# AI Text Detection Method Based on Perplexity Features with Strided Sliding Window

Notebook for the PAN Lab at CLEF 2024

Xurong Liu, Leilei Kong*

*Foshan University, Foshan , Guangdong, China*

**Abstract**

In recent years, the application of Large Language Models (LLMs) in various Natural Language Processing (NLP) tasks has become prevalent, significantly enhancing text generation, machine translation, language understanding, and conversational systems. However, this widespread use has introduced new ethical and legal challenges, particularly the difficulty in distinguishing human-generated content from AI-generated content. This paper addresses this issue by treating it as an authorship verification problem, aiming to identify whether a given text is AI-generated. We investigate the distinct characteristics of human and AI-generated texts and employ a strided sliding window approach based on GPT-2 to extract perplexity features. For the task of Voight Kampff Generative AI Author Verification 2024, we determined AI text and human text by comparing perplexity features. The results demonstrated that by leveraging the perplexity metric, which measures the unpredictability of a text, we were able to capture distinct patterns characteristic of AI-generated content.

**Keywords**

AI Detection, Perplexity, GPT-2

## 1. Introduction

In recent years, Large Language Models (LLMs) have been widely applied to various Natural Language Processing (NLP) tasks and applications, including text generation, machine translation systems, and so on. These models have significantly assisted in many aspects of daily life. However, their potential uses have also introduced a series of new ethical and legal issues. As algorithms continuously iterate and innovate, it becomes increasingly difficult to distinguish between human-generated content and AI-generated content. Therefore, strengthening the identification and regulation of AI-generated content has become particularly important. Currently, the academic consensus is to treat the detection of AI-generated text as an authorship attribution problem, which aims to identify whether a given text is generated by AI. [1]

In this paper, we briefly study the characteristics of human text and AI-generated text and chose to use a strided sliding window based on GPT-2 (Generative Pre-trained Transformer 2)to extract the perplexity features of the text [2]. For the task of Voight Kampff Generative AI Author Verification 2024, we determined AI text and human text by comparing perplexity features. Additionally, we analyzed its applicability in text classification and explored its effectiveness in this task.

## 2. Background

With LLMs improving at breakneck speed and seeing more widespread adoption every day, it is getting increasingly hard to discern whether a given text was authored by a human being or AI[3]. As developers of ChatGPT, OpenAI approaches the detection of AI-generated text as a binary classification problem. They conduct research on fine-tuning models based on RoBERTa and GPT-2 detector models to distinguish between non-AI-generated text and text generated by GPT-2. However, as the size of

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

*Corresponding author.

Liu_xurongx@163.com (X. Liu); kongleilei@fosu.edu.cn (L. Kong)

0009-0000-4386-5336 (X. Liu); 0000-0002-4636-3507 (L. Kong)

the text generation model increases, the performance of the classifier tends to decline[4]. By studying existing generative AI models, GPTZero analyzes two metrics of text: "perplexity" and "burstiness". GPTZero is capable of detecting text generated by various AI models, including Google's LaMD (also known as Bard), Facebook's LLaMa, and OpenAI's GPT-3 and GPT-4[5]. Biyang Guo collected a dataset named the Human ChatGPT Comparison Corpus (HC3) and studied the differences between human and AI-generated texts in both Chinese and English based on this dataset. By analyzing the perplexity feature at both the sentence and text levels, it was found that ChatGPT has relatively lower PPLs compared to the text written by humans[6]. In the work of Lorenz Mindner[7], traditional and novel features were explored to distinguish AI-generated text from human text and AI-rewritten text. When using GPT-2 to calculate perplexity and analyze based on this feature, it was found that the perplexity of approximately 25% of AI-generated texts was significantly lower compared to nearly 50% of human texts. Additionally, they used XGBoost for text classification and achieved good results. And some researchers used perplexity as a feature on GPT-2 to distinguish between human-generated and AI-generated text based [8, 9, 10]. Many studies assert that linguistic analysis indicates humans exhibit greater logicality, semantic coherence, and contextual understanding in language use. When expressing ideas, humans tend to minimize information quantity while maintaining semantic clarity and effective communication, resulting in lower entropy. In contrast, AI-generated texts often have more complex syntactic structures but lower lexical complexity. In most cases, the perplexity of AI text is lower than that of human text[8, 9, 11, 12].

