MOSAIC at GENAI Detection Task 3 : Zero-Shot Detection Using an Ensemble of Models

Matthieu Dubois
Sorbonne Université, CNRS, ISIR
duboism@isir.upmc.fr

François Yvon
Sorbonne Université, CNRS, ISIR
yvon@isir.upmc.fr

Pablo Piantanida
International Laboratory on Learning Systems (ILLS)
Quebec AI Institute (MILA)
CNRS, CentraleSupélec, Université Paris-Saclay
pablo.piantanida@mila.quebec

Abstract

The rapid advancement of Large Language Models (LLMs) has raised issues concerning the misuse of their text generation abilities for creating forged content, fostering the need for reliable detection methods. While most methods are *supervised* and require training samples of human vs. artificial texts, we propose instead to consider *unsupervised detection* approaches. In a nutshell, most unsupervised methods rely on one or several detector model(s), whose (low) perplexity scores serve as a signal of machine-generated contents. Such approaches can be brittle as their performances strongly depend on the choice of a particular detector. To address these limitations, we evaluate a method *for combining multiple detectors* and enhance robustness. In this submission, we report evaluation results on the RAID benchmark, a comprehensive English-centric testbed for machine-generated texts. These results were obtained in the context of the "Cross-domain Machine-Generated Text Detection" shared task. We show that our approach can be competitive for a variety of domains and generator models, but also that it is challenged by adversarial attacks and by changes in the text generation strategy.

1 Introduction

Large Language Models (LLMs) have greatly improved the fluency and diversity of machine-generated texts. The release of ChatGPT and GPT4 by OpenAI has sparked global discussions regarding the effective use of AI-based writing assistants. This progress has also introduced major threats related to the generation of fake news (Zellers et al., 2019), of toxic or dishonest content (Crothers et al., 2023), or more generally regarding misuses of machine generation abilities. In response, the automatic detection of such Machine Generated Texts (MGT) has attracted a lot of recent work.

From a bird's eye view, MGT detection uses *detector* models to discriminate *generator* models' outputs from human writings. Multiple instances of this basic text classification problem have been considered, varying e.g. the number of possible categories to distinguish, the amount of available supervision or the granularity of the task (e.g. at the text, sentence, or even token level). Owing to its large user base and applications, most efforts to date have focused on specifically detecting ChatGPT, for which training and test data is easily obtained. A more difficult problem, that we study here, is **unsupervised generator-agnostic artificial text detection**, where the models to detect are not known in advance, and for which we also assume no training data.

As pointed out e.g., in (Antoun et al., 2024; Hans et al., 2024; Wang et al., 2024), the performance of MGT detection systems varies depending on the choices of the detector(s) / generator(s) pair. The detector may serve to assess probabilities, as in (Mitchell et al., 2023; Bao et al., 2024), or to regenerate content, as e.g., in (Mao et al., 2024; Yang et al., 2024). In most cases, optimal detection performance will require a systematic exploration of the space of possible detectors. As the number and diversity of LLMs keep increasing, such exploration seems not only challenging but also unrealistic. Furthermore, (Dugan et al., 2024) demonstrated that the current detection methods are brittle and easily fooled by changing the generator or altering the associated sampling method, a finding that we reproduce in this study.

In an attempt to increase the robustness of existing detectors, we consider here *ensemble methods*, where a coalition of several models is exploited to build the detector. For this, we generalize perplexity-based approaches, which flag as "artificial" texts having a suspiciously small perplexity.

Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect), pages 371–376
January 19, 2025.

As perplexity is also an encoding measure, our ensemble technique will seek to identify time-varying mixture models, in order to minimize the worst-case expected encoding size. The corresponding architecture is in Figure 1. Further details, explanations, and proofs can be found in a companion paper (Dubois et al., 2024). Not only is this method fully unsupervised, it also dispenses with the need to search for the best detector(s). This method nonetheless helps to develop MGT detection systems that can robustly detect multiple generators. In this short contribution, we briefly present the detection task, then our detection algorithms, before discussing and analyzing the experimental results.

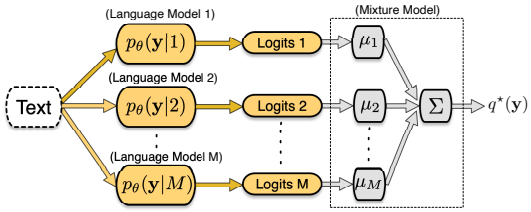

Figure 1: Mixture Model.

