La détection de textes générés par des modèles de langue : une tâche complexe? Une étude sur des textes académiques

Vijini Liyanage, Davide Buscaldi

Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

La détection de textes générés par des modèles de langue : une tâche complexe? Une étude sur des textes académiques

Vijini Liyanage Davide Buscaldi
Le Laboratoire d'Informatique de Paris-Nord (LIPN), France
liyanage@lipn.univ-paris13.fr , davide.buscaldi@lipn.univ-paris13.fr

RÉSUMÉ

L'émergence de modèles de langue très puissants tels que GPT-3 a sensibilisé les chercheurs à la problématique de la détection de textes académiques générés automatiquement, principalement dans un souci de prévention de plagiat. Plusieurs études ont montré que les modèles de détection actuels ont une précision élevée, en donnant l'impression est résolue. Cependant, nous avons observé que les ensembles de données utilisés pour ces expériences contiennent des textes générés automatiquement à partir de modèles pré-entraînés. Une utilisation plus réaliste des modèles de langue consisterait à effectuer un affinage sur un texte écrit par un humain pour compléter les parties manquantes. Ainsi, nous avons constitué un corpus de textes générés de manière plus réaliste et mené des expériences avec plusieurs modèles de classification. Nos résultats montrent que lorsque les ensembles de données sont générés de manière réaliste pour simuler l'utilisation de modèles de langue par les chercheurs, la détection de ces textes devient une tâche assez difficile.

ABSTRACT

How difficult is it to detect LLM-generated text ? A study on academic text

Prevalence of competent models such as GPT-3 has fostered many researches to focus on the detection of automatically generated academic text. Several studies have shown that current detection models have high accuracy, which may give the impression that the task is solved. However, we observed that the datasets used for these experiments often contain automatically generated texts from pre-trained models. A more realistic use of language models would be to fine-tune them on an original text to complete its missing parts (e.g. abstract or conclusions). Therefore, we built a corpus of texts generated in a more realistic way and conducted experiments with several classification models. Our results show that when datasets are generated realistically to simulate the use of language models by researchers, detecting these texts becomes more challenging.

MOTS-CLÉS : Modèles de langue, Génération automatique de texte, Détection, Classification.

KEYWORDS: LLMs, Automatic text generation, Detection, Classification.

1 Introduction

Depuis l'introduction de grands modèles de langue (ML) pour la génération de texte, les chercheurs ont essayé de déterminer si le texte généré peut être détecté et avec quelle efficacité. Par exemple, Giant Language Testing Room (GLTR) (Gehrmann *et al.*, 2019) est un modèle de visualisation qui aide les humains à détecter le texte généré artificiellement, en se basant sur le principe qu'une prévalence de tokens très probables (selon un modèle de langue pre-entrainé - dans leur cas GPT-

2) est un indicateur. (Rodriguez *et al.*, 2022) ont étudié la détectabilité de documents techniques falsifiés par des modèles de classification basés sur BERT, obtenant des précisions allant de 86 à 95%. DetectGPT (Mitchell *et al.*, 2023) évalue l'effet de perturbations aléatoires du texte sur les probabilités de génération des tokens par un modèle GPT, de façon similaire a GLTR. Ils affichent une AUROC de 0.95.

Récemment, certaines tâches partagées ont été proposées sur sujet de la détection de texte academique généré automatiquement. DAGPap22 a été accueilli dans le cadre du Third Workshop on Scholarly Document Processing (Cohan *et al.*, 2022). Les organisateurs ont rassemblé des articles originaux sur MICPRO (microprocesseurs et microsystèmes) et les objectifs de développement durable des Nations unies, et ont créé des résumés artificiels en utilisant diverses méthodes (modèles de résumé et modèles de type GPT). La tâche s'est avérée plus facile à résoudre que prévu, le meilleur modèle étant basé sur trois variantes de BERT : SciBERT(Beltagy *et al.*, 2019), RoBERTa(Liu *et al.*, 2019) et DeBERTa(He *et al.*, 2020). En utilisant un ensemble de ces trois modèles, il a obtenu un score F1 maximal de 99,24% (Glazkova & Glazkov, 2022). SynSciPass (Rosati, 2022) utilise le modèle SciBERT pour la détection et ont obtenu une précision de 98,3% pour DAGPap22.