## 3. System Overview

The Generative AI Authorship Verification Task @ PAN is organized in collaboration with the Voight-Kampff Task @ ELOQUENT Lab in a builder-breaker style: Given two texts, one authored by a human and one by a machine, pick out the human. Test data for this task will be compiled from the submissions of ELOQUENT participants and will comprise multiple text genres such as news articles Wikipedia intro texts or fanfiction. Additionally, a bootstrap dataset is provided[10, 3].

Due to the imbalance in the quantity of human-generated texts versus AI-generated texts of 2024 PAN, we investigated the following features: average length ($L$) which is the average number of words per text; vocabulary size ($V$) which is the number of unique words used across all responses; and density ($D$) calculated as

$$D = \frac{100 \times V}{L \times N} \tag{1}$$

where $N$ is the number of texts. Density measures the concentration of unique words used in the text. A higher density indicates a greater variety of different words used within texts of the same length.[6]

The text features are shown in the table 1 . The features $L$ and $V$ show that human-generated texts are relatively longer and use a more extensive vocabulary. However, for more advanced large models, these characteristics are less pronounced. Similarly, this phenomenon is prominently reflected in the $D$ feature. To obtain accurate results, it is necessary to use the features of entire sentences for classification.

Perplexity (PPL) is one of the most common metrics for evaluating language models. Perplexity is defined as the exponentiated average negative log-likelihood of a sequence[13]. If we have a tokenized sequence $X = (x_0, x_1, \ldots, x_t)$ then the perplexity of $X$ is

$$PPL(X) = \exp\left(-\frac{1}{t}\sum_{i}^{t} \log p_\theta(x_i|x_{<i})\right) \tag{2}$$

Where $\log p_\theta(x_i|x_{<i})$ is the log-likelihood of the $i$-th token conditioned on the preceding tokens $x_{<i}$ according to the model. This is also equivalent to the exponentiation of the cross-entropy between the data and model predictions.

We chose GPT-2 as the base model for calculating perplexity. GPT-2 is a large language model developed by OpenAI based on the Transformer architecture. It is pre-trained in an unsupervised

**Table 1**
Data information of Authorship verification datasets

| Text | Average Length (L) | Vocab Size (V) | Density (D) |
|---|---|---|---|
| human | 494.10 | 234617 | 43.68 |
| alpaca-7b | 141.05 | 63380 | 41.34 |
| bigscience-bloomz-7b1 | 324.51 | 71080 | 20.15 |
| chavinlo-alpaca-13b | 163.50 | 60976 | 34.31 |
| gemini-pro | 451.33 | 201654 | 41.10 |
| gpt-3.5-turbo-0125 | 402.78 | 204240 | 46.65 |
| gpt-4-turbo-preview | 517.58 | 264543 | 47.02 |
| meta-llama-llama-2-70b-chat-hf | 495.60 | 190104 | 35.29 |
| meta-llama-llama-2-7b-chat-hf | 429.88 | 163386 | 34.97 |
| mistralai-mistral-7b-instruct-v0.2 | 530.64 | 221285 | 38.36 |
| mistralai-mixtral-8x7b-instruct-v0.1 | 528.15 | 237193 | 41.32 |
| qwen-qwen1.5-72b-chat-8bit | 420.10 | 224780 | 49.22 |
| text-bison-002 | 512.92 | 235416 | 42.22 |
| vicgalle-gpt2-open-instruct-v1 | 418.59 | 92515 | 20.33 |

manner on a large text dataset containing billions of words, enabling it to generate text that closely resembles human language. It can handle contexts up to 1024 tokens, allowing it to consider more context information and thus predict the next word more accurately[2]. In summary, GPT -2 can provide more accurate assessment of perplexity.