2 Data and Task Description

2.1 The Task

The Cross-Domain Machine-Generated Text Detection Task at the COLING 2025 Workshop on Detecting AI-Generated Content (Dugan et al., 2025) aims to challenge detection methods on the RAID test dataset (Dugan et al., 2024), containing samples spanning multiple domains, generators, and decoding methods. The dataset can be split into two parts, each corresponding to a separate subtask:

- Subtask A : Original human-authored texts and MG counterparts, one for each model and decoding method proposed in the RAID dataset;
- Subtask B : The same data with adversarial attacks such as misspellings or paraphrasing.

As our method scores texts in a zero-shot manner, we use the same detection model for both subtasks. The metric used in the competition is the True Positive Rate when the False Positive Rate equals 5% (TPR@5%FPR). For this metric, our method simply outputs one score per text, with larger scores corresponding to natural texts, and lower scores to artificial texts.

2.2 Data Description

Both training and testing sets for this task are described in (Dugan et al., 2024). RAID is a comprehensive benchmark designed to assess the robustness of MGT detection systems. The test set contains about 6 million generated texts produced by 11 models, across 8 domains (Arxiv Abstracts, Book Plot Summaries, BBC News Articles, Poems, Reddit Posts, Recipes, IMDb Movie Reviews and Wikipedia Articles). Each human-written document was paired with a generation prompt used to produce outputs for all models, employing both zero-shot chat and non-chat templates depending on each model's intended usage. When applicable, multiple decoding methods were used, e.g. greedy decoding or ancestral sampling, also varying the repetition penalty for a total of 4 combinations. To further challenge detectors, each text is assigned variations, using 11 types of adversarial attacks such as paraphrasing, alternative spelling, and synonym replacements. As each human entry gets a corresponding version for each model and available decoding strategy, the dataset is mostly comprised of machine generations. When adding the adversarially attacked variations, they make up the majority of the data.

3 Our Method : MOSAIC

Language models predict the probability of a token conditioned on the preceding ones, thus defining a probability distribution over the set of all possible sequences. The probability of generating a sequence $\mathbf{y} = \langle y_0, y_1, \ldots, y_T \rangle$ is computed as the product of conditional probabilities for each token, given its preceding context.

A central concept in our method is **information**, which measures the "surprise" of observing a particular token for a model parameterized by θ. This surprisal is quantified as $-\log p_\theta(y_t|\mathbf{y}_{<t})$, where lower values indicate higher predictability. This is akin to compression in information theory: the lower the surprisal, the better the corresponding token can be compressed by the model $p_\theta(.|\mathbf{y}_{<t})$.

Instead of relying on a detector single model, as in most unsupervised methods, our method leverages a diverse set of LLMs, denoted as $\mathcal{P}_\mathcal{M}(\mathcal{Y})$. The key idea is to assign each token in a sequence to the model that best "explains" it, i.e., the model that can compress it most effectively. Given a sequence $\mathbf{y}_{<t}$, we combine the models logits to obtain $q_t^\star$, the distribution minimizing the excess codelength w.r.t

any distribution $p_\theta \in \mathcal{P}_\mathcal{M}$.

$$q^\star(y_t|\mathbf{y}_{<t}) \triangleq \arg\min_{q\in\mathcal{P}(\Omega)} \max_{m\in\mathcal{M}} \mathcal{R}_\theta(m, q, \mathbf{y}_{<t})$$

$$\mathcal{R}_\theta(m, q\ \mathbf{y}_{<t}) \triangleq \mathbb{E}\Big[-\log q(y_t|\mathbf{y}_{<t})\Big] - \mathcal{H}_\theta(Y_t|m, \mathbf{y}_{<t})$$

where Ω is the model vocabulary, $\mathcal{H}_\theta(Y_t|m, \mathbf{y}_{<t})$ is the conditional entropy, and the expectation $\mathbb{E}$ is computed over $y_t \sim p_\theta(y_t|m, \mathbf{y}_{<t})$. It can be shown that the optimal distribution is a mixture distribution, whose weights are computed by the Blahut-Arimoto algorithm (Blahut, 1972; Arimoto, 1972). Our scoring method then evaluates the difference in codelength between the observed text on the one hand, and an averaged equivalent measure for generated texts (for models in $\mathcal{M}$), using:

$$S_{\text{Av}}(\mathbf{w}) = \frac{1}{TM}\sum_{t=1}^{T}\sum_{m\in\mathcal{M}}\Big[\mathcal{L}^\star(w_t|\mathbf{w}_{<t}) - \sum_{y_t\in\Omega} p_\theta(y_t|m, \mathbf{w}_{<t})\mathcal{L}^\star(y_t|\mathbf{w}_{<t})\Big], \quad (1)$$

where $\mathcal{L}^\star(y_t|\mathbf{w}_{<t})$ represents the optimal codelength obtained using a mixture of models. The first term measures how well the optimal model compresses the actual input text, while the second term captures how well the LLMs in the set $\mathcal{M}$ compress typical machine-generated outputs.