Toutefois, ces excellents résultats peuvent être trompeurs. En effet, si l'on regarde plus attentivement aux jeux de données utilisés, nous trouverons que les sous-ensembles de données générés sont souvent assemblés en utilisant des ML pre-entrainés. Par exemple, (Mitchell *et al.*, 2023) utilisent 500 articles d'un jeu de donnés d'actualités pour les échantillons originaux, et 500 articles générés automatiquement par 4 différents ML avec les premiers 30 mots de chaque article original. Le jeu de données WikiGPT [1] a été créé avec le prompt "Introduction de 200 mots de style wikipedia sur {titre} {texte_intro}" où *titre* est le titre de la page wikipedia, et *texte_intro* est constitué par les sept premiers mots de l'introduction Wikipedia originale.

2 Un jeu de données plus réaliste

Nous avons commencé à construire un corpus avec l'objectif de simuler au mieux la façon dont un auteur humain utiliserait les LMs pour ses travaux : par exemple, pour combler les trous dans certaines parties de l'article ou pour compléter un résumé en donnant le corps de l'article. Après, notre objectif c'était aussi celui de vérifier si la détectabilité des textes générés automatiquement restait élévée aussi dans ce cas. Nous avons donc assemblé un jeu de données composé des sections suivantes :

- Un ensemble de données composé d'articles générés par un réglage fin du modèle GPT-2 avec un paramètre de température de 0,7. **Entièrement généré GPT-2 affiné (t = 0.7) (D1)** - 200 articles, 1250 mots par article en moyenne ;
- Un ensemble de données composé de résumés qui contiennent à la fois du contenu généré par une machine (en utilisant GPT-2 affiné sur ArXiv) et du contenu écrit par un humain. **Abstraites hybride ArXiv-NLP pré-entraîné (t = 0.7) (D2)** - 200 articles, 150 mots par article en moyenne ;
- Le jeu de données du concours DAGPap22 [2]. **DAGPap22(D3)** - 5350 articles, 160 mots par article en moyenne ;
- Un ensemble de données composé d'articles générés par un réglage fin du modèle GPT-2 avec un paramètre de température de 0,9. **Entièrement généré GPT-2 affiné (t = 0.9)(D4)** - 200

1. https://huggingface.co/datasets/aadityaubhat/GPT-wiki-intro
2. https://www.kaggle.com/competitions/detecting-generated-scientific-papers

articles, 1250 mots par article en moyenne ;

— Un jeu de données de résumés obtenu en exploitant le modèle GPT-2 pré-entraîné, sans réglage fin. **Abstraites GPT-2 pré-entraîné (t = 0.7) (D5)** - 200 articles, 125 mots par article en moyenne ;

Pour construire les ensembles de données D1 et D4, nous avons effectué un réglage d'un modèle GPT-2 sur les articles originaux. Pour s'assurer qu'il n'y a pas de chevauchement entre les données d'entraînement de GPT-2 et nos données, nous avons sélectionné pour notre jeu de données uniquement des articles récents (à partir de 2022). Afin d'obtenir des résultats plus fidèles à partir du modèle pré-entraîné, nous avons limité notre jeu de données au même domaine (NLP) et à la même source (ArXiv). Pour chaque article original, nous envoyons un prompt de 50 mots provenant de l'article lui-même au modèle afin de produire un nouvel article. La génération continue jusqu'à ce que la longueur du nouvel article soit similaire à la longueur de l'article original, afin d'éviter tout biais de longueur lors de la classification. La différence entre D1 et D4 réside uniquement dans les températures utilisées pour les générateurs, respectivement $0,7$ et $0,9$. Le modèle GPT-2 de base était la version avec 12 couches, 12 têtes d'attention et 124M de paramètres. Le modèle a été affiné par un seul article original à la fois et une moyenne de 45 minutes GPU a été consommée pour chaque article (avec un GPU Nvidia T4). Ainsi, pour l'ensemble du jeu de données, environ 75 heures de GPU ont été allouées.

Le jeu de données D2 a été créé avec une intervention humaine, en utilisant l'interface 'write with transformer' sur Huggingface [3]. Les auteurs, agissant en tant qu'experts du domaine, ont supprimé la partie conclusive des résumés, et les ont complétés avec des phrases proposées par le modèle GPT-2 mis au point sur ArXiv-NLP. Nous avons choisi l'une des trois meilleures complétions proposées par le modèle. Ce scénario est équivalent au scénario de 'tampering' proposé dans (Rodriguez *et al.*, 2022), avec des curateurs manuels.

