https://doi.org/10.31891/2307-5732-2025-351-61
УДК 004.773.2

ТАРАСОВ АНДРІЙ
Національний університет «Львівська політехніка»
https://orcid.org/0009-0006-9925-1847
e-mail: andrii.tarasov.kn.2021@lpnu.ua
ШАХОВСЬКА НАТАЛІЯ
Національний університет «Львівська політехніка»
https://orcid.org/0000-0002-6875-8534
e-mail: Nataliya.b.shakhovska@lpnu.ua

ВИЯВЛЕННЯ МАШИННО ЗГЕНЕРОВАНОГО ТЕКСТУ ЗА ЙОГО СТАТИСТИЧНИМИ ВЛАСТИВОСТЯМИ

У статті розглянуто модифікацію методу виявлення машинно згенерованого тексту Fast-DetectGPT. На відміну від оригінального методу, що використовує логарифмічні імовірності всіх токенів у словнику для класифікації тексту, запропонований метод використовує лише ймовірності 20 найбільш імовірних токенів. Таким чином зменшується кількість даних, які необхідно опрацювати для класифікації: 20 токенів замість 50 000 для кожного токена вхідного тексту. Необхідні 20 токенів можна отримати через запит до хмарних сервісів, що надають доступ до великих мовних моделей, а не запускати модель локально на пристрої. Запропонований метод використовує статистичні параметри тексту та імовірних токенів, тож результат класифікації можна пояснити. Це важливо у випадку встановлення автора тексту. Класифікація текстів відбувається на основі певних характеристик тексту, таких як середня імовірність токенів у тексті, середня імовірність можливих токенів у словнику під час генерації, perplexity та d_score. D_score це метрика з методу Fast-DetectGPT.

Ключові слова: машинно згенерований текст, велика мовна модель, Fast-DetectGPT, логарифмічна імовірність.

ANDRII TARASOV, SHAKHOVSKA NATALIYA
Lviv Polytechnic National University

DETECTING MACHINE-GENERATED TEXT BY ITS STATISTICAL PROPERTIES

The article considers the modification of the machine-generated text detection method Fast-DetectGPT. Unlike the original method, which uses the logarithmic probabilities of all tokens in the dictionary to classify text, the proposed method uses only the probabilities of the 20 most probable tokens. This reduces the amount of data that needs to be processed for classification: 20 tokens instead of 50,000 for each token of the input text. The required 20 tokens can be obtained by querying cloud services that provide access to large language models, rather than running the model locally on the device. The proposed method uses statistical parameters of the text and probable tokens, so the classification result can be explained. This is important in the case of establishing the author of the text. To detect machine-generated text, you can use not only the d_score metric, but also perplexity, prompt_logprobs_mean, and vocab_samples_mean. Also, to calculate d_score and vocab_samples_mean, you can use the logarithmic probabilities of only the 20 most likely tokens, rather than the entire dictionary. In the case of the facebook/opt125m and gpt2 models, such classification by the vocab_samples_mean value allows you to get even better accuracy than using the entire dictionary. When using the d_score value, the results for these two models are opposite - more accurate classification occurs based on the entire dictionary. The accuracy of macro f1 classification by prompt_mu_score and perplexity is not less than 0.97 in all experiments, but these values do not depend on the probabilities of possible tokens. The accuracy of classification by prompt_logprobs_mean is high even when using only the 20 most likely tokens instead of the entire dictionary. This approach allows the use of cloud services for access to large language models and commercial models, and does not require the deployment of a local model.

Keywords: machine-generated text, large language model, Fast-DetectGPT, logarithmic probability

Стаття надійшла до редакції / Received 07.04.2025
Прийнята до друку / Accepted 20.04.2025

Вступ

За допомогою сучасних великих мовних моделей (Large language model - LLM) можна згенерувати текст різного стилю на різні теми. Такий інструмент має багато застосувань, зокрема це значно спрощує переклад та реферування текстів, створення контенту для вебсайтів і соціальних мереж тощо. Проте зі збільшенням кількості машинно згенерованого тексту (МГТ) з'являється необхідність відрізняти МГТ від того, який написала людина. Важливо мати можливість виявити МГТ у статтях та новинах, серед постів та коментарів у соціальних мережах. Наприклад, текст статті, згенерований мовною моделлю може містити помилки. Читачі, які не знають про це, можуть повірити наданій інформації, оскільки вважають, що автор-журналіст написав матеріал самостійно та перевірив достовірність [1, 2].

