MOSAIC : Mélange d'experts pour la détection de textes artificiels

Matthieu Dubois

03/07/25 - CORIA-TALN

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

Introduction

Présentation

Matthieu Dubois, doctorant à l'ISIR (CNRS, Sorbonne Université), sous la direction de François Yvon (ISIR) et Pablo Piantanida (ILLS, MILA).
MOSAIC, une méthode ensembliste pour la détection de textes générés à l'aide de modèles de langues, ACL Findings 2025.

Définition du problème

La détection de texte générés se décline actuellement sous les formes suivantes :

- Classification binaire : **Humain** ou **Machine**
- Identification du modèle auteur : **Humain** vs **ChatGPT**, **Llama**, **Mistral** . . .
- Granularité plus fine : étiquettes au **niveau du mot**

MOSAIC est une méthode qui fonctionne surtout pour le premier cas, mais par construction peut être appliquée au niveau des phrases ou des mots.

Détection Supervisée

Le cadre le plus basique est celui de la **détection supervisée**, où l'on dispose d'un ensemble d'entraînement. Des modèles Transformer tels que RoBERTa sont généralement utilisés, ce qui conduit à d'excellents résultats (proches de la perfection).

Un exemple : le classement RAID

Détecteur	Moy.	Modèle générateur										
		chatgpt	gpt4	gpt3	gpt2	mistral	mistral-c	cohere	cohere-c	llama-c	mpt	mpt-c
Leidos	0.977	0.994	0.996	0.993	0.979	0.963	0.989	0.927	0.945	0.988	0.975	0.987
Pangram	0.977	0.996	0.996	0.968	0.986	0.957	0.994	0.916	0.948	0.998	0.967	0.989

"-c" dénote les modèles « chat ». Métrique utilisée : TPR@5%FPR. Tableau complet sur https://raid-bench.xyz/shared-task, jeu de données introduit par Dugan et al. (2024).

Réécriture

- Demander à un LLM de réécrire le texte
- Plus la réécriture est différente du texte original, plus on considère que l'original était humain.
- Ne fonctionnera pas si votre modèle est trop différent de celui utilisé pour générer le texte que vous souhaitez analyser

	Human	*GPT*	Input	Rewrite
News			Therefore, we seriously doubt that the regulations will work, meet consumer demands or be accepted by the private sector unless the administration radically changes its approach immediately,\" the alliance added.	Thus, unless the administration swiftly alters its approach, the alliance strongly questions the efficacy, consumer satisfaction, and industry acceptance of the regulations.
			It preserves the historical differentiation between the exchange markets and the over-the-counter markets,\" said Paul Raisler, a partner at Fried, Frank, Harris, Shriver & Jacobson, which represented the Banks of New York.	*Preserving the historical differentiation between the exchange markets and the over-the-counter markets,\" emphasized Paul Raisler, a partner at Fried, Frank, Harris, Shriver & Jacobson, who represented the Banks of New York.*
Creative Writing			The two started yelling back and forth. At me. At each other. I zoned out, closed my eyes. I blocked them out and drew my gun.	The two engaged in a heated exchange, their voices reverberating in my ears. They directed their anger towards me, throwing accusatory words in my direction. Simultaneously, they turned their anger on each other, unleashing a barrage of insults. Overwhelmed by the commotion, I instinctively tuned out their voices, shutting my eyes and concentrating on blocking out the chaotic scene unfolding before me. In an effort to regain control of the situation, I reached for my gun.
			I thought you superheroes wore costly designer suits? DeFrayne insisted.	*I thought you superheroes were all about those expensive designer suits?" DeFrayne insisted, intrigued*

Figure 1 – Raidar (Mao et al., 2024)

Méthodes non supervisées

Les méthodes non supervisées s'appuient sur des différences statistiques entre les textes générés et les textes ordinaires.

- Comptage de n-grammes
- Taille du vocabulaire
- Densité lexicale (Vocabulaire / Longueur)

Cependant, ces mesures nécessitent d'avoir des textes très longs, ce qui n'est pas le cas de la plupart des contenus en ligne.