The text is always limited by a model's context size when evaluating the model's perplexity by autoregressively factorizing a sequence and conditioning on the entire preceding subsequence at each step. The largest version of GPT-2 has a fixed length of 1024 tokens, so we cannot calculate $logp_\theta(x_i|x_{<i})$ directly when $t$ is greater than 1024. we then approximate the likelihood of a token $x_t$ by conditioning only on the fixed tokens that precede it rather than the entire context. So, when evaluating the model's perplexity of a sequence, we break the sequence into disjoint chunks and independently add up the decomposed log-likelihoods of each segment. To solve the model that will have less context at most of the prediction steps, we evaluate with a sliding-window strategy so that the model has more context when making each prediction. This is a closer approximation to the true decomposition of the sequence probability and will typically yield a more favorable score. The downside is that it requires a separate forward pass for each token in the corpus. So, we employ a strided sliding window, moving the context by 512 token strides rather than sliding by 1 token a time. This allows computation to proceed much faster while still giving the model a large context to make predictions at each step. For the detailed algorithm, refer to Algorithm 1.

**Algorithm 1** Strided Sliding Window for Evaluate PPL

```
Input:  model max_length, stride = 512
Output:  ppl
seq_len = encodings.input_ids.size {Number of tokens for input text}
prev_end_loc = 0 {Index of the previous ending position}
nlls = [] {List to store negative log likelihoods}
while prev_end_loc < seq_len do
    begin_loc = prev_end_loc
    end_loc = min(prev_end_loc + stride, seq_len)
    input_ids = encodings.input_ids[:, begin_loc:end_loc] {Input sequence}
    target_ids = input_ids.clone()
    target_ids[:, :-stride] = -100 {Fill the beginning of the target sequence with -100}
    with torch.no_grad():
    outputs = model(input_ids, labels=target_ids)
    neg_log_likelihood = outputs.loss
    nlls.append(neg_log_likelihood) {Store the negative logarithmic likelihood of each block}
    prev_end_loc = end_loc {Update index}
    if end_loc == seq_len then
        break
    end if
end while
ppl = torch.exp(torch.stack(nlls).mean())
```

For the task of Voight Kampff Generative AI Author Verification 2024, which we addressed by treating it as an authorship attribution problem, after fully extracting the perplexity features of the text, we determined AI text and human text by comparing the magnitude of perplexity features, the text with lower perplexity is AI-generated.

## 4. Results

Following the above experiment design, the results are as table 2 and table 3 follows[3, 14]:

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

| Approach | ROC-AUC | Brier | C@1 | $F_1$ | $F_{0.5u}$ | Mean |
|---|---|---|---|---|---|---|
| adjacent-rate | 0.746 | 0.799 | 0.744 | 0.728 | 0.733 | 0.75 |
| Baseline Binoculars | 0.972 | 0.957 | 0.966 | 0.964 | 0.965 | 0.965 |
| Baseline Fast-DetectGPT (Mistral) | 0.876 | 0.8 | 0.886 | 0.883 | 0.883 | 0.866 |
| Baseline PPMd | 0.795 | 0.798 | 0.754 | 0.753 | 0.749 | 0.77 |
| Baseline Unmasking | 0.697 | 0.774 | 0.691 | 0.658 | 0.666 | 0.697 |
| Baseline Fast-DetectGPT | 0.668 | 0.776 | 0.695 | 0.69 | 0.691 | 0.704 |
| 95-th quantile | 0.994 | 0.987 | 0.989 | 0.989 | 0.989 | 0.990 |
| 75-th quantile | 0.967 | 0.925 | 0.939 | 0.929 | 0.935 | 0.938 |
| Median | 0.907 | 0.890 | 0.878 | 0.866 | 0.865 | 0.887 |
| 25-th quantile | 0.705 | 0.769 | 0.673 | 0.654 | 0.662 | 0.692 |
| Min | 0.131 | 0.265 | 0.005 | 0.006 | 0.007 | 0.224 |