Іншим прикладом є цілеспрямоване поширення дезінформації чи маніпуляцій. Звичайно, тексти з дезінформацією чи помилками може написати і людина, проте використання LLM дозволяє автоматизувати, а отже спростити та значно прискорити процес створення контенту на задану тему, або використовуючи певний наратив. Модель не перевірятиме достовірність інформації та логічність висновків, а лише згенерує відповідь, яка найкраще відповідає запиту користувача. Тож мовні моделі можуть бути використані у ботофермах для імітації діалогу або просто створення коментарів та публікацій. У будь-якому разі, читання тексту, створеного мовною моделю потребує більше уваги, щоб виявити можливі проблеми, пов'язані з МГТ. Таким чином вдасться уникнути проблем через

споживання інформації що містить помилки, є неточною, не перевіреною або спеціально маніпулятивною. Тож попит на інструменти для автоматизованого виявлення МГТ зростатиме.

Аналіз літературних джерел

У статті [3] автори досліджують можливість виявлення машинно згенерованого тексту з різних наукових галузей за допомогою моделі, яка спеціалізована до конкретної дисципліни. Зокрема вказано, що експерти у певній предметній галузі (subject matter expert, SME) можуть дуже точно відрізняти технічний МГТ, від написаного людиною. Водночас це важко зробити для оцінювачів, які не є фахівцями у цьому домені. Тому, враховуючи легкість, з якою можна створити великі обсяги штучно створеного тексту за допомогою сучасних великих мовних моделей, необхідні автоматизовані засоби його виявлення, щоб полегшити навантаження на SME та на читачів.

Один з підходів, за яким можна визначити авторство тексту це стильометрія. Для цього необхідно порівняти статистичні параметри текстів, що точно належать автору, та текстів, які нобхідно перевірити. Оскільки кожному автору притаманний свій набір часто вживаних слів, фраз та інші особливості написання тексту, таким чином можна визначити чи текст написаний автором.

Подібний підхід можна застосувати і для текстів, створених за допомогою мовних моделей, оскільки під час створення тексту вони використовують імовірності токенів, і кожна модель має власні (ваги) значення, що характеризують її авторський профіль. За дослідженням [4] машинно створений текст має 80% маси ймовірності в 500 найпоширеніших словах, а людський текст містить рідковживані слова. Також згадано такий недоліки великих мовних моделей як зменшення узгодженості зі збільшенням обсягу тексту. Це вказує на можливість виявлення МГТ за допомогою аналізу стилю написаного тексту та порівнянням зі стилем потенційного автора.

Виявлення машинно згенерованого тексту за допомогою аналізу імовірностей токенів використане у методі DetectGPT [2] та Fast-DetectGPT [1]. У DetectGPT за допомогою допоміжної LLM створюються варіації тексту, щоб визначити чи текст згенерований певною мовною моделлю, чи ні. Для класифікації потрібно порівняти імовірність створення цільовою моделлю початкового тексту з імовірностями створення згенерованих версій [2]. Такий підхід, використаний, наприклад, у DetectGPT, вимагає багато обчислень для створення перефразованих текстів та оцінювання кожного з них потім.

Алгоритм Fast-DetectGPT [1] є оптимізацією методу DetectGP. Класифікація відбувається після порівняння імовірностей токенів тексту та імовірностей токенів у словнику для кожної позиції у вхідному тексті. Цей метод передбачає лише 1 виклик мовної моделі для визначення імовірностей токенів тексту.

Методи та моделі

Опис наборів даних

Використані дані – це близько 2000 текстів з набору даних kaggle LLM Detect AI Generated Text Dataset [5]. Кількість текстів згенерованих LLM та написаних людьми однакова, а тексти двох класів подібні за обсягом (рис. 1). Частина записів у наборі даних є некоректними, а певні тексти завеликі для котексту моделі gpt2 [7]. Тому для експериментів обрано майже 1600 текстів, які мають довжину більше 15 речень та можуть бути опрацьовані усіма моделями. Інформація про те, якою моделлю було згенеровано набір даних відсутня, тому в контексті виявлення МГТ це випадок чорної скриньки.