Méthodes basées sur la perplexité

Intuitivement, les textes rédigés par un LM devraient être moins « surprenants » (pour ce modèle de langage) que ceux écrits par des humains, on peut donc détecter les textes machines car leur perplexité est plus faible.

$$PPL(\mathbf{Y}) = \exp\left(-\frac{1}{T}\sum_{t=1}^{T}\log p_\theta(y_t \mid \mathbf{y}_{<t})\right)$$

Où $\mathbf{Y} = (y_1, \ldots, y_T)$ sont les tokens du texte, et $p_\theta(y_t \mid \mathbf{y}_{<t})$ représente la probabilité du token y_t connaissant le contexte $\mathbf{y}_{<t}$.

Utilisation de la perplexité

Seuil

Les premières méthodes de détection (par ex. GPTZero) utilisaient simplement un seuil δ sur la perplexité :

$$\begin{cases} \text{Humain} & PPL(\mathbf{Y}) \geq \delta, \\ \text{Généré} & PPL(\mathbf{Y}) < \delta. \end{cases}$$

Cela fonctionne encore très bien pour des générations simples et des modèles basiques, mais devient moins fiable avec différents paramètres de décodage et des modèles plus récents.

DetectGPT

Mitchell et al. (2023) développent davantage l'intuition de la « surprisal », en partant du principe que des perturbations rendent les textes artificiels moins probables, et proposent l'algorithme suivant :

- **Entrée :** passage $\mathbf{Y}$, modèle source q, fonction de perturbation p, nombre de perturbations k, seuil de décision ϵ
- Générer des perturbations : $\tilde{y}_i \sim q(\cdot \mid \mathbf{Y})$ pour $i \in [1..k]$
- Calculer la log-vraisemblance moyenne : $\tilde{\mu} \leftarrow \frac{1}{k} \sum_i \log q(\tilde{y}_i)$
- Estimer la différence : $\hat{d}_y \leftarrow \log q(\mathbf{Y}) - \tilde{\mu}$
- Calculer la variance : $\tilde{\sigma}_y^2 \leftarrow \frac{1}{k-1} \sum_i \big(\log q(\tilde{y}_i) - \tilde{\mu}\big)^2$
- **Si** $\dfrac{\hat{d}_y}{\sqrt{\tilde{\sigma}_y}} > \epsilon$ **alors** renvoyer `true`, **sinon** renvoyer `false`

Exemple rapide

Passage candidat $\mathbf{Y}$: « Joe Biden a récemment déménagé à la Maison-Blanche en emmenant avec lui son chien »

- Perturbation : $\tilde{y}_1$= "~~chien~~" $\rightarrow$ "chat", . . ., $\tilde{y}_k$= "~~emmenant~~" $\rightarrow$ "apportant"
- Score : $q(\mathbf{Y}), q(\tilde{\mathbf{Y}}_1), \ldots, q(\tilde{\mathbf{Y}}_k)$
- Comparaison : $\frac{1}{k}\sum_{i=1}^{k} \log \frac{q(\mathbf{Y})}{q(\tilde{\mathbf{Y}}_i)}$ et ϵ

FastDetectGPT

Plutôt que de calculer des perturbations, FastDetectGPT (Bao et al., 2024) échantillonne de manière indépendante N séquences depuis un autre modèle $\{\tilde{y}_i \sim p(\mathbf{Y}_t \mid \mathbf{y}_{<t})\}_{i=1}^{N}$, calcule l'entropie croisée empirique et renvoie ensuite le score :

$$S_{p,q}^{\text{Fast}}(\mathbf{Y}) = \frac{-\log q(y_t \mid \mathbf{y}_{<t}) + \frac{1}{N}\sum_{i=1}^{N} \log q(\tilde{y}_i \mid \mathbf{y}_{<t})}{\tilde{\sigma}(\mathbf{y}_{<t})},$$

où

$$\tilde{\sigma}^2(\mathbf{y}_{<t}) \triangleq \frac{1}{N-1}\sum_{i=1}^{N}\Big(-\log q(y_i \mid \mathbf{y}_{<t}) + \frac{1}{N}\sum_{j=1}^{N}\log q(y_j \mid \mathbf{y}_{<t})\Big)^2$$

est un terme de normalisation.

Binoculars

En utilisant la véritable entropie croisée, Hans et al. (2024) ont conçu leur score Binoculars :

$$B_{p,q}(\mathbf{y}) \triangleq \frac{\sum_{t=1}^{T} \sum_{y \in \Omega} \mathbb{1}[y = y_t] \mathcal{L}_q(y_t \mid \mathbf{y}_{<t})}{\sum_{t=1}^{T} \sum_{y \in \Omega} p(y \mid \mathbf{y}_{<t}) \, \mathcal{L}_q(y \mid \mathbf{y}_{<t})},$$

avec $\mathcal{L}_q(y_t \mid \mathbf{y}_{<t}) = -\log q(y_t \mid \mathbf{y}_{<t})$. C'est mathématiquement équivalent à FastDetectGPT, mais sous forme de ratio plutôt que de différence.

Ces deux méthodes constituent l'état de l'art actuel, obtenant d'excellents résultats lorsque les modèles utilisés (p et q) sont adaptés au dataset.

Utilisation de plusieurs modèles : MOSAIC

Peut-on généraliser ces méthodes ?

Les deux méthodes utilisent une paire fixe de modèles p et q pour calculer les log-probabilités.

Cette dépendance à deux modèles engendre de la fragilité, en effet, selon le LLM utilisé pour générer les textes, la paire p, q optimale varie. Ces méthodes nécessitent donc un jeu de données de validation, et peuvent sous-performer quand il y a plusieurs modèles générateurs.

MOSAIC

Dans MOSAIC, pour éviter cette dépendance, nous généralisons ces méthodes à un ensemble de modèles :

- pour q, nous combinons tous les modèles de l'ensemble de façon à maximiser l'information mutuelle
- pour p, nous proposons un critère de choix du meilleur « modèle de référence »

Définition de $q^\star$

$$q^\star(y_t \mid \mathbf{y}_{<t}) \triangleq \sum_{m \in \mathcal{M}} \mu^\star(m \mid \mathbf{y}_{<t})\, p_m(y_t \mid \mathbf{y}_{<t}),$$

où $\mu^\star(\cdot \mid \mathbf{y}_{<t})$ satisfait :

$$\mu^\star(\cdot \mid \mathbf{y}_{<t}) \triangleq \arg \max_{\mu \in \mathcal{P}(\Omega)} \mathcal{I}_p\big(\mathbb{M}; Y_t \mid \mathbf{y}_{<t}\big).$$

Les coefficients $\{\mu^\star(m \mid \mathbf{y}_{<t})\}_{m \in \mathcal{M}}$ sont calculés via l'algorithme de Blahut–Arimoto (Arimoto, 1972; Blahut, 1972).

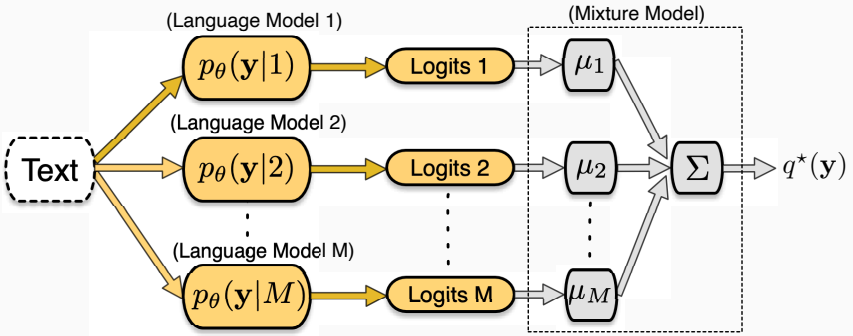

Figure 2 – Notre modèle en mélange

Choix de p

$$m^\star(\mathbf{y}_{\text{hum}}) \triangleq \arg\min_{m\in\mathcal{M}} -\sum_{t=1}^{T} \log p_m(y_t \mid \mathbf{y}_{<t}).$$

Autrement dit, le modèle de référence $p_{m^\star}$ doit être celui de l'ensemble $\mathcal{M}$ présentant la log-perplexité la plus faible pour des échantillons de texte « humains » $\mathbf{y}$.

Score MOSAIC

Pour une phrase d'entrée $\mathbf{y} = \langle y_0, y_1, \ldots \rangle$, et des modèles indexés par $\mathcal{M} = \{1, \ldots, M\}$ partageant un même tokenizer, le score MOSAIC est défini comme :

$$S_{m^\star,\mathcal{M}}(\mathbf{y}) \triangleq \frac{1}{T} \sum_{t=1}^{T} \sum_{y \in \Omega} \Big[\underbrace{\mathbb{1}_{\{y=y_t\}}\, \mathcal{L}_{q^\star}(y_t \mid \mathbf{y}_{<t})}_{\text{encodage du token observé}} - \underbrace{p_{m^\star}(y_t \mid \mathbf{y}_{<t})\, \mathcal{L}_{q^\star}(y_t \mid \mathbf{y}_{<t})}_{\text{encodage du token généré par } m^\star} \Big]$$

Expériences

Modèles utilisés

Nous avons utilisé 4 LLMs dans nos expériences :

- Llama-2-7b
- Llama-2-7b-chat
- TowerBase-7b
- TowerBase-13b

Ces 4 modèles partagent tous un même tokenizer, nécessaire pour le calcul de l'entropie croisée.

Nous comparons Binoculars et FastDetectGPT avec toutes les paires possibles (12 configurations), et MOSAIC utilisant les 4.

Jeux de données

Nous présentons des expériences sur le jeu de données RAID (Dugan et al., 2024), actuellement le plus complet pour la détection de textes générés, et M4 (Wang et al., 2024) pour ses textes multilingues.

Modèles de référence

	chatgpt	cohere-c	cohere	gpt2	gpt3	gpt4	llama-c	mistral-c	mistral	mpt-c	mpt
Bino (max)	0.996	0.985	0.979	0.812	0.999	0.969	1.000	0.998	0.915	0.999	0.946
Configuration	T13b/L-c	L/L-c	T13b/L-c	T13b/T7b	T13b/L-c	T13b/L-c	L/L-c	T13b/L-c	T13b/T7b	T7b/L-c	T13b/T7b
Bino (min)	0.511	0.688	0.711	0.459	0.945	0.376	0.741	0.560	0.609	0.661	0.637
Bino (moy)	0.837	0.900	0.870	0.652	0.983	0.720	0.928	0.876	0.774	0.895	0.798
Fast (max)	0.994	0.981	0.979	0.858	0.996	0.974	1.000	0.993	0.923	0.995	0.966
Configuration	T13b/L-c	L/L-c	L/L-c	T7b/L	L/L-c	T13b/L-c	L/L-c	T13b/L-c	T13b/L-c	T13b/T7b	L/L-c
Fast (min)	0.505	0.673	0.705	0.501	0.914	0.363	0.740	0.552	0.606	0.647	0.636
Fast (moy)	0.802	0.853	0.860	0.684	0.961	0.691	0.918	0.869	0.796	0.866	0.830

Table 1 – Résumé des AUROC de Binoculars et FastDetectGPT sur RAID. Les cellules « max », « moy » et « min » indiquent respectivement le score maximal, moyen et minimal parmi toutes les configurations. « T » et « L » désignent respectivement TowerBase et Llama-2-7b, tandis que « -c » correspond à la version « chat ».

Comparaison avec MOSAIC

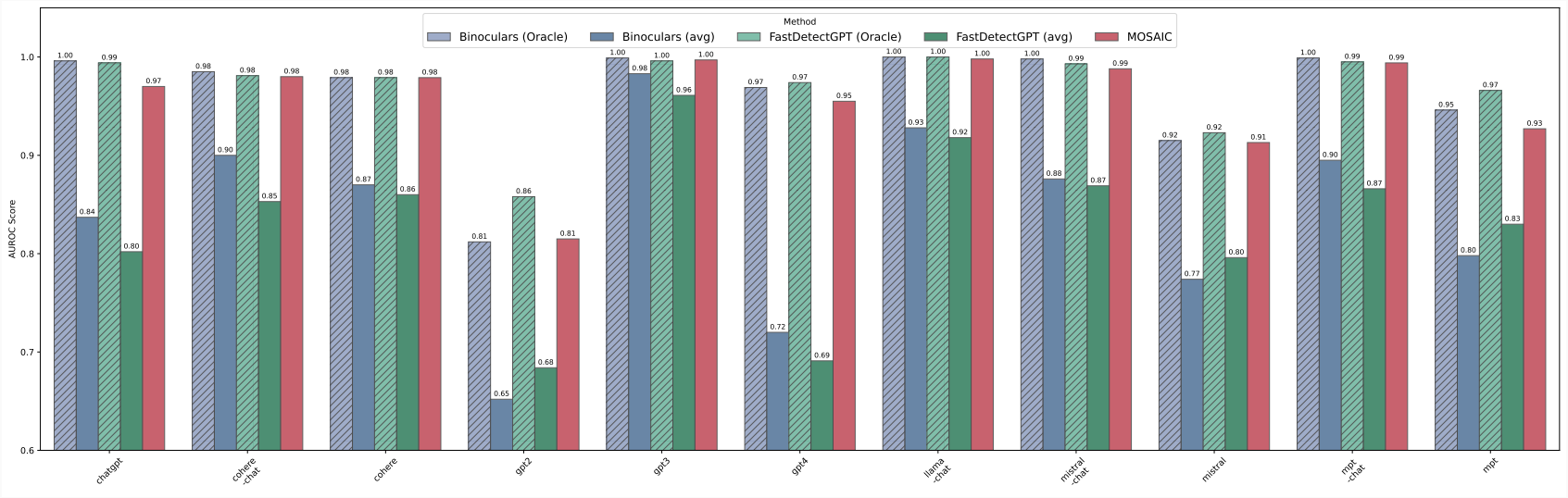

Figure 3 – AUROC de Binoculars, FastDetectGPT et MOSAIC sur RAID

Influence du modèle de référence

	Arabe	Bulgare	Chinois	Allemand	Russe
TowerBase-13B	**1.2743**	**1.8052**	2.3047	**1.4912**	**1.5069**
TowerBase-7B	1.3929	1.9839	2.3527	1.6169	1.6084
Llama-2-7b-chat	1.7379	2.3175	2.6800	2.1189	2.2917
Llama-2-7b	1.3506	1.8291	**2.1286**	1.6117	1.7778

Table 2 – Valeurs de log-perplexité de nos modèles pour les textes « humains » du jeu M4

Modèle m	Arabe	Bulgare	Chinois	Allemand	Russe
TowerBase-13B	**0.9563**	**0.9888**	**0.9752**	**0.9311**	**0.9148**
TowerBase-7B	0.9111	0.9578	0.9558	0.8679	0.8569
Llama-2-7b-chat	0.7768	0.8262	0.5849	0.6751	0.4321
Llama-2-7b	0.8947	0.9762	0.9059	0.9200	0.6814

Table 3 – AUROC de MOSAIC sur le jeu M4 en faisant varier le modèle « référence » $m^\star$

Résistances aux attaques

	homoglyph	nombre	article -	paragraphes	fautes	min/maj	espace	espace zéro	synonyme	paraphrase	GB/US
AUROC	0.961	0.936	0.920	0.952	0.948	0.928	0.927	0.754	0.681	0.944	0.947
TPR@5%FPR	0.749	0.736	0.693	0.785	0.771	0.699	0.707	0.007	0.315	0.752	0.771

Table 4 – AUROC et TPR@5%FPR de MOSAIC pour les différentes attaques sur RAID. Pour référence, MOSAIC obtient une AUC moyenne de 0.956 sur tous les générateurs sans attaques.

Merci de votre attention.

Références

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.

Références ii

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.

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.

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 of the 40th International Conference on Machine Learning, ICML'23. JMLR.org.

Références iii

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4GT-bench: Evaluation benchmark for black-box machine-generated text detection. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers), pages 3964–3992, Bangkok, Thailand. Association for Computational Linguistics.