LLM Project: A.I Generated Code Detection

Spotting LLMs With Binoculars: Zero-Shot Detection of Machine-Generated Text

Based on Hans et al., ICML 2024

Willy VO, Minh Duc NGUYEN, and Matheo

Université Paris-Dauphine – PSL

December 2025

1. Introduction

AI-generated text detection is a task that is becoming increasingly important due to the risk of misinformation, bots, fake reviews, and plagiarism.

The paper *Spotting LLMs with Binoculars* (Hans et al., ICML 2024) proposes a zero-shot solution for detecting AI-generated texts. Other existing detectors often rely on supervised training with a fixation on a specific model and fail to transfer to new models or text domains.

In contrast, the paper proposes **Binoculars**, a zero-shot, model-agnostic framework that uses no training data. It outperforms both open-source and commercial systems (e.g., GPTZero, Ghostbuster), achieving over a 90% true positive rate (TPR) at an exceptionally low 0.01% false positive rate (FPR).

2. Methodology

2.1. Overview

Binoculars "looks" at text through two "lenses": it uses two LLMs that are similar but not identical. **The central hypothesis is that when a text is AI-generated, two language models tend to agree in their predictions, whereas for human-written text, their predictions diverge.** Furthermore, by using two distinct language models instead of relying on a single one, Binoculars reduces dependence on an individual LLM (resulting in better generalization).

2.2. Formal Definitions

Let s be a text sequence tokenized as $\vec{x} = (x_1, x_2, \ldots, x_L)$, where L is the number of tokens. A language model M predicts the probability of each token given its previous context.
The **log-perplexity** of M on text s is defined as:

$$\text{logPPL}_M(s) = -\frac{1}{L}\sum_{i=1}^{L} \log M(s)_i = -\frac{1}{L}\sum_{i=1}^{L} \log P(x_i|x_{<i})$$

It measures how predictable the text is to the model - lower perplexity means the sequence is easier for the model to predict, while higher perplexity indicates it finds the text less predictable or more surprising.
The **cross-perplexity** extends this idea to a pair of models (M_1, M_2):

$$\text{logX-PPL}_{M_1,M_2}(s) = -\frac{1}{L}\sum_{i=1}^{L} M_1(s)_i \cdot \log M_2(s)_i = -\frac{1}{L}\sum_{i=1}^{L} P_{M_1}(x_i|x_{<i}) \log P_{M_2}(x_i|x_{<i})$$

It quantifies how **similar** the predictions of model M_2 are to those of model M_1 - lower values indicate strong alignment between the two models, while higher values reflect greater divergence in their predictions.
Finally, the **Binoculars score** is the ratio:

$$B_{M_1,M_2}(s) = \frac{\text{logPPL}_{M_1}(s)}{\text{logX-PPL}_{M_1,M_2}(s)}$$

A lower score indicates machine-like text (generally $B_{M_1,M_2}(s) \leq 1$), whereas a higher score indicates human-like text.

Note : The paper innovates with this method by introducing the cross perplexity term ,using two models to detect generated text instead of one.

Intuitive interpretation : Intuitively, human writing diverges more strongly from the statistical patterns shared by two LLMs than one model's output diverges from another's. Thus, for human texts, the numerator and denominator differ substantially, leading to a higher ratio. For LLM outputs, both models' probability distributions align closely, yielding a smaller ratio.

In the Binoculars framework, the authors also define a **cross-perplexity** between two models M_1 and M_2, which is the exponential of their mutual cross-entropy: $H(p, q) = -\sum_x p(x) \log q(x)$.

Challenges : Another challenge is that these models are highly sensitive to prompts : the wording, tone, or style of a query can lead to vastly different outputs, making detection even more difficult, as the same model can produce texts that vary widely in fluency, structure, and predictability.

3. Expriment Results

3.1. Datasets and Generation Process

We implemented a Binoculars benchmark using two core datasets: **news_cnn** (which aggregates articles from CNN) and **train_v2_drcat_02.csv** (a dataset containing Argumentative Essays drafted by both humans and AIs).

To conduct this experiment, we decided to make our own datasets. For each of these datasets, we extracted the **first 30 words** (the 'amorce' or seed) and asked the **GPT-4o mini** model to generate the text. At the end, we deleted, the 30 first tokens to keep only the generated text.

Prompts for Text Generation We defined three distinct prompts to generate text continuations, each designed to test a different aspect of the LLMs' output predictability.