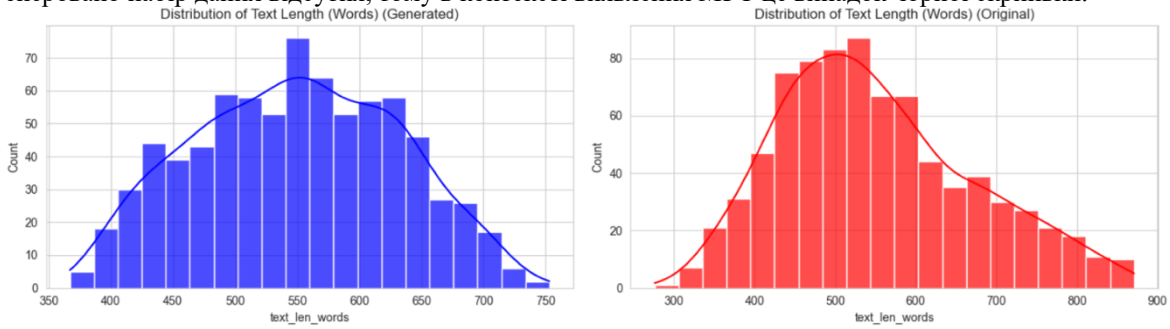

Рис.1. Статистичні параметри текстів з набору даних

Запропонований метод виявлення машинно згенерованого тексту є оптимізацією Fast-DetectGPT [1] та передбачає використання лише 20 найімовірніших токенів зі словника. На основі множини зразків з можливих токенів та ймовірностей токенів, які формують вхідних текст, обчислені такі значення:

- *D_score* - основна метрика методу Fast-DetectGPT,
- *Perplexity* - міра невпевненості моделі у передбаченому результаті,
- *Vocab_logprobs_mean* - середня імовірність можливих токенів,
- *Prompt_logprobs_mean* - середня імовірність можливих токенів.

Ці метрики обчислені для кожного тексту, а *d_score* та *vocab_logprobs_mean* двома способами: з використанням всього словника та лише 20 токенів. Для експериментів використано моделі з малою кількістю параметрів (до 0,5 млрд) Opt-125m[6], Gpt2[7], SmolLM2-360M[8], Qwen2.5-0.5B[9]. А також Llama-3.2-1B[10], Gemma-2-2b[11] та Open_llama_7b_v2[12]. Далі наведено результати опрацювання тексту різними моделями.

Результати експериментів

Нетипові дані, у яких значення стандартизованої оцінки більше 3 вилучено для кращої візуалізації розподілів, оскільки їхня частка незначна та вони не впливають на якість класифікації. Візуалізацію розподілів параметрів текстів після опрацювання за допомогою facebook/opt125m зображено на рис 2 – 3. Для інших моделей результати аналогічні (рис 4). Виділяється лише модель Qwen2.5 0.5B [9]. Тут значення d_score мало відрізняється залежно від кількості токенів, що використані. Проте решта параметрів дозволяють відрізнити машинно згенерований текст.

