УДК 004.9

**О. А. Довгань**

**К. В. Овчинников**

# АЛГОРИТМИ ТОКЕНІЗАЦІЇ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

Вінницький національний технічний університет

***Анотація***

*Дана робота присвячена огляду алгоритмів текстової токенізації сучасних великих мовних моделей.*

**Ключові слова:** токенізація, алгоритм, велика мовна модель.

***Abstract***

*This work is dedicated to the review of algorithms for text tokenization of modern large language models.*

**Key words:** tokenization, algorithm, large language model.

## Вступ

Токенізація – це детермінований алгоритм поділу тексту на менші частини, слова, букви та символи, словосполучення [1]. Будь-який процес токенізації в сучасній великій мовній моделі складається з чотирьох етапів, де перший етап – нормалізація вхідного тексту, другий – претокенізація, у Claude, GPT, Mistral це розбиття послідовності по пробілах для подальшого застосування алгоритму токенізації розбиття на під слова, у Llama та GPT – застосування регулярних виразів, третій – застосування алгоритму токенізації для отримання токенів, четвертий – додавання спеціальних токенів до токенізованої послідовності, який притаманний кожній моделі.

Токенізація впливає на час витрачений на тренування моделі, на затримку під час інференсу та на точність генерації тексту великою мовною моделлю [2].

У даній роботі досліджується токенізація великих мовних моделей.

## Результати досліджень

BPE – це алгоритм компресії даних. Його принцип роботи криється в заміні послідовності байтів, що часто повторюються, не використаним досі байтом. Автори статті Neural Machine Translation of Rare Words with Subword Units [3], що працювали над вдосконаленням нейронно-машинного перекладу, використали BPE для сегментації текстових даних, що призвело до незначного поліпшення якості перекладу та створення ефективнішої системи, за рахунок меншого розміру словника, у порівнянні з іншими системами. У їх роботі [3] вони видозмінили принцип функціонування алгоритму BPE. Спочатку створювався словник символів, який складався з всіх текстових символів тренувального сету, де ключ – це текстовий символ, а значення – частота символу в тренувальному датасеті, потім, за формулою:

$$score = (first\ pair + second\ pair)/vocabulary, \quad (1)$$

де score – це частота комбінації, first pair – це частота першої пари, second pair – частота другої пари, а vocabulary – частота всіх символів, рахувалась частота всіх можливих комбінацій, які можна утворити з символів у словнику, і на основі найпопулярнішої комбінації до словника додавалась нова пара символів і правило їх поєднання. Таким чином, метою тренування є створення n-грам текстових комбінацій. Один з недоліків, який описали автори, це нездатність сегментувати символи, що відсутні в початковому словнику. Це призводить до втрати інформації при перетворенні тексту, як наслідок, погіршенню результату перекладу або генерації тексту. Даний алгоритм має байт реалізацію, яка усуває даний недолік, членами початкового словника якого виступають байти (початковий словник у byte-BPE складається з 256 байтів). Byte BPE

дозволяє не втрачати інформацію у токенізованих послідовностях, тому що будь-який символ можна репрезентувати за рахунок байтів. Це призводить до збільшення розуміння вхідної послідовності та точності генерації тексту великою мовної моделлю. Інференс відбувається за наступним принципом: вхідна послідовність розбивається на найменші одиниці та почергово до неї застосовуються правила поєднання, які були вивченні алгоритмом під час тренування. Розмір словника у BPE рівний кількості початковим символам, утвореним комбінаціям і вивченим правилам. Вивченні правила – це гіперпараметр, що визначає, як довго треба тренувати алгоритм. BPE використовують наступні великі мовні моделі: GPT, Mistral, Llama3.

WordPiece – це алгоритм токенізації розроблений компанією Google. Даний алгоритм токенізації тренується аналогічно до BPE. Проте він має деякі відмінності. Під час тренування зберігається лише словник з утвореними комбінаціями. Частота комбінацій рахується за рахунок формули:

$$score = P(combination)/(P(first\ pair) \times P(second\ pair)), \quad (2)$$

де score – оцінка релевантності комбінації, P(combination) – ймовірність комбінації, P(first pair) і P(second pair) ймовірність окремих складових комбінації. Під час інференсу, алгоритм розбиває вхідну послідовність на складові зліва на право, поступово знаходячи найдовший фрагмент, що точно збігається з наявною комбінацією у словнику. Таким чином, крок за кроком, вхідна послідовність розкладається на елементи, які алгоритм уже вивчив. Алгоритм WordPiece використовують моделі BERT, DistilBERT [4].

SentencePiece – це алгоритм токенізації тексту, що працює на основі BPE або WordPiece. Алгоритми токенізації WordPiece та BPE не підтримують знак пробілу та під час етапу претокенізації даний знак видаляється. Це унеможливлює етап точного відновлення послідовності, яку подали на вхід для виконання токенізації. Тобто, послідовність символів ‘\n’ заміниться на пробіл. SentencePiece усуває даний недолік. Він на етапі токенізації додає до кожного елемента послідовності символ ‘_’, а для алгоритму токенізації використовує BPE або WordPiece. Даний алгоритм для роботи з китайською, корейською, японською мовами не потребує етапу претокенізації, тоді як BPE і WordPiece потребують етапу претокенізації, де для цих трьох мов застосовують спеціальні алгоритми сегментації тексту [5]. Даний алгоритм токенізації використовують моделі T5, ALBERT.