Prompt 1: Standard Continuation This serves as the baseline prompt for continuing the text seed.

```
Here is the beginning of a news article:
"{amorce}..."

Task: Continue this article
Write about 150 words.
Return only the continuation, without any introduction.
```

Prompt 2: The Whimsical Role Injection (Pirate Style) This prompt aims to introduce a **whimsical role** to the LLM to observe the model's reaction and determine if this deviation in style manages to "trick" it (i.e., make its text less predictable for the detector).

```
Here is the beginning of a news article:
"{amorce}..."

Task: Continue this article using pirate style and vocabulary.
Write about 150 words.
Return only the continuation, without any introduction.
```

Prompt 3: The Adversarial Human Style The goal of this prompt is to **trick the model as much as possible** by forcing it to adopt a highly sophisticated and non-robotic journalistic/essay-writing style, which is intended to be the most challenging to detect.

```
Here is the beginning of a news article: "{amorce}..."
```

```
Task: Continue this article acting as a skilled human journalist.
Focus on a natural flow, varying your sentence structure (mixing
short and long sentences) to avoid a robotic tone.
Do not use clichéd transitions like "In conclusion," "Furthermore,"
or "It is important to note."
Write about 150 words. Return only the continuation, without any introduction.
```

Sample Size For each of the dataset, we kept 100 human-written text, and we generate 100 new text, following one of the previous task. So each of the datasets contains 200 samples. For the *newscnn* and the *trainv2drcat02* , we generated the text following the the three previous prompts using the same text.

3.2. Models Used for Binoculars Score Calculation

At the time of implementation, there was a limited availability of high-performing open-source models. For the Binoculars framework, which requires a pair of similar yet distinct models (Base and Instruction-tuned versions are commonly used), we selected the following pairs:

Small-Scale Models (Gemma) for Size Impact Analysis We included a first pair of models, `google/gemma-3-27` and `google/gemma-3-270m-it` (Instruction-tuned), which are also recent and performant despite their smaller size. This pair was added specifically to investigate the impact of **model size** on the resulting Binoculars score.

Recent High-Performance Models (Mistral) We chose the following pair of very recent and highly performant models from Mistral AI: `mistralai/Ministral-3-3B-Instruct-2512` and `mistralai/Ministral-3-3B-Base`

Replication of the Original Paper's Models (Falcon) Finally, to provide a direct point of comparison and to replicate the findings of the original research paper *Spotting LLMs with Binoculars*, we utilized the model pair used by the authors:

3.3. Evaluation Metrics

Since text detection is a binary classification problem, common metrics like the Area Under the ROC Curve (AUC) or the F1-score are reported for completeness. However, the authors argue that these metrics do not fully capture real-world reliability. In practice, the most important quantity is the **true positive rate (TPR) at an ultra-low false positive rate (FPR)** - specifically at 0.01% - to avoid wrongly labeling human text as machine-generated.

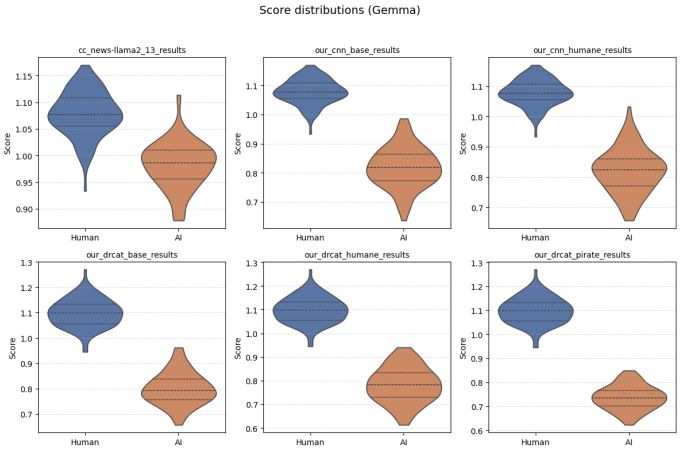
NB : Computation of TPR and FPR. For a binary classifier, let each sample be labeled as either *AI* (positive) or *Human* (negative). Given a detection threshold τ on the Binoculars score B, we define:

$$\mathrm{TPR}(\tau) = \frac{\mathrm{TP}(\tau)}{\mathrm{TP}(\tau) + \mathrm{FN}(\tau)}, \qquad \mathrm{FPR}(\tau) = \frac{\mathrm{FP}(\tau)}{\mathrm{FP}(\tau) + \mathrm{TN}(\tau)}.$$