A larger score indicates that the text is likely human-written because the observed codelength is significantly larger than what would be expected from AI-generated content. Conversely, a lower score suggests that the text closely resembles outputs from the models in $\mathcal{M}$. Since this competition requires the machine outputs to get higher scores, we simply inverted use a negated version of (1) before submission. More information about MOSAIC can be found in (Dubois et al., 2024).

4 Results and Discussion

4.1 Experimental Setup

For our submission to the shared task, we used two settings: MOSAIC-4 and MOSAIC-5, where the former uses an ensemble of four models (Tower-7b, Tower-13b (Alves et al., 2024), Llama-2-7b and Llama-2-7b-chat (Touvron et al., 2023)), and the latter additionally includes Phi-3-4k-Instruct (Abdin et al., 2024). As the gold labels were not provided, the results discussed below are based on the online leaderboard,[1] reporting the True Positive Rate @ 5% False Positive Rate.

4.2 Results

As our method requires no training at all, we went with the default implementation without ever looking at the training set. This explains why our standings are below other teams who perform tuning on the RAID training set. However, our results are competitive with a similar method evaluated on the RAID leaderboard, the Binoculars approach of Hans et al. (2024). Binoculars obtains a score of 0.790 without adversarial attacks, while MOSAIC-4 and 5 get 0.752 and 0.745 respectively. In the following, unless explicitly specified, we report the results of MOSAIC-4, our default version.

The detailed results for the sampling configurations and various attacks are shown in Figure 2. The values presented for adversarial attacks are averaged across all decoding methods, making them directly comparable to the "all" setting in the table displaying the sampling results.

4.2.1 Impact of the decoding strategy

When **Greedy Decoding** is used, generated texts are very unsurprising, thus our method gets great overall results (over 95% of TPR@5%FPR on average). However, texts generated by GPT2, Mistral, and MPT are harder to reliably detect, getting scores of 0.781, 0.900, and 0.897 respectively. We can only speculate that GPT2 is the furthest from our ensemble's distribution. We notably obtain a perfect score on Llama-chat, which makes sense since our models are Llama-2 variations, the generator in this case is arguably the closest to the probability distribution provided by the models of our ensemble.

In the case of **Ancestral Sampling**, the irregularities added to the text led to a drop in performance for our method, with MOSAIC-4 and MOSAIC-5 scores falling down to 0.785 and 0.799. Not only do the worse generator models (GPT2, Mistral and MPT) become even harder to identify (0.333, 0.571, and 0.609 respectively), but GPT4 generations also join them, as MOSAIC-4 results go from 0.979 to 0.584 when changing the decoding method. Llama-chat texts are still (near)-perfectly identified (TPR@5%FPR=0.999), and so are MPT-chat's. They happen to be the only two open-source models' instruct versions in

[1] https://raid-bench.xyz/shared-task

the generators, allowing us to speculate that either these versions' outputs do not significantly differ when switching from greedy to sampling, or that our ensemble's distribution is very suited to these instruction-tuned models.

Adding **Repetition Penalty** when greedy decoding is used does not change our results much except for MPT generations, the detection score of which drops drastically from 0.897 to 0.343. Similarly, MPT-chat's score goes from near perfect (0.996) to second-to-last (0.621). However, combining sampling and repetition penalty makes the generated text very surprising and completely breaks our detection approach, leading to results close to 0 for GPT2, Mistral, and MPT (0.005, 0.002 and 0.018). Even in this scenario, Llama-chat remains easy to detect, keeping our average results afloat with a score of 0.864.

4.2.2 Adversarial Attacks

As the golden labels are not provided, we can only hypothesize that the test set is constructed in a similar way as the training set. If that is the case, the attacks are also applied to the human texts and thus produce interesting results. While changing the decoding method could only affect the machine-generated outputs, adversarial techniques modify all samples and can sometimes make human texts even more surprising, improving our results. We report scores on average over all decoding techniques, i.e., when decoding strategy and repetition penalty are both set to "all". MOSAIC-4 goes from 0.752 without attacks to 0.693 on average over all of them, while MOSAIC-5 drops from 0.745 to 0.694. Unless otherwise specified, the scores mentioned below correspond to MOSAIC-4.