Finalement, le jeu de données D5 réprésente le cas où le modèle est utilisé sans réglage, avec uniquement un prompt de 50 mots de l'article original. Comme on peut apprécier en Table 1, DetectGPT n'est pas capable de détecter correctement le fait que dans D1 et D2 le contenu des articles a été manipulé par GPT-2 (en Figure 1 c'est aussi possible de voir la sortie de GLTR pour un exemple dans les différents datasets, même si nous n'avons pas calculé les résultats de GLTR sur tous les exemples). Nous n'avons pas le résultat final pour D4 pour des contraintes de temps, mais le corpus étant très similaire à D1, nous attendons des résultats comparables.

Modèle	**D1**	**D2**	**D5**
Z-score	-0,351	0,239	0,911

TABLE 1 – Z-scores moyens obtenus par DetectGPT pour les jeux des données D1, D2 et D5. Z-scores supérieurs à 0.7 indiquent que DetectGPT considère les textes synthétiques.

3 Expériences

Pour comprendre le degré de similarité du texte généré avec l'original et donc estimer la difficulté inhérente des jeux de données, nous avons calculé les scores BLEU (Bilingual Evaluation Understudy) (Papineni *et al.*, 2002) et ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) (Tableau 2). Lorsque le paramètre de température d'un modèle de génération est augmenté, le caractère aléatoire du contenu généré est plus élevé. Par conséquent, la similarité n-gram entre le contenu généré et le contenu de référence (original) correspondant devient faible. Ainsi, comme le montre le

3. https://transformer.huggingface.co/

Tableau 2, BLEU et ROUGE, qui sont des scores basés sur les n-grammes, sont plus faibles pour le D4 plus “chaud" que les scores produits par le jeu de données D1 plus “froid".

Les scores BLEU et ROUGE du jeu de données D5, généré à l’aide du modèle GPT-2 sans réglage, sont très faibles par rapport aux corpus générés à l’aide du modèle GPT-2 affiné. Il s’agit d’un comportement attendu puisque lorsqu’un modèle n’est pas affiné, le contenu généré ne peut pas être très similaire au contenu original, ce qui rend la similarité n-grammes très faible.

Méthode de construction de la base de données	**UGL-BLEU**	**S-BLEU**	**Rouge-1**	**Rouge-2**	**Rouge-L**
(D1)GPT-2 affiné (t = 0.7)	0.867	0.809	0.853	0.810	0.853
(D4)GPT-2 affiné (t = 0.9)	0.858	0.766	0.830	0.772	0.834
(D5)GPT-2 pré-entraîné (t = 0.7)	0.467	0.356	0.549	0.431	0.533
(D3) ArXiv-NLP pré-entraîné (t = 0.7)	0.824	0.792	0.882	0.840	0.881

TABLE 2 – Scores moyens BLEU et ROUGE(Recall) , UGL-BLEU : Niveau d’unigramme BLEU, S-BLEU : Phrase BLEU

Pour la classification des textes en généré ou original, nous avons exploité plusières variantes de BERT telles que SciBERT, RoBERTa, DeBERTa and ELECTRA (Clark *et al.*, 2020). Le tableau 5 en annexe fournit les paramètres de chaque variante. Tous les jeux de données ont été divisés aléatoirement en 80 : 20 pour les tests.

Calculé pour chaque modèle le score F1 (voir Table 3). Tous ces scores sont la moyenne de trois expériences avec une repartition aléatoire des jeux d’entrainement et test. En général, les scores sont élevés pour les jeux de données D3 et D5, ce qui prouve qu’ils sont plus faciles à détecter. D3 n’utilise pas de méthodes neuronales pour la partie générée, tandis que D5 est généré sans réglage. D2 est un jeu de données altéré, ce qui signifie que le texte synthétique est produit en incluant de légères modifications au contenu original. La détection est donc difficile, comme le prouvent les scores de classification relativement faibles pour ce jeu de données.

Modèle	**D1**	**D2**	**D3**	**D4**	**D5**
BERT	33.33	56.04	53.29	60.11	89.90
SciBERT	84.65	52.23	95.02	79.16	94.99
RoBERTa	67.83	55.23	96.87	33.33	84.85
DeBERTa	67.03	48.85	97.17	48.13	95.00
Electra$_{small}$	60.11	51.00	93.95	56.36	92.50
Electra$_{base}$	56.16	41.30	96.64	48.13	95.00

TABLE 3 – F-1 scores obtenus par les variantes BERT

4 Conclusions

Cette étude a examiné la détection de textes synthétiques générés à l’aide de modèles de langue, en se concentrant sur les modèles entraînés sur des articles académiques. Les résultats ont montré que lorsque les modèles sont affinés à l’aide d’articles originaux, la détection des textes synthétiques devient plus difficile, car le texte généré peut contenir des éléments de l’article original. En revanche, textes générés par des modèles non affinés sont plus faciles à détecter. Les textes modifiés manuellement représentent un défi important pour les modèles, mais les résultats peuvent varier en fonction de

l'architecture du classificateur ou de la construction de l'ensemble de données. L'étude a également examiné l'effet de l'ajustement de la température du modèle de génération sur la détection, mais les résultats sont mitigés et nécessitent des recherches supplémentaires.

Références

BELTAGY I., LO K. & COHAN A. (2019). SciBERT : A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 3615–3620, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1371.

CLARK K., LUONG M., LE Q. V. & MANNING C. D. (2020). ELECTRA : pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* : OpenReview.net.

COHAN A., FEIGENBLAT G., FREITAG D., GHOSAL T., HERRMANNOVA D., KNOTH P., LO K., MAYR P., SHMUELI-SCHEUER M., DE WAARD A. & WANG L. L. (2022). Overview of the third workshop on scholarly document processing. In *Proceedings of the Third Workshop on Scholarly Document Processing*, p. 1–6, Gyeongju, Republic of Korea : Association for Computational Linguistics.

GEHRMANN S., STROBELT H. & RUSH A. (2019). GLTR : Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics : System Demonstrations*, p. 111–116, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-3019.

GLAZKOVA A. & GLAZKOV M. (2022). Detecting generated scientific papers using an ensemble of transformer models. In *Proceedings of the Third Workshop on Scholarly Document Processing*, p. 223–228, Gyeongju, Republic of Korea : Association for Computational Linguistics.

HE P., LIU X., GAO J. & CHEN W. (2020). Deberta : Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv :2006.03654*.

LIN C.-Y. (2004). Rouge : A package for automatic evaluation of summaries. In *Text summarization branches out*, p. 74–81.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). Roberta : A robustly optimized bert pretraining approach. *arXiv preprint arXiv :1907.11692*.

MITCHELL E., LEE Y., KHAZATSKY A., MANNING C. D. & FINN C. (2023). Detectgpt : Zero-shot machine-generated text detection using probability curvature.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, p. 311–318.

RODRIGUEZ J., HAY T., GROS D., SHAMSI Z. & SRINIVASAN R. (2022). Cross-domain detection of GPT-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1213–1233, Seattle, United States : Association for Computational Linguistics. DOI : 10.18653/v1/2022.naacl-main.88.

ROSATI D. (2022). SynSciPass : detecting appropriate uses of scientific text generation. In *Proceedings of the Third Workshop on Scholarly Document Processing*, p. 214–222, Gyeongju, Republic of Korea : Association for Computational Linguistics.

Annexe

Origine	Extrait
Document original	Moreover, when temporal word embeddings provide words' representations that are more inclined towards their literal core meaning (and not the metaphorical one), models exploiting end up correctly identifying metaphors more easily. **Furthermore, embeddings of words belonging to some language domains in specific time periods can be biased towards their metaphorical meaning, leading to words being used in metaphorical contexts way more than in literal ones. This would prevent neural models from correctly identifying the words as metaphors.**
D1	Moreover, when temporal word embeddings provide words' representations that are more inclined towards their literal core meaning (and not the metaphorical one), models exploiting end up correctly identifying metaphors more easily. **One way to investigate these hypotheses further is to explore the nearest neighbors of a word in the word embeddings used in a figurative way inside a sentence, both in a static (atemporal) word embedding space, e.g., obtained with GloVe [Jeffrey Pennington, 2014] and in a decade-specific temporal space, e.g., obtained from the CoHa1 corpus with Procrustes [Edouard Grave, 2018].**
D4	Moreover, when temporal word embeddings provide words' representations that are more inclined towards their literal core meaning (and not the metaphorical one), models exploiting end up correctly identifying metaphors more easily. **One way to investigate these hypotheses further is to explore the nearest neighbors of a word in the word embeddings used in a figurative way inside a sentence, both in a static (atemporal) word embedding, i.e., a decade, time period, and/or a temporal (osfringes) word embedding, i.e., a decade, time period.**
D5	Moreover, when temporal word embeddings provide words' representations that are more inclined towards their literal core meaning (and not the metaphorical one), models exploiting end up correctly identifying metaphors more easily. **For example, if a model is able to identify a metaphor that is more likely to be associated with the literal meaning of a word, then it will be more accurate in identifying the metaphor in the context of the word itself. In this paper, we present a novel approach to the identification of metaphors in language.**
D2	Moreover, when temporal word embeddings provide words' representations that are more inclined towards their literal core meaning (and not the metaphorical one), models exploiting end up correctly identifying metaphors more easily. **In contrast to sentence embeddings, the method of segmentation also exploits the concept of end-to-end (or equivalence) for metaphors and the corresponding temporal word embeddings. In this work, we conduct a multidisciplinary investigation of a well-known technique for identifying and segmenting metaphors.**

