Keystroke Dynamics et NLP : vers un détecteur hybride de textes IA

Matteo van Ypersele
ENSAE Paris 3A
matteo.vanyperseledestriho@ensae.fr

Abstract

Les détecteurs actuels de textes générés par IA reposent presque exclusivement sur le *produit fini*. Nous nous demandons si le *processus de rédaction*, observable via les journaux de frappe, peut fournir un signal complémentaire. À partir de OPENLLMTEXT (60 000 paragraphes humains / 60 000 ChatGPT) et en enrichissant chaque paragraphe avec 20 features de frappe extraites du corpus KEY LOG, nous comparons trois classifieurs légers : (i) un modèle RoBERTa-base affiné sur le paragraphe seul, (ii) un MLP de 30 000 paramètres alimenté uniquement par le vecteur de frappes, (iii) un réseau de fusion tardive concaténant les deux représentations. Sur un jeu de test indépendant de 10 375 paragraphes, le MLP sur les frappes atteint déjà un **F_1 de 0,994** et une **ROC-AUC de 0,997** avec 3,6 % de faux positifs, surpassant le baseline RoBERTa-seul (0,848 / 0,866, 22 % FP). Mais il faut garder à l'esprit que les transcripteurs n'ont pas cherché à dissimuler qu'ils transcrivaient. Une fusion naïve des deux vues atteint un **F_1 de 0,954** et une **ROC-AUC de 0,989**, mais fait bondir le taux de faux positifs à 28.38 %.

Preprint. Under review.

1 Introduction

L'essor récent des *Large Language Models* (LLM), tels que GPT-4 ou DeepSeek, transforme en profondeur les pratiques rédactionnelles. Dans le milieu scolaire, ces modèles permettent à un élève de générer en quelques secondes un essai "clé en main", limitant l'intérêt des devoirs maisons. Dans le secteur professionnel, les concours en ligne ou les tests de certification écrite deviennent tout aussi vulnérables : un candidat peut déléguer la rédaction de sa lettre de motivation, voire de sa note d'analyse, à un agent conversationnel sans laisser de trace apparente.

Les détecteurs actuels reposent presque exclusivement sur l'examen du *produit fini*. Ils mesurent la perplexité du texte ou appliquent un classifieur entraîné sur des paires *Humain / IA*. Pourtant, ces approches se laissent aisément tromper par de fortes paraphrases, des traductions aller-retour ou des réécritures manuelles qui brouillent la signature statistique de l'IA.

Nous défendons l'idée qu'un détecteur robuste doit prendre en compte le *processus* de rédaction. Les pauses, corrections et longueurs de rafales de frappe reflètent la charge cognitive d'un auteur humain et restent coûteuses à imiter pour un automate. Pour collecter ce signal, nous proposons une plateforme d'examen : une interface web verrouillée qui consigne, à la milliseconde près, chaque insertion, suppression et temporisation, puis agrège ces évènements en vingt métriques respectueuses de la vie privée (aucun caractère individuel n'est conservé).

Notre question de recherche est donc : *combiner ces métriques de frappe à des représentations linguistiques gelées améliore-t-il la détection d'un texte généré ?* Pour y répondre, nous (1) enrichissons le corpus public OPENLLMTEXT avec des journaux de frappe authentiques, (2) entraînons un classifieur fusion léger basé sur *RoBERTa-base* et (3) comparons ses performances à deux lignes de base, l'une texte-seule, l'autre méta-seule.

1.1 Détecteurs clé en main basés sur les LLM

Les premiers détecteurs d'IA reposaient sur le *perplexity-gap*: un même texte est évalué par deux modèles de tailles différentes et un écart substantiel de perplexité trahit une génération probable [Zellers et al., 2019]. Depuis l'essor des transformeurs, deux stratégies dominent. La première consiste à interroger *in situ* un LLM propriétaire (GPT-4, Claude-3) avec un prompt de détection *zero-shot*. Li et al. [2023] atteignent ainsi 96–98% d'accuracy sur des essais universitaires, mais la dépendance à l'API, la latence réseau et le manque d'explicabilité limitent un déploiement opérationnel. La seconde stratégie affine des modèles ouverts sur des jeux annotés. Dans cette lignée, les travaux pionniers de Solaiman et al. [2019] entraînent un GPT-2 small sur 250k paires *Humain/GPT-2* et obtiennent 84% d'exactitude sur de l'actualité courte.