Most attacks only cause a slight performance decrease, indicating that they add more surprise to machine outputs than human ones. Shuffling numbers, inserting paragraph breaks, switching between the British spelling and American spelling of some words, deleting some articles, and adding common misspellings are adversarial techniques that lead to score drops lower than 0.05. **Swapping the lower or upper case of words and adding spaces between characters** have more impact on our results but these changes remain minor. Both methods decrease our performance by about 0.07.

Swapping tokens with synonyms chosen by BERT is by far the best attack against our detection method. As pointed out in the DetectGPT paper (Mitchell et al., 2023), synonyms have lower model log-probability on average in machine-generated samples while human-written text does not exhibit this tendency. This heavily disrupts our method's underlying assumptions and makes our TPR@5%FPR drop down to 0.285.

Using Homoglyphs leads to an interesting outcome, as the attack actually improves our performance, making MOSAIC-5 the best performing submission of the competition when only considering homoglyphs attacks for some generators (ChatGPT, Cohere, Cohere-chat and Llama-chat). We suspect it is due to the Tower models having seen Cyrillic data in their training.

Inserting zero-width space is the most peculiar of the lot, as it leads to a slight MOSAIC-4 deterioration and a MOSAIC-5 improvement. Our interpretation is that Phi-3 saw this Unicode character during its pretraining, while the other models in our ensemble probably did not.

Overall, our method proves to be quite resilient to adversarial attacks even though it was not designed for this purpose as we operate in a completely zero-shot and tuning-free setting. This further demonstrates the robustness of our method.

The detailed results for the sampling configurations and various attacks are shown in Figure 2. The values presented for adversarial attacks are averaged across all decoding methods, making them directly comparable to the "all" setting in the table displaying the sampling results.

5 Conclusion

About our system and its underlying models In this task, we used the MOSAIC scoring algorithm presented in equation 1, using either 4 of 5 models, (Tower-7b, Tower-13b Llama-2-7b and Llama-2-7b-chat with Phi-3-4k-Instruct in the 5 models version). None of these are used as generators in the RAID test set and they are all Llama-2 variants, as mentioned in (Alves et al., 2024). The only Llama model present in the dataset is Llama-chat, and is the easiest generator to detect according to the competition results. Furthermore, the whole test set is in English and 5 of the 11 generator models are chat versions. The assumption behind our score is that the generated texts' distribution are close to our models', considering two of our members are specialized in multilingualism, and only one is a chat version, our ensemble choice was not optimized for this task at all. We believe this showcases the generalization capabilities of the method. Further

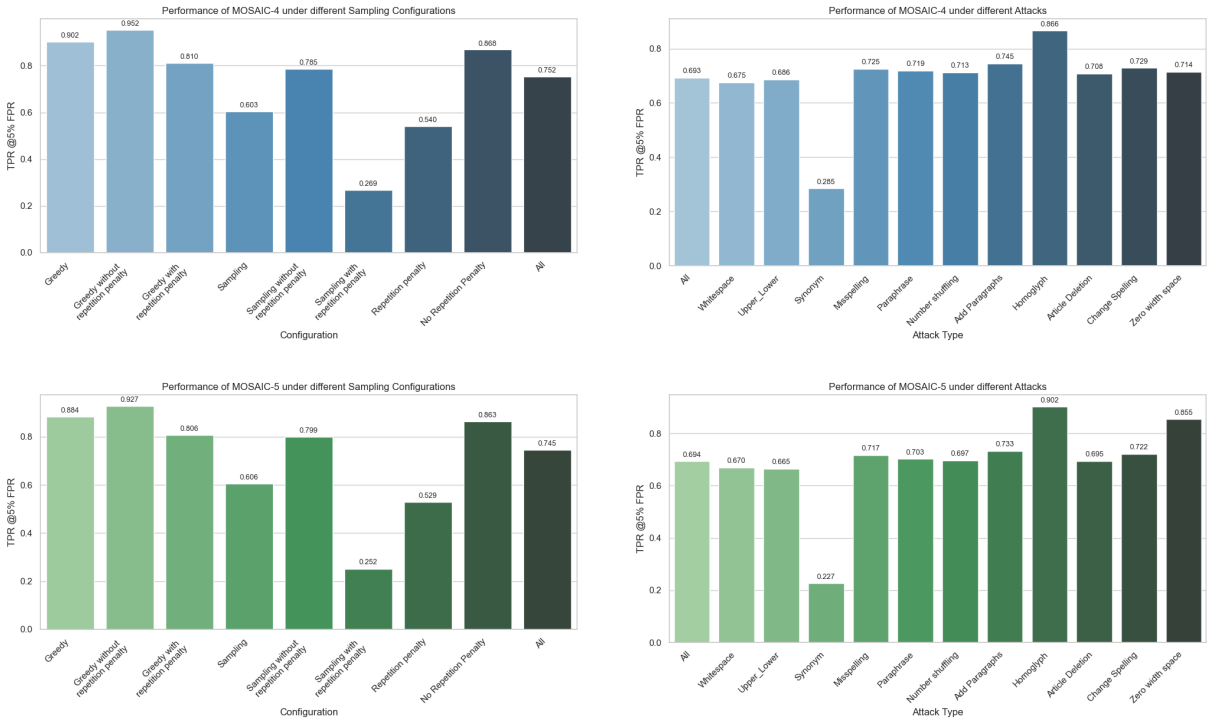

Figure 2: TPR@5% for all Sampling options and Attacks for both MOSAIC configurations

details can be found in the MOSAIC paper (Dubois et al., 2024).

Insights gained from this competition Participating in this shared Task 3 with our MOSAIC method has allowed us to gain valuable insights as to how our method fares against opponents using supervised methods, and take a better look at the effects on detectability of the decoding techniques used to generate the text. Since we underlyingly use language model's probability distribution to identify machine outputs, we expected sampling to affect our performance, and it did. The same observation holds for the use of a repetition penalty and the combination of these generation parameters. However, the adversarial attacks, which were never considered when developing our scoring system, only slightly weaken our results, confirming that our approach is robust to not only changes in the generator model and domain but also resilient to many forms of noise.

Acknowledgements

This work was performed using HPC resources from GENCI–IDRIS (Grant 2023-AD011014903)

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Qin Cai, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Yen-Chun Chen, Yi-Ling Chen, Parul Chopra, Xiyang Dai, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Victor Fragoso, Dan Iter, Mei Gao, Min Gao, Jianfeng Gao, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Ce Liu, Mengchen Liu, Weishung Liu, Eric Lin, Zeqi Lin, Chong Luo, Piyush Madan, Matt Mazzola, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Xin Wang, Lijuan Wang, Chunyu Wang, Yu Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Haiping Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Jianwei Yang, Ziyi Yang, Yifan Yang, Donghan Yu, Lu Yuan, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. *Preprint*, arxiv:2404.14219.

Duarte Miguel Alves, José Pombal, Nuno M Guerreiro, Pedro Henrique Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and Andre Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. In *First Conference on Language Modeling*.

Wissam Antoun, Benoît Sagot, and Djamé Seddah. 2024. From text to source: Results in detecting large language model-generated content. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7531–7543, Torino, Italia. ELRA and ICCL.

Suguru Arimoto. 1972. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 18(1):14–20.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*.

Richard Blahut. 1972. Computation of channel capacity and rate-distortion functions. *IEEE Transactions on Information Theory*, 18(4):460–473.

Evan N. Crothers, Nathalie Japkowicz, and Herna L. Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 11:70977–71002.

Matthieu Dubois, François Yvon, and Pablo Piantanida. 2024. Zero-shot machine-generated text detection using mixture of large language models. *Preprint*, arXiv:2409.07615.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492, Bangkok, Thailand. Association for Computational Linguistics.

Liam Dugan, Andrew Zhu, Firoj Alam, Preslav Nakov, Marianna Apidianaki, and Callison-Burch Chris. 2025. Genai content detection task 3: Cross-domain machine generated text detection challenge. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, Abu Dhabi, UAE. International Conference on Computational Linguistics.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting LLMs with binoculars: Zero-shot detection of machine-generated text. In *Forty-first International Conference on Machine Learning*.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: geneRative AI detection viA rewriting. In *The twelfth international conference on learning representations*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature. In *Proceedings International Conference on Machine Learning*, ICML.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *Preprint*, arxiv:2302.13971.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. 2024. DNA-GPT: Divergent n-gram analysis for training-free detection of GPT-generated text. In *The twelfth international conference on learning representations*, (ICLR), Vienna, Austria.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending Against Neural Fake News. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.