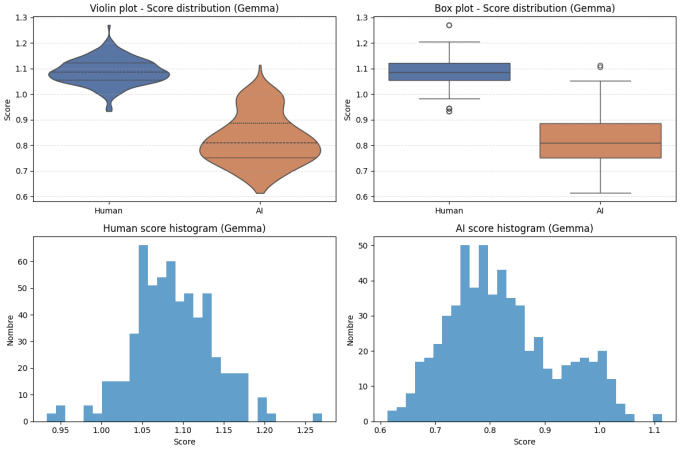
The threshold τ is applied directly to the Binoculars score B, which quantifies the ratio between two cross-entropy. Its optimal value is determined empirically and does not necessarily correspond to $B = 1$. $\tau^\star$ is chosen such that $\mathrm{FPR}(\tau^\star) = 0.01\%$, and the corresponding $\mathrm{TPR}(\tau^\star)$ is reported as TPR@FPR=0.01%. This ensures that the detector's performance is measured under an extremely low false-positive constraint.

We find the 2 optimal thresholds for 2 goals: one maximizing F1 score, the other minimizing FPR.

3.4. Quantitative Results



(a) Detailed score distribution views (Violin, Box, and Histograms) for Human vs. AI text (Google Gemma).

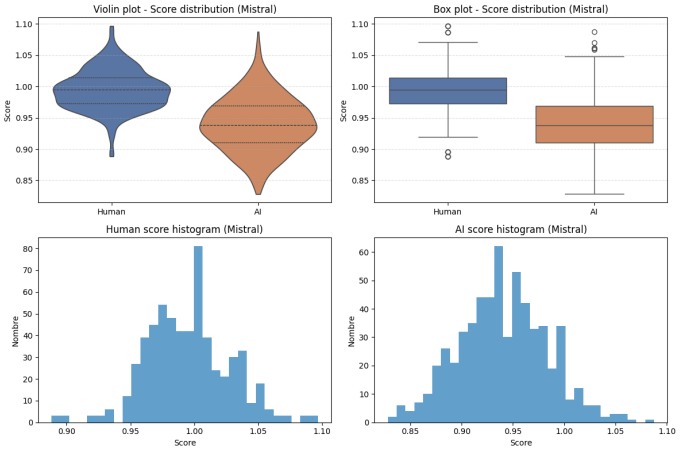


(b) Violin plot comparisons of score distributions across various datasets.

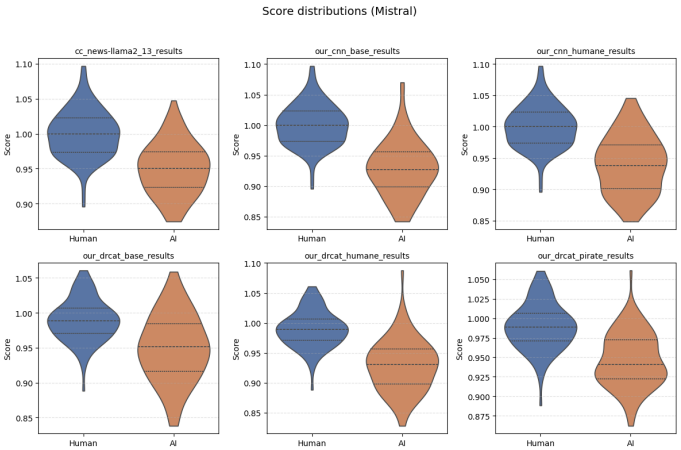
Figure **??** shows the scores distribution via violin, box, and histogram plots, for the Binoculars method with Gemma. The Violin Plot and the Box Plot show a distinct distributions of for human and AI texts. For human, the distribution are densely clustered around a high median value, approximately 1.08. This is also confirmed by the Histogram, which shows a sharp distribution with one peak near the median. This indicates the model consistently makes high observer score for human text.