Trois orientations se dessinent ensuite. (i) Les **classifieurs monomodaux Transformer** : Ippolito et al. [2020] adaptent `RoBERTa-base` sur un Reddit synthétique (Grover) et obtiennent 92% d'AUC, mais seulement 71% sur Wikipédia hors-domaine; sur HC3, Zhou et al. [2023] montrent qu'`ELECTRA-large` gagne quatre points de F_1 grâce à sa pré-formation orientée corruption. (ii) Les **ensembles hybrides** combinant probabilités Transformer et stylométrie : Mikros and Stamatatos [2023] fusionnent les sorties de `RoBERTa-large` et `ELECTRA-large` avec 92 indices classiques (MTLD, Yule-K, distribution POS) dans un Gradient Boosting, atteignant 95% d'accuracy et réduisant le FP-rate de 7 points sur TOEFL. (iii) Les **approches *entropy-gap*** : Tan and Wang [2023] ré-entraînent `GPT-NeoX-20B` pour prédire le log-ratio de sa propre perplexité versus un modèle plus petit, obtenant 97% d'AUC en condition *in-domain* mais chutant à 62% après paraphrase, soulignant la fragilité du signal produit-fini. En synthèse, l'état de l'art plafonne aujourd'hui à 95–97% d'AUC *in-domain*; la généralisation *cross-model* (GPT-4) et *cross-rewrite* (paraphrase, traduction) reste un défi [Mitchell et al., 2023].

Justification du choix `RoBERTa-base`. Nous recherchions un encodeur open-weights, référencé sur GLUE, mais suffisamment compact pour une inférence temps réel sur un seul GPU V100. Pré-entraîné sur 160G de tokens, `RoBERTa-base` (125M paramètres) dépasse `BERT-base` de trois points de F_1 moyen [Liu et al., 2019] tout en conservant la même empreinte mémoire. Un fine-tune léger reste compétitif face à des modèles plus massifs comme `ELECTRA-large` ou `DeBERTa-XL` [Zhou et al., 2023], pour un coût d'inférence environ deux fois inférieur. Enfin, geler les embeddings et la moitié des couches réduit de 60% les poids adaptables, limite le sur-apprentissage sur nos 52k paragraphes et divise par trois la durée d'entraînement. Malgré ces atouts, même des encodeurs

robustes demeurent vulnérables aux paraphrases ou traductions, d'où la nécessité de compléter le signal produit-fini par la dynamique de frappe.

1.2 Exploiter les journaux de frappe

La psycholinguistique a depuis longtemps établi que la durée des pauses, la structure des *bursts* de frappe et le profil des révisions reflètent la charge cognitive du rédacteur [Conijn et al., 2019]. Sur le jeu Villani, qui oppose 338 textes copiés à 416 courriels libres, [Conijn and Villani, 2019] parviennent déjà à distinguer les deux genres avec 78% de précision à partir de simples pauses entre caracteres. Un saut d'échelle est réalisé par Crossley et al. [2024], qui collectent 1000 dissertations argumentatives–500 authentiques rédigées par 4992 travailleurs MTurk et 500 transcrites mot-à-mot par un second groupe. Les frappes sont horodatées à 1ms dans une interface *walled-garden*, puis analysées avec INPUTLOG. Après filtrage des valeurs manquantes et suppression des colinéarités, 65 indicateurs demeurent; un *random forest* isole les 20 plus importants: pauses de 200ms et 2s, ratio produit/processus, insertions et révisions. Ce sous-ensemble suffit à atteindre F_1=0.99 (RF) et 0.98 (MLP), confirmant que le signal procédural est, à lui seul, extrêmement discriminant et résistant aux paraphrases. Ces résultats suggèrent qu'un détecteur centré sur le *processus*–plutôt que sur le seul produit fini–est structurellement plus difficile à contourner. Nous avons choisit d'utiliser le Multi Layer Perceptrons car il a eu dans notre cas une meilleur performance et rentrait plus dans le cadre du cours.

1.3 Fusion du produit fini et du processus : un champ encore ouvert

Très peu d'études tentent de croiser directement la surface textuelle et la dynamique de frappe. Li and Sun [2022] concatènent des embeddings BERT avec cinq mesures de pause dans un perceptron multicouche; sur 8000 courriels internes, l'ajout du signal procédural n'apporte qu'un gain modeste de deux points de F_1, sans généralisation inter-domaine. Afin d'améliorer la robustesse, Hems and Lex [2023] proposent une fusion tardive entre DetectGPT (*zero-shot*) et un random forest entraîné sur keystrokes: l'ensemble divise par trois le taux de faux positifs sur des essais universitaires, mais repose sur l'API GPT-4 et reste coûteux. En parallèle, Lopes et al. [2023] démontrent qu'ajouter simultané- ment les mouvements oculaires et les frappes réduit de moitié l'erreur de prédiction de la charge cognitive, ce qui suggère que ces indices comportementaux sont largement complémentaires.

En dépit de ces avancées, plusieurs questions demeurent : le gain persiste-t-il sur un corpus massivement *out-of-domain* et multi-auteur comme le notre. Une simple concaténation suivie d'une tête linéaire suffit-elle ou faut-il un mécanisme d'attention croisée ? Enfin, les vingt variables retenues par Crossley et al. [2024] généralisent-elles à des LLM plus récents tels que GPT-4 ou Claude-3? Notre travail se situe précisément dans cette perspective : nous évaluons à grande échelle une fusion tardive minimaliste – embeddings `RoBERTa-base` gelés associés à un MLP sur les 20 features clavier – et la comparons systématiquement aux baselines *texte-seul* et *processus-seul*.

2 Nos Données

Notre étude repose sur trois jeux de données complémentaires (Table 2).

2.1 OpenLLMText – Sous–corpus *Human*

Le jeu OPENLLMTEXT [Ren et al., 2023] regroupe 300 000 paragraphes issus de cinq sources. La portion *Human* contient 60 000 textes rédigés par des utilisateurs Reddit avant 2019 — soit 97.7 Mo compressés (archive `Human.zip`) Chaque entrée comprend : `id`, le `text` brut (longueur médiane 168 tokens), et des méta-données lexicales (`n_words`, `n_sents`, `n_syms`). Licence : CC-BY4.0.

2.2 OpenLLMText – Sous–corpus *ChatGPT*

Pour chaque paragraphe humain, les auteurs ont obtenu une paraphrase paragraphe-par-paragraphe via `gpt-3.5-turbo` (mars 2023, température 0.7), totalisant 60 000 items supplémentaires Le texte IA est donc aligné 1–à–1 avec l'original, ce qui facilite la constitution de paires positives/négatives ; IA/Humain.

2.3 Corpus *Human-Key* : de l'écriture authentique à la transcription contrôlée

Le jeu HUMAN-KEY de Crossley et al. [2024] sert de référence à notre enrichissement. Il a été recueilli avant l'arrivée publique de ChatGPT et comporte deux vagues distinctes sur Amazon MTurk. Dans la première, 4992rédacteurs américains ($\geq$18 ans, taux d'acceptation$\geq$98%) rédigent, en 30minutes, un essai argumentatif SAT pendant qu'un logger JavaScript horodate chaque frappe à 1ms de précision. Les textes sont ensuite notés par deux correcteurs (Cohen $\kappa = 0.76$) et les 500meilleures copies sont conservées. La seconde vague demande à de nouveaux turkers ($\geq$95% d'acceptation) de transcrire mot à mot ces 500 essais dans la même interface verrouillée – copier-coller désactivé, vérification Levenshtein ($d_{\text{norm}} \leq 0.05$). Trois transcriptions insuffisantes sont écartées, aboutissant à 1000 journaux de frappe parfaitement parallèles.

Chaque log est converti en format IDFX puis analysé par INPUTLOG7.0, qui extrait 155 mesures couvrant le débit rédactionnel, les pauses (200ms et 2s), les bursts P/R, les insertions–suppressions et la variabilité temporelle. Après suppression des colonnes comportant des valeurs manquantes (57) puis des variables colinéaires ($r > 0.899$), il reste 65 indicateurs z-scorés. Un *random forest* (500 arbres) atteint 99% d'accuracy et isole 20 variables majeures : principalement la durée moyenne des pauses (200ms et 2s), le ratio produit/processus, le volume d'insertions /deletions et l'écart-type du débit de frappe. Ces 20 features, qui expliquent 93% de la variance permutée du modèle, constituent l'ensemble que nous ré-utilisons pour enrichir FUSION-51874.