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

| Approach | Minimum | 25-th Quantile | Median | 75-th Quantile | Max |
|---|---|---|---|---|---|
| adjacent-rate | 0.086 | 0.429 | 0.656 | 0.750 | 0.944 |
| Baseline Binoculars | 0.342 | 0.818 | 0.844 | 0.965 | 0.996 |
| Baseline Fast-DetectGPT (Mistral) | 0.095 | 0.793 | 0.842 | 0.929 | 0.958 |
| Baseline PPMd | 0.270 | 0.546 | 0.750 | 0.770 | 0.863 |
| Baseline Unmasking | 0.250 | 0.653 | 0.673 | 0.697 | 0.762 |
| Baseline Fast-DetectGPT | 0.159 | 0.579 | 0.677 | 0.719 | 0.982 |
| 95-th quantile | 0.862 | 0.968 | 0.978 | 0.990 | 1.000 |
| 75-th quantile | 0.756 | 0.863 | 0.926 | 0.959 | 0.995 |
| Median | 0.593 | 0.622 | 0.874 | 0.884 | 0.943 |
| 25-th quantile | 0.343 | 0.470 | 0.637 | 0.683 | 0.703 |
| Min | 0.015 | 0.038 | 0.231 | 0.235 | 0.252 |

## 5. Conclusion

In this study, we explored the identification of AI-generated text using a combination of perplexity features extracted by a strided sliding window based on GPT-2. We determined AI text and human text by comparing the magnitude of perplexity features. The results demonstrated that by leveraging the perplexity metric, which measures the unpredictability of a text, we were able to capture distinct patterns characteristic of AI-generated content, but the performance is poor and further improvement is needed. In addition, our study is not without limitations. The variability in text characteristics across different AI models suggests that our method might need further adaptation to handle new and emerging models. Additionally, the computational intensity of the sliding window approach, despite its accuracy, could be a bottleneck in real-time applications. Future work should focus on optimizing the computational efficiency of our method and exploring its adaptability to newer, more advanced LLMs. Furthermore, integrating additional features and leveraging ensemble methods could enhance detection accuracy and robustness.

## Acknowledgments

This work is supported by the Natural Science Platforms and Projects of Guangdong Province Ordinary Universities (Key Field Special Projects) (No.2023ZDZX1023).

## References

[1] A. Uchendu, Z. Ma, T. Le, R. Zhang, D. Lee, Turingbench: A benchmark environment for turing test in the age of neural text generation, arXiv preprint arXiv:2109.13296 (2021).

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

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

[4] OpenAI, Ai classifier, 2023. URL: https://openai.com/, (2024).

[5] E. Tian, Gptzero: An ai text detector, 2023. URL: https://news.gptzero.me/thoughtful-thorough-solution-development-gptzero-x-anthology, (2024).

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

[7] L. Mindner, T. Schlippe, K. Schaaff, Classification of human-and ai-generated texts: Investigating features for chatgpt, in: International Conference on Artificial Intelligence in Education Technology, Springer, 2023, pp. 152–170.

[8] S. Gehrmann, H. Strobelt, A. M. Rush, Gltr: Statistical detection and visualization of generated text, arXiv preprint arXiv:1906.04043 (2019).

[9] S. Mitrović, D. Andreoletti, O. Ayoub, Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text, arXiv preprint arXiv:2301.13852 (2023).

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

[11] E. Crothers, N. Japkowicz, H. L. Viktor, Machine-generated text: A comprehensive survey of threat models and detection methods, IEEE Access (2023).

[12] Y. Liu, Z. Zhang, W. Zhang, S. Yue, X. Zhao, X. Cheng, Y. Zhang, H. Hu, Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models, arXiv preprint arXiv:2304.07666 (2023).

[13] huggingface, calculating-perplexity-with-gpt-2, 2023. URL: https://huggingface.co/docs/transformers/en/perplexity, (2024).

[14] J. Bevendorff, X. B. Casals, B. Chulvi, D. Dementieva, A. Elnagar, D. Freitag, M. Fröbe, D. Korenčić, M. Mayerl, A. Mukherjee, A. Panchenko, M. Potthast, F. Rangel, P. Rosso, A. Smirnova, E. Stamatatos, B. Stein, M. Taulé, D. Ustalov, M. Wiegmann, E. Zangerle, Overview of PAN 2024: Multi-Author Writing Style Analysis, Multilingual Text Detoxification, Oppositional Thinking Analysis, and Generative AI Authorship Verification, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. M. D. Nunzio, P. Galuščáková, A. G. S. de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2024.