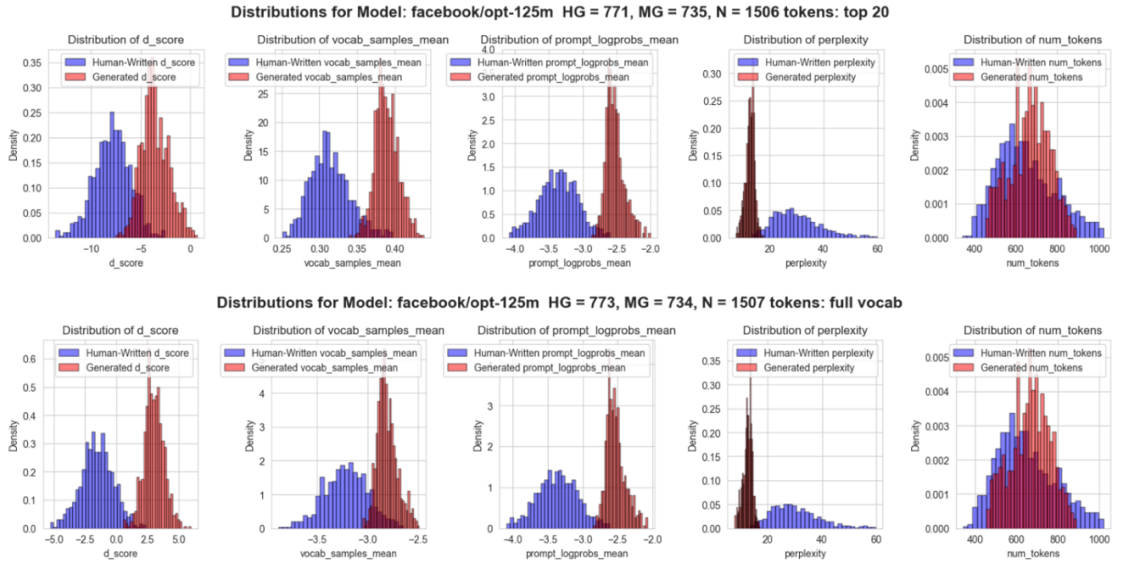

Рис.2. Розподіл параметрів тексту, опрацьованого з facebook/opt125m

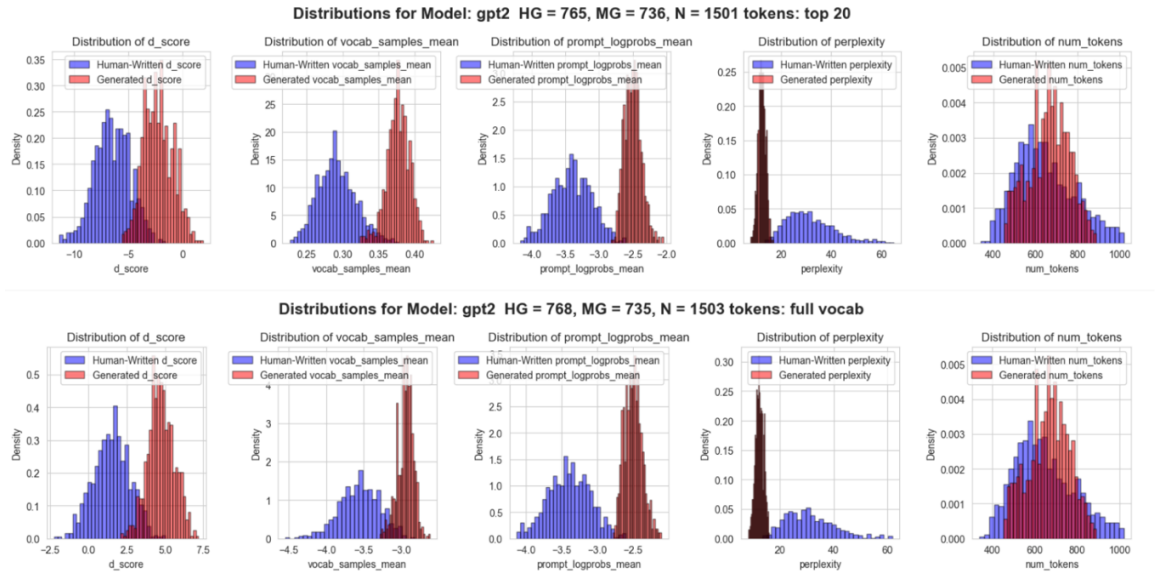

Рис.3. Розподіл параметрів тексту, опрацьованого з gpt2

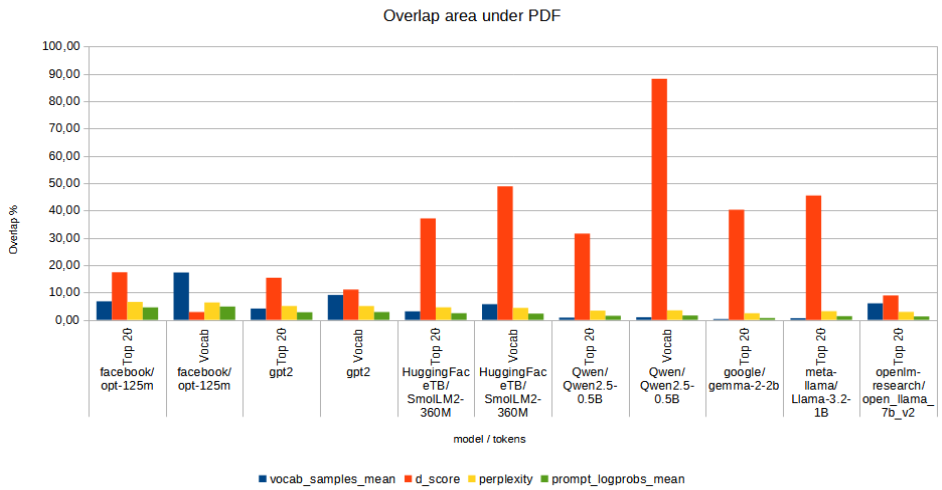

Рис.4. Площа перетину функцій густини імовірності

Видно, що розподіл значень *d_score* відрізняється залежно від того, чи використовується словник чи лише 20 токенів. Натомість площа перетину розподілів значення *vocab_logprobs_mean* не перевищує 20% не залежно від кількості використаних токенів.

Порівняємо точність порогової класифікації текстів за різними властивостями. Результати з найвищим значенням метрики *macro f1* наведено в Таблиці 1.

Таблиця 1.