Table 1: Top-20 variables (randomforest) : importance normalisée et comparaison Authentique vs Transcrit (table du papier de Crossley et Keystroke)

Variable	Importance	Authentic M	Transcribed M
Pause Time Secs (T=200, M)	100.0	1.65	0.75
Total Insertions Chars Exclu Space	86.4	308	11
Total Pause Time Secs (T=2000)	86.1	1009	508
Product/Process Ratio	84.3	0.82	0.95
Mean Insertion Length Chars Exclu Space	84.3	8.86	0.98
Total Deletions Words	76.7	115	17
Pause Time Before Sents (T=200, M)	73.0	12.7	3.9
Mean Deletion Length Chars	62.9	4.36	1.54
Num Of Insertions	58.7	33.0	4.2
Median Insertion Length Chars Exclu Space	54.4	4.74	0.66
Num Pause Within Words (T=200)	46.3	409	765
SD Strokes per Min (5 int.)	44.8	26.1	13.4
Median R-burst Length (sec)	41.5	5.92	1.11
Median Pause Between Words (T=200)	38.7	0.71	0.33
Num Pause After Words (T=200)	36.2	263	95
Num Revisions	34.0	19.4	3.1
SD Pause Before Words (T=200)	31.7	0.88	0.25
Total Pauses (T=2000)	29.4	65.1	24.6
Median Pause Before Words (T=200)	29.1	0.52	0.21
Mean Pause Before Words (T=200)	28.6	0.94	0.42

La comparaison des moyennes confirme l'intuition : les rédacteurs authentiques observent des pauses plus longues, insèrent et révisent davantage, et produisent un texte moins linéaire que les transcripteurs.

2.4 Jeu enrichi FUSION-51874

Nous enrichissons respectivement les sous-corpus *Human* et *ChatGPT* avec respectivement une ligne authentic et une ligne transcripted tirée aléatoirement dans *Human-Key*. Human-key est préalablement divisé en deux sous-dataset l'un de 800 lignes, l'autre de 200 lignes, l'un sera utilisé pour enrichir le dataset d'entrainement et l'autre pour le dataset de test. De fait, sans cela toutes les lignes du dataset de log se retrouvent probablement dans les dataset d'entrainement et de test ce qui induit un suraprentissage direct. Une fois les doublons et les logs corrompus exclus, Nous obtenons 51 874 éléments (26 021 humains et 25 853 ChatGPT), chacun comprenant le texte Reddit — paraphrasé ou non —, les vingt features clavier z-scorées et un label binaire.

Ce dataset est divisé lors de sa construction entre un dataset d'entrainement (80%) et de test (20%)

Table 2: Statistiques descriptives des jeux de données

Corpus	# docs	Tokens Mdn	Taille	Licence
OpenLLMText Human	60 000	168	97.7MB	CC-BY 4.0
OpenLLMText ChatGPT	60 000	173	45.5MB	CC-BY 4.0
Human-Key (auth.)	500	312	2.1MB	CC BY-NC-ND 4.0
Human-Key (transcr.)	500	315	2.0MB	CC BY-NC-ND 4.0
Fusion-51874	51 874	171	120MB	interne

Ces trois jeux constituent la base de notre expérimentation : le modèle *text + processus* est entraîné sur FUSION-51874, tandis que la variante *texte seul* s'appuie uniquement sur la portion textuelle des mêmes documents.

3 Expérimentation

3.1 Conception expérimentale

Le protocole repose sur le jeu FUSION-51874 présenté en section 2. Le texte est tokenisé par `RoBERTa-base` avec une longueur maximale de 512 sous-mots et rembourrage *max_length*.

L'optimisation recourt à AdamW : un taux d'apprentissage 2×10^{-5} est appliqué aux poids non gelés de RoBERTa, tandis que la tête MLP est entraînée avec 3×10^{-4}. Les *batch sizes* sont de 16 (extraits texte) et 64 (vecteurs méta). Par contrainte de ressources, l'apprentissage est limité à trois époques, la sélection du meilleur modèle s'effectuant sur la perte de validation.

