

# ПОРІВНЯЛЬНИЙ АНАЛІЗ МОЖЛИВОСТЕЙ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ALPACA, VICUNA, FALCON НА ОСНОВІ ТРАНСФОРМЕР-АРХІТЕКТУРИ

Вінницький національний технічний університет

## **Анотація**

*Розглянуто популярні великі мовні моделі з відкритим доступом на основі трансформер-архітектури. Проаналізовано їхні переваги, недоліки та обмеження.*

**Ключові слова:** Alpaca, Vicuna, Falcon, трансформери, тонке налаштування з учителем.

## **Abstract**

*Popular large language models with open access based on the transformer architecture are considered. Their advantages, disadvantages, and limitations are analyzed.*

**Keywords:** Alpaca, Vicuna, Falcon, transformers, supervised fine-tuning.

## **Вступ**

Відбувся величезний прогрес у обробці природної мови з часу появи трансформер-архітектури у 2017 році [1]. Трансформери або їх частини стали основою сучасних мовних моделей. Великі мовні моделі створюються на основі енкодера (BERT та ін.), енкодера-декодера (BART, T5 тощо) і тільки декодера (GPT-1 та інші). У світі спостерігається справжній бум навколо великих мовних моделей на основі декодера [2], завдяки їх вражаючим результатам в машинному перекладі, генеруванні і узагальненні тексту, відповіді на запитання тощо. Конкуренція в цьому напрямку підштовхує компанії випускати нові великі мовні моделі одна за одною, покращуючи їх характеристики і результати відносно своїх конкурентів чи попередніх версій продукту.

## **Огляд великих мовних моделей, що знаходяться у відкритому доступі**

Метою дослідження є огляд популярних великих мовних моделей, що знаходяться у відкритому доступі та аналіз їхніх можливостей і обмежень.

Розглянемо і проаналізуємо три великі мовні моделі з 7 мільярдами параметрів, в основі яких лежить декодер: Alpaca, Vicuna і Falcon.

Alpaca - це велика мовна модель з відкритим кодом, створена дослідниками Стенфордського університету на основі іншої великої мовної моделі LLaMA від Meta AI, шляхом тонкого налаштування з учителем (supervised fine-tuning (SFT) або behavior cloning), використовуючи демонстраційний набір даних. Дослідники підготували демонстраційний набір даних з 52 тисяч інструкцій (запит-відповідь) за допомогою ще однієї великої мовної моделі text-davinci-003 від OpenAI [4]. Такий підхід тонкого налаштування з учителем над попередньо навченою на мільярдах токенів мовною моделлю дає набагато кращі результати. OpenAI довели, що модель з 1,3 мільярдів параметрів показує кращі результати, ніж модель з 175 мільярдами параметрів без тонкого налаштування з учителем [5]. Завдяки невеликій, порівняно з аналогічними моделями, максимальній довжині контексту, що становить 512 символів [5], зменшуються вимоги до оперативної пам'яті графічного процесора. Баланс між відносно якісним результатом роботи у своєму класі та помірним використанням внутрішньої оперативної пам'яті графічного процесора робить Alpaca найпопулярнішою серед моделей, що розглядаються. Про це свідчить розмір спільноти і кількість модифікацій даної мовної моделі. Недоліками і обмеженнями

мовної моделі Alrasa можуть бути: дозвіл на використання тільки в дослідницьких проектах, здатність генерувати тексти неетичного і токсичного змісту, галюцинації.

На основі LLaMA створено ще одну велику мовну модель з відкритим кодом - Vicuna. Загальні кроки процесу навчання даної моделі такі ж як і в Alrasa. Але тонке налаштування з учителем проводилось з набором даних (70 тисяч пар запитів-відповідей), що зібраний з веб-сайту sharegpt.com, де користувачі діляться своїми діалогами з ChatGPT. Варто зазначити, що автори цієї мовної моделі запропонували новий спосіб оцінки якості результату, використовуючи GPT-4 як експерта-оцінювача. Згідно з цим підходом, Vicuna показує понад 90% якості ChatGPT і Bard та перевершує за якістю результату модель Alrasa в більше, ніж 90% випадків [6]. Автори Vicuna покращили розуміння моделлю довшого контексту порівняно з Alrasa. Для цього вони збільшили максимальну довжину контексту до 2048 символів. Однак це має свій недолік, оскільки значно збільшило використання оперативної пам'яті графічного процесора порівняно з Alrasa. Недоліки і обмеження моделі Alrasa також властиві Vicuna, а також вона погано справляється із завданнями, пов'язаними з міркуваннями чи математикою [7].

Нещодавно Інститут технологічних інновацій в Абу-Дабі (Technology Innovation Institute - TII) випустив нову велику мовну модель Falcon у двох варіантах - 40 і 7 мільярдів параметрів. Falcon-40B демонструє найкращі результати і займає перше місце в таблиці лідерів [3], тоді як Falcon-7B є однією з найкращих у своєму класі. Ліцензія Apache 2.0 дозволяє використовувати дану мовну модель в дослідницьких і комерційних проектах, на відміну від попередніