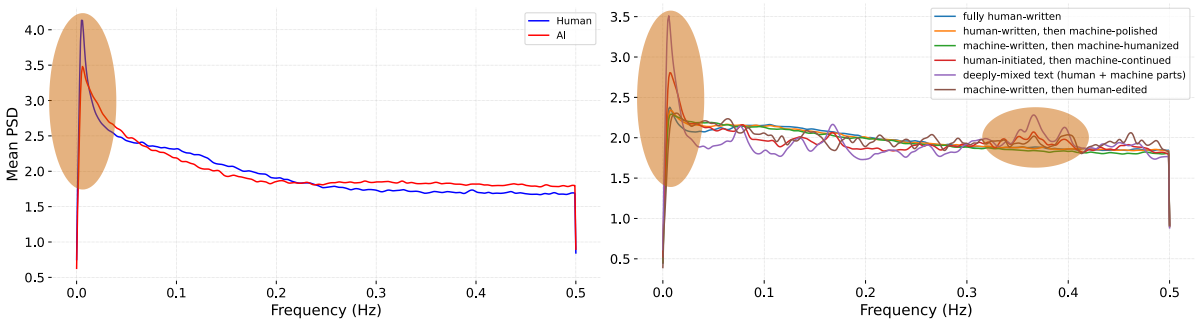


**Figure 4:** The mean periodograms (power density spectrum) of the individual classes show distinct differences (highlighted in orange) in both task 1 (left) and task 2 (right).

After calculating $P_T$ for all $T \in \mathcal{T}$, we average the values of these periodograms within each individual class; Figure 4 shows that there are indeed distinct differences in the mean power density spectra of the correlation scores.

For task 1, we can see that both classes have a peak in the low-frequency range, which means that occurring patterns change slowly across the texts. In our context, this would indicate that the correlation scores remain mostly positive or negative over many words. This aligns with our expectation that human- and machine-authored segments typically span full sentences or paragraphs rather than just single words.

We see a similar trend in task 2. There are two large low-frequency peaks for *human-initiated and machine-continued* text as well as *deeply-mixed* texts, suggesting that machine- and human-authored parts are interleaved on the sentence- or paragraph-level. As expected, such a peak does not exist for *fully human-written* texts. Interestingly, we can see a minor peak at higher frequencies for the *machine-written, then human-edited* category. This could indicate that human editors made small local changes, such as modifying individual words or short phrases, rather than rewriting entire segments. Such finer-grained edits introduce higher-frequency peaks in the correlation signal.