For AI-Generated text, the score are significantly lower, centering a median value of around 0.75 to 0.85. Its distribution is notably broader and exhibits longer tail towards lower scores, indicating the higher variance applied by the observer model.

This separation in median scores (≈ 0.3) and the minimal overlap between the two distributions strongly suggested that the Binoculars method can perform consistently in the AI-generated text detection task.



(a) Detailed score distribution views (Violin, Box, and Histograms) for Human vs. AI text (Mistral).

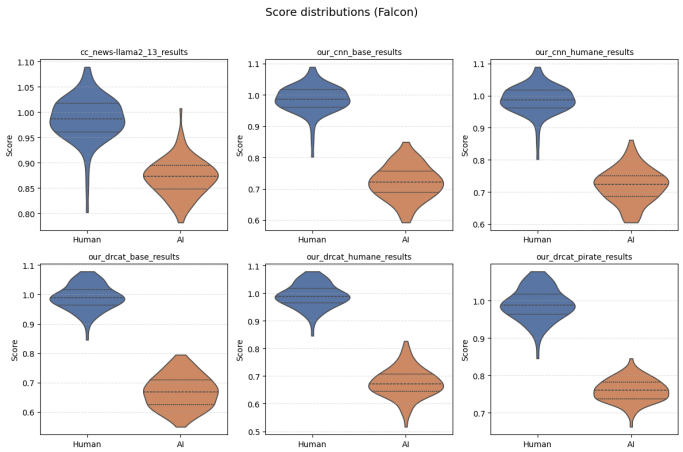


(b) Violin plot comparisons of score distributions across various datasets.

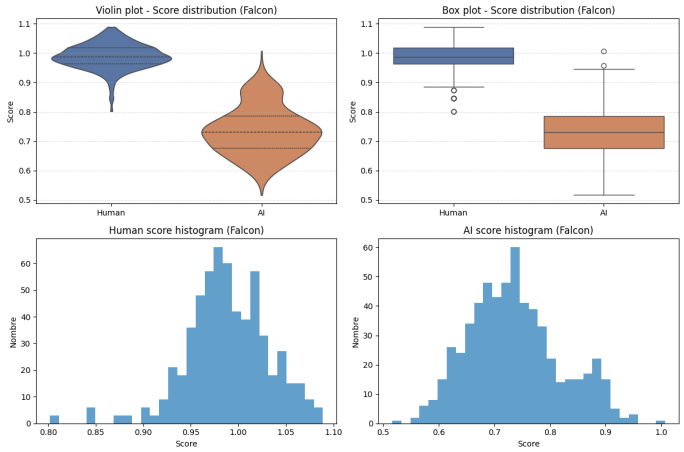
Mistral is currently the most ineffective for Binoculars, as evidenced by the wide distributions and large overlapping for the scores of both human and AI-generated texts. However, the characteristics of human distribution is still prevalent: concentrate at a high peak around 0.96, while the AI-generated samples has a wider distribution, centering around 0.92 to 0.97, and resembling a Gaussian mixture.

Overall, the median of these distribution still display a classification power: human texts have higher score in general. Also, we can see the effect of prompting affecting the difficulty of the task: for base results, we get consistent distribution, suggesting that non-specific prompted generations would not cause much impact for the classifier. However, once we used prompting use guide the generation towards: as human as possible, the

classifier instantly struggled more, as indicated by the much broader distribution of the AI-generated samples. However, with the DRCAT dataset, we observed that the AI-generated text distributions are spreading even more, and prompting the generated texts to be more human, caused a reverse effect where it's easier for the model to classify.



(a) Detailed score distribution views (Violin, Box, Histograms) for Human vs. AI text (Falcon).



(b) Violin plot comparisons of score distributions across various datasets.

Figure 3: **Binoculars method score distributions for Human vs. AI text.**

Falcon model is empirically the current best classifier so far, with consistent highest performance across all datasets, with the score of human and AI polarized: one at 1 and the latter at around 0.6.

Model Name	$B_{\textbf{low\_fpr\_opt}}$	$B_{\textbf{f1\_opt}}$
Gemma	0.943769	1.005392
Mistral	0.919118	0.968254
Falcon	0.845671	0.918431

Table 1: Optimal Threshold Scores for Binoculars Method Across Different Models

We also report the optimal threshold we found, under different objectives. Overall, they stays near the value of 0.9.