TABLE 4 – Extraits d'un article et ses versions générées à partir des différentes sections de l'ensemble de données (non applicable pour D3).

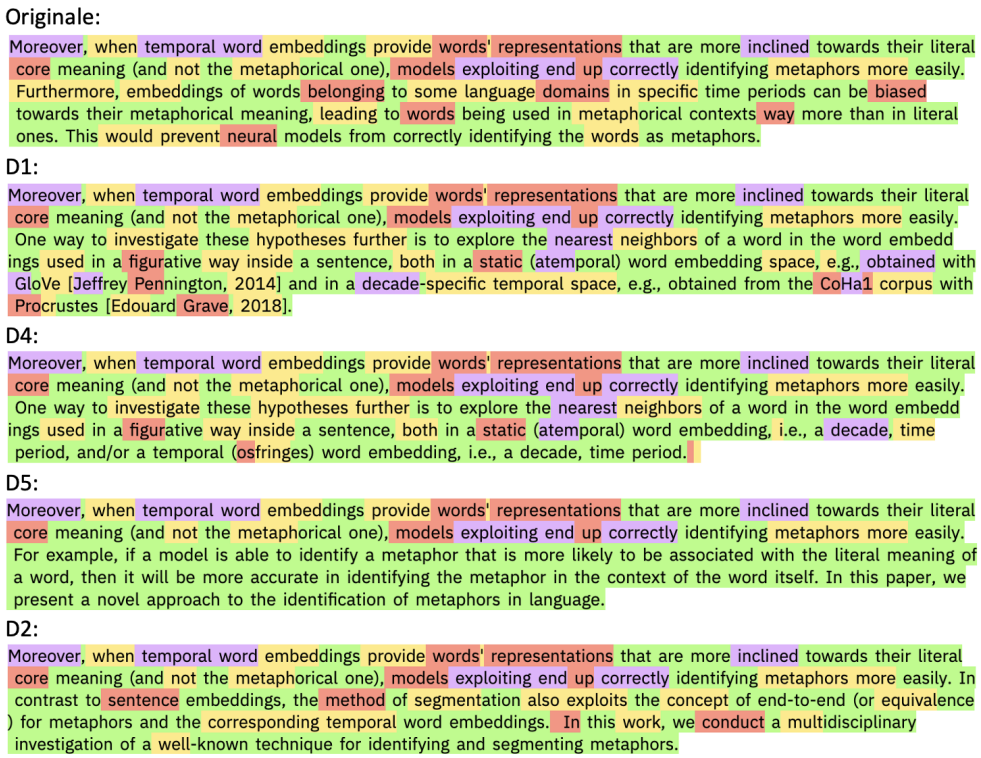

FIGURE 1 – Sortie du modèle GLTR pour les exemples du tableau 4. Vert : le token est dans le top 10 des tokens les plus probables pour le LM. Jaune : token dans le top 100. Rouge : token dans le top 1000. Violet : le token n'apparaît pas dans les 1000 meilleures options pour le LM.

Modèle	**Vocab (K)**	**Taille cachée**	**Couches**	**Taille du lot**	**Époques**	**Paramètres(M)**
BERT_{base}	30	762	12	64	3	110
DistilBERT	30	768	6	16	3	66
SciBERT_{base}	30	768	12	16	3	110
RoBERTa_{large}	50	1024	16	16	3	355
DeBERTa_{large}	50	1024	24	16	3	350
Electra_{small}	30	256	12	16	3	14
Electra_{base}	30	768	12	16	3	110
XLNet_{base}	32	768	12	16	3	110

TABLE 5 – Hyper Paramètres des variantes de BERT considérées