Точність порогової класифікації текстів за різними властивостями

Model	tokens	vocab_samples_mean	d_score	perplexity	prompt_logprobs_mean
facebook/opt-125m	Vocab	0,911	0,973	0,969	0,970
facebook/opt-125m	Top 20	0,948	0,906	0,969	0,970
gpt2	Vocab	0,939	0,922	0,983	0,983
gpt2	Top 20	0,955	0,913	0,983	0,983
HuggingFaceTB/SmolLM2-360M	Vocab	0,970	0,792	0,982	0,979
HuggingFaceTB/SmolLM2-360M	Top 20	0,965	0,820	0,982	0,979
Qwen/Qwen2.5-0.5B	Vocab	0,984	0,538	0,983	0,981
Qwen/Qwen2.5-0.5B	Top 20	0,980	0,829	0,983	0,981
google/gemma-2-2b	Top 20	0,981	0,788	0,988	0,988
meta-llama/Llama-3.2-1B	Top 20	0,978	0,764	0,987	0,985
openlm-research/open_llama_7b_v2	Top 20	0,936	0,945	0,985	0,985

За результатами експериментів виявлено, що точність класифікації за значенням *d_score* суттєво залежить від моделі - оцінювача, і у різних випадках варто використовувати імовірності всіх токенів у словнику, або достатньо лише 20.

Характеристика тексту *vocab_samples_mean* дозволяє виявити машинно згенерований текст з точністю *macro f1* більше ніж 0,91 зі всіма використаними моделями. За допомогою Qwen/Qwen2.5-0.5B або google/gemma-2-2b можна досягнути значення 0,98. Значення perplexity та prompt_logprobs_mean залежать лише від вхідного тексту, та дозволяють класифікувати текст з точністю від 0,97 до 0,985, залежно від моделі - оцінювача.

Висновок

Для виявлення машинно згенерованого тексту можна використати не лише метрику d_score, а також perplexity, prompt_logprobs_mean та vocab_samples_mean. Також, для обчислення d_score та vocab_samples_mean можна використовувати логарифмічні імовірності лише 20 найімовірніших токенів, а не всього словника. У випадку з моделями facebook/opt125m та gpt2 така класифікація за значенням vocab_samples_mean дозволяє отримати навіть кращу точність ніж з використанням усього словника. При використанні значення d_score результати для цих двох моделей протилежні - точніша класифікація відбувається на основі всього словника. Точність класифікації macro f1 за prompt_mu_score та perplexity не менша ніж 0,97 в усіх експериментах проте ці значення не залежать від імовірностей можливих токенів. Точність класифікації за prompt_logprobs_mean висока навіть при використанні лише 20 найімовірніших токенів замість усього словника. Такий підхід дозволяє використовувати хмарні сервіси доступу до великих мовних моделей та комерційні моделі, та не вимагає розгортання локальної моделі.

Література

1. Bao, G., Zhao, Y., Teng, Z., Yang, L., & Zhang, Y. (2023). *Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature*. arXiv. https://arxiv.org/abs/2310.05130

2. Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., & Finn, C. (2023). *DetectGPT: Zero-shot machine-generated text detection using probability curvature*. In *Proceedings of the 40th International Conference on Machine Learning*.

3. 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* (pp. 1213–1233). https://doi.org/10.18653/v1/2022.naacl-main.88

4. Ippolito, D., Duckworth, D., Callison-Burch, C., & Eck, D. (2019). *Automatic detection of generated text is easiest when humans are fooled*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

5. Thite, S. (n.d.). *LLM – Detect AI Generated Text Dataset*. Kaggle. https://www.kaggle.com/datasets/sunilthite/llm-detect-ai-generated-text-dataset

6. Facebook AI. (n.d.). *OPT: Open pre-trained transformer language models* [Model card]. Hugging Face. https://huggingface.co/facebook/opt-125m

7. OpenAI Community. (n.d.). *GPT-2* [Model card]. Hugging Face. https://huggingface.co/openai-community/gpt2

8. HuggingFaceTB. (n.d.). *SmolLM2-360M* [Model card]. Hugging Face. https://huggingface.co/HuggingFaceTB/SmolLM2-360M

9. Qwen Team. (n.d.). *Qwen2.5-0.5B* [Model card]. Hugging Face. https://huggingface.co/Qwen/Qwen2.5-0.5B

10. Meta AI. (n.d.). *Llama-3.2-1B* [Model card]. Hugging Face. https://huggingface.co/meta-llama/Llama-3.2-1B

11. Google. (n.d.). *Gemma 2* [Model card]. Hugging Face. https://huggingface.co/google/gemma-2-2b

12. OpenLM Research. (n.d.). *OpenLLaMA: An open reproduction of LLaMA* [Model card]. Hugging Face. https://huggingface.co/openlm-research/open_llama_7b_v2