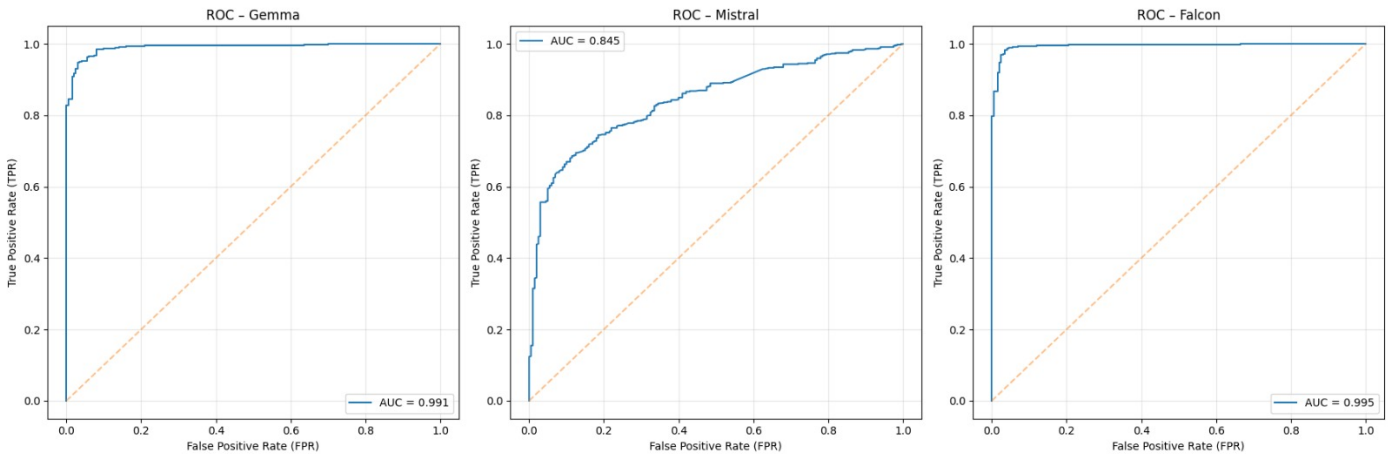


Figure 4: **Receiver Operating Characteristic (ROC) Curves for Binoculars Detection.**

The plots in Figure 4 compare the detection performance using Gemma (left), Mistral (center), and Falcon (right) models. The Area Under the Curve (AUC) is annotated in the bottom right of each subplot, indicating overall classification capability (higher is better). The dashed orange line represents random guessing performance (AUC = 0.5). As expected, The best performer is Falcon (AUC = 0.995), following by Gemma (AUC

= 0.991), then finally Mistral (AUC = 0.845) as the weakest performer.