Серія моделей Llama1 та Llama2 використовує токенізатор SentencePiece & BPE, у порівнянні з WordPiece, даний алгоритм токенізації дозволяє уникнути втрати інформації під час токенізації (відсутній токен [UNK]) та алгоритми SentencePiece і BPE є опен соурсними, їх розвиток може бути підтриманий ком’юніті розробників, тоді як WordPiece є приватною власністю Google.

Серія моделей Llama3, компанії Meta, перейшла з SentencePiece & BPE алгоритма токенізації до BPE, що базується на бібліотеці tiktoken [6], розроблений OpenAI. І для цього є декілька причин, перша причина, бібліотека tiktoken була розроблена спеціально для поліпшення швидкості процесу токенізації, автори бібліотеки порівняли швидкість їх імплементації з токенізатором GPT2 від Huggingface, результати зображені на рисунку 1.

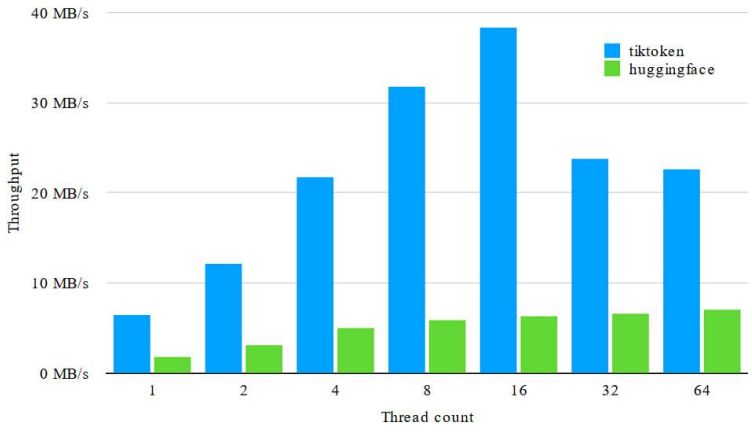


Рисунок 1 – Порівняння швидкості сегментації тексту двома токенізаторами

Друга причина, відповідно до звіту Meta щодо серії моделей Llama3 [7], однією з ключових причин покращення роботи великої мовної моделі стало розширення контекстного до 128 тисяч токенів. Із них 100 тисяч припадають на токени, які обробив tiktoken, а решта 28 тисяч — на неангломовні мови, зокрема українську. Це розширення дозволило збільшити показник компресії даних із 3.17 до 3.94, що свідчить про зменшення розміру вихідної послідовності, яку токенізує алгоритм. Таким чином, поліпшення компресії призвело до того, що велика мовна модель здатна обробляти більший обсяг інформації за той самий бюджет.

## Висновок

У даній роботі були розглянуті алгоритми токенізації сучасних великих мовних моделей. Проведений аналіз змін алгоритмів токенізації у моделях Llama1, Llama2 та Llama3. Продемонстровані переваги алгоритма токенізації Llama3, такі як, збільшення швидкості токенізації, відсутність втрати інформації під час токенізації, збільшення показника компресії даних, що призвело до поліпшення результату генерування тексту в великих мовних моделях серії Llama3.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ


1. What is Tokenization? [Електронний ресурс] – Режим доступу: https://www.datacamp.com/blog/what-is-tokenization
2. Tokenizer Choice For LLM Training: Negligible or Crucial? [Електронний ресурс] – Режим доступу: https://aclanthology.org/2024.findings-naacl.247/
3. Neural Machine Translation of Rare Words with Subword Units [Електронний ресурс] – Режим доступу: https://arxiv.org/abs/1508.07909v5
4. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation [Електронний ресурс] – Режим доступу: https://arxiv.org/abs/1609.08144v2
5. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing [Електронний ресурс] – Режим доступу: https://arxiv.org/abs/1808.06226v1
6. tiktoken [Електронний ресурс] – Режим доступу: https://github.com/openai/tiktoken
7. The Llama 3 Herd of Models [Електронний ресурс] – Режим доступу: https://arxiv.org/abs/2407.21783



***Довгань Олексій Андрійович*** *— студент групи 1ІСТ-23м, магістр кафедри системного аналізу та інформаційних технологій, Вінницький національний технічний університет, Вінниця, e-mail: odovhan08@gmail.com*

***Овчинников Костянтин Вячеславович*** *– к.т.н., доцент кафедри автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: k_ovchinnikov@vntu.edu.ua*

***Dovhan Oleksii Andriovich*** *– student of 1IST-23m, master of the department of system analysis and information technologies, Vinnytsia National Technical University, Vinnytsia, e-mail: odovhan08@gmail.com*

***Ovchynnykov Kostyantyn Vyacheslavovich*** *– Associate Professor of Automation and Intelligent Information Technologies, Faculty of Computer Systems and Automatics Vinnytsia National Technical University, Vinnytsia, e-mail: k_ovchinnikov@vntu.edu.ua*