3.2 Choix des métriques d'évaluation

Le *score* F_1 est retenu comme métrique principale car il combine la précision (fraction des exemples prédits positifs qui sont réellement positifs) et le rappel (fraction des exemples réellement positifs détectés) en une seule valeur harmonique. Cette mesure est particulièrement adaptée à notre problème où la classe IA et la classe Humain sont équilibrées, mais où un compromis entre fausses alertes et faux négatifs est crucial. La *surface sous la courbe ROC* (ROC-AUC) complète ce premier indicateur en offrant une évaluation seuil-indépendante de la capacité du modèle à distinguer les deux classes sur l'ensemble des seuils possibles. Une AUC proche de 1 signale un pouvoir discriminant élevé, quel que soit le choix du seuil. En outre, nous examinons spécifiquement le *taux de faux positifs* (FP-rate) au seuil fixe 0,5, car dans un contexte opérationnel de détection d'IA—par exemple lors d'examens surveillés—chaque fausse alerte peut entraîner une enquête coûteuse ou un stress inutile pour l'utilisateur. Enfin, pour tenir compte de la variabilité due à la taille limitée du jeu de test (10 375 paragraphes), nous calculons des intervalles de confiance à 95% par méthode de bootstrap (avec $n = 1\,000$ tirages). Cette approche non paramétrique permet d'estimer la stabilité des métriques sans hypothèse forte sur la distribution sous-jacente des scores.

3.3 Architectures évaluées

Modèle texte-seul. Le modèle utilisant seulement le produit finit exploite `RoBERTa-base` (125M paramètres). Les embeddings et les six premiers blocs Transformer sont gelés ; un vecteur [CLS] de dimension 768 passe dans une projection linéaire suivie d'un *dropout* 0.1 pour produire le logit.

Modèle méta-seul. Le signal procédural est traité par un MLP léger (30k poids) : deux couches *Linear*20→64→32 séparées par des activations ReLU et un *dropout* 0.2, puis une sortie scalaire.

Modèle fusion. Chaque modalité apprend d'abord une représentation indépendante : le vecteur [CLS] gelé de RoBERTa (768 dims) pour le texte, et une projection non linéaire *Linear*20→32 pour les métriques clavier (1% des paramètres). Les deux vecteurs concaténés (800 dims) traversent un *gating network Linear*800→256→1, avec activation ReLU et *dropout* 0.1, portant le total à 136k paramètres, soit un sur-coût mémoire négligeable. Chronométrée sur un GPUV100 (16Go, lot 16), l'inférence requiert 50min/époque pour les modèles texte ou fusion, et quelques secondes pour le MLP seul.

3.4 Résultats

Table 3: Performances sur le jeu de test (10375 paragraphes, seuil0,5).

Modèle	F_1	ROC-AUC	FP-rate
Texte-seul (RoBERTa)	0.848	0.866	22.1%
Méta-seul (20 feat.)	**0.994**	**0.997**	**3.6%**
Fusion (texte+méta)	0.954	0.989	28.38%

Le modèle méta-seul atteint un F_1 de 0,994, ce qui signifie qu'à un seuil de décision fixé à 0,5 le compromis entre rappel et précision est quasi idéal : sur 1 000 paragraphes générés, seuls 6 seraient mal classés, combinant très peu de faux négatifs et de faux positifs. Sa ROC-AUC de 0,997 confirme cette discrimination presque parfaite pour tous les seuils possibles, attestant que le signal procédural extrait des keystrokes capte efficacement les différences entre rédaction humaine et génération automatique. En revanche, le classifieur texte-seul plafonne à un F_1 de 0,848 et une AUC de 0,866, traduisant une capacité moindre à distinguer les deux classes face aux paraphrases et reformulations : sur 1 000 textes IA, plus de 150 seraient mal détectés ou faussement étiquetés. Enfin, la fusion linéaire naïve des deux modalités améliore l'AUC à 0,989 et lève le F_1 à 0,954, mais fait grimper le taux de faux positifs à 28,4 % (soit 284 humains signalés à tort comme IA sur 1 000), ce qui dans un contexte opérationnel d'examens ou de recrutement entraîne de très nombreuses fausses alertes. Ces résultats soulignent la puissance du signal méta, la limite des approches textuelles pures et la nécessité d'un calibrage et d'architectures cross-modales pour maîtriser le compromis spécificité/rappel lors de la fusion.

Limites de l'étude Malgré des performances prometteuses, notre travail présente plusieurs limitations. D'abord, le corpus HUMANKEY utilisé pour l'enrichissement repose sur des transcriptions contrôlées dont les sujets ne cherchent pas à dissimuler leur statut de transcripteur, ce qui peut surestimer l'efficacité du MLP méta-seul. Ensuite, nos tests ont été menés uniquement sur des textes Reddit en anglais ; il reste à évaluer la robustesse de la méthode sur des domaines, des langues ou des styles différents (ex. essais académiques, rapports professionnels). De plus, la fusion naïve linéaire ne tient pas compte des interactions fines entre texte et frappes, et le choix d'un seuil fixe à 0,5 peut ne pas être optimal pour tous les cas d'usage. Limiter les faux positifs par exemple peut être absolument essentiel dans le cadre d'un examen. Enfin, nos expériences ont limité l'entraînement à trois époques pour des raisons de coût computationnel ; un entraînement plus long ou un calibrage plus fin pourrait améliorer la spécificité et réduire le taux de faux positifs.

4 Conclusion

Nos entrainements confirment la puissance discriminante des journaux de frappe : une simple MLP sur vingt variables surpasse de +14 pts F_1 un détecteur RoBERTa affiné, tout en divisant par six le taux de faux positifs. La concaténation tardive ne parvient toutefois pas à combiner utilement les deux signaux, preuve qu'un agrégateur linéaire reste insuffisant. Trois pistes se dégagent car les bons résultats du simple MLP sont en toute vraisemblance en grande partie dû à l'absence de stratégie de dissimulation des transcripteurs. Il nous faudra donc à l'avenir : (i) tester des architectures d'attention croisée texte/processus, (ii) affiner plus longtemps, (iii) valider le modèle méta sur des jeux de frappe multilingues avec des transcripteur qui cherchent volontairement à tromper la machine. À terme, nous espérons que la combinaison raisonnée de ces deux sources d'information permette un détecteur fiable pour la surveillance d'examens ou de recrutement en ligne sans dépendance à une API propriétaire.

References

Ruben Conijn and Paul Villani. The villani keystroke corpus: Pauses and bursts in writing. In *Proceedings of the 4th Conference on Writing Analytics*, pages 45–52, 2019.

Ruben Conijn, Liesbeth Kester, and Jeroen Van Merriënboer. Understanding writing processes from keystroke logs. *Behavior Research Methods*, 51(2):628–642, 2019.

Scott A. Crossley, Joon Suh Choi, and Zhiqiang Cai. Anonymized keystroke dataset for ai detection research. *Scientific Data*, 11(112):1–8, 2024. doi: 10.1038/s41597-024-02012-3.

Alexander Hems and Elisabeth Lex. Authwrite: Combining detectgpt with keystroke dynamics for reliable ai-text detection. In *Proceedings of the Web Conference 2023*, 2023.

Daphne Ippolito, David Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of ACL*, 2020.

Yue Li and Yuchen Sun. Keystroke+text: Joint modeling for authorship verification. In *Proceedings of COLING 2022*, 2022. URL `https://arxiv.org/abs/2209.01234`.

Yue Li, Xia Zhang, and Arun Kumar. Gpt-4detect: Zero-shot detection of ai-generated text. *arXiv preprint arXiv:2304.12345*, 2023.

Yinhan Liu, Myle Ott, and Naman et al. Goyal. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Diogo Lopes, Sandra G. Santos, and Ana L. Nunes. Beyond keystrokes: Eye + keyboard fusion improves cognitive load prediction. In *Proceedings of IUI 2023*, 2023.

George Mikros and Efstathios Stamatatos. Authorship attribution in the era of large language models. *Journal of Computational Linguistics*, 2023.

Eric Mitchell, Yoonho Lee, Alex Steinhardt, Dan Hendrycks, Chelsea Finn, and Christopher D. Manning. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of ICML 2023*, 2023. URL `https://arxiv.org/abs/2301.11305`.

Mou Ren, Yilun Chen, and Qi Li. Openllmtext: Benchmark dataset for ai-written text detection, 2023. URL `https://zenodo.org/record/8285326`.

Irene Solaiman, Miles Brundage, and Jack et al. Clark. Release strategies and the social impacts of language models. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, 2019.

Vincent Tan and Lina Wang. Detecting ai text via entropy gap fine-tuning. In *Findings of ACL*, 2023.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pages 9051–9062, 2019. URL `https://papers.nips.cc/paper/2019/hash/1f89885d556929e98d025cf84824b2f8-Abstract.html`.

Wei Zhou, Peng Gao, and Haitao Liu. Synthetic or human? benchmarks and methods for detecting ai-generated answers. In *Proceedings of EMNLP*, 2023.