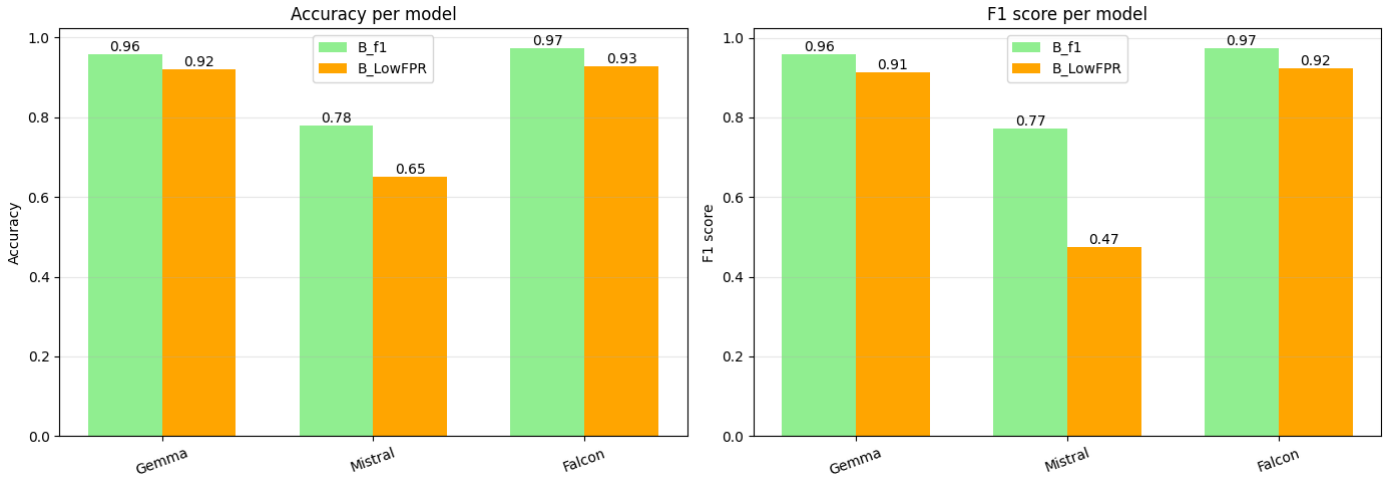


Figure 5: **Accuracy Comparison across 2 choices of threshold.**

Gemma 270M. Despite its relatively small parameter count, Gemma achieves consistently strong detection performance across both operating points. This indicates that compact models can still produce generation patterns that remain statistically distinguishable from human-written text. Overall, Gemma demonstrates that model size alone is not sufficient to evade likelihood-based detection.

Ministral-3-3B. Mistral, while considered state-of-the-art in terms of generative quality, exhibits the weakest detection performance, particularly under the low false-positive rate regime. Its strong alignment with human writing style appears to reduce the divergence between observer and performer likelihoods exploited by the Binoculars method. Consequently, higher generative fluency is associated with increased difficulty in reliable detection.

Falcon 7B. Falcon-7B achieves a good detection performance, with high accuracy and F1 scores across both threshold strategies. However, the model has a lot of parameteers to achieve the same performance as Gemma 270M. But this can be explained because the model is older than Gemma but it still gives good performance.

3.5. Supervised Classifier Implementation

As an alternative to the zero-shot Binoculars approach, we implemented a dedicated supervised classifier to detect machine-generated text, allowing us to compare the robustness and efficiency of the two methodologies.

To benchmark the robustness of Binoculars, we compared it against a standard supervised classifier. We add and fine-tuned a linear layer on top of the `OrdalieTech/Solon-embeddings-mini-beta-1.1` embedding model (210M parameters), keeping the backbone frozen. The model was trained for 10 epochs on the **HC3 dataset** (24k QA pairs), a standard resource for LLM detection.

We evaluated this classifier on the same test sets used for Binoculars. As shown in Table 2, the supervised model shows severe limitations in generalization.

Table 2: Supervised Classifier Performance (Solon Model)

Dataset	Precision Human	Precision AI	Total Accuracy
DRCAT (Essays)	8.00%	100.00%	54.00%
DRCAT Pirate	8.00%	52.00%	30.00%
CNN (News)	70.00%	97.00%	83.50%
CNN Pirate	70.00%	66.00%	68.00%

While the classifier performs decently on standard news articles (CNN), it fails catastrophically on out-of-distribution data such as student essays (DRCAT) or stylized text (Pirate prompt), dropping to random guessing or worse (30% accuracy).

Conclusion: This comparison highlights the key strength of Binoculars. Unlike a supervised classifier that overfits to specific training domains and fails when the writing style shifts, Binoculars' zero-shot approach remains robust and effective across diverse domains and creative prompts without requiring retraining.

4. Conclusion

Despite our initial skepticism regarding the efficacy of the method, the results obtained with Binoculars have proven to be highly convincing. The method demonstrated remarkable robustness, successfully handling challenging adversarial datasets such as the "Pirate" style and obfuscated human texts—where our supervised classifier struggled to generalize.

The primary advantage of Binoculars lies in its **zero-shot** nature. It requires neither a training phase nor the creation of specific labeled datasets, which inherently eliminates the risk of overfitting to a particular domain.

Our experiments highlighted two particularly surprising findings:

- **Model Efficiency:** Even very small models (such as Gemma 270M) achieved competitive results.
- **Architecture Performance:** Most unexpectedly, the **Falcon 7B** model (both Base and Instruct versions) yielded the best overall performance, outshining more recent architectures despite being an older model.

However, this method is not without drawbacks. The main trade-off for its zero-shot versatility is its **computational cost**. Unlike a lightweight supervised classifier, the Binoculars method requires significantly more resources during the inference phase to compute the perplexity scores, which is a factor to consider for real-time applications.

5. Disclosure of LLM Utilization

In this work, we used LLM for helping us understand the concepts, polishing words choice, and aid in analyzing the results. We used LLM to generate our datasets, and to help us on some codes.

References

[1] Hans, S., Meng, Y., Zhao, S., Wang, Y. (2024). *Spotting LLMs with Binoculars: Zero-Shot Detection of Machine-Generated Text.* In **Proceedings of the 41st International Conference on Machine Learning (ICML 2024)**. URL: `https://arxiv.org/abs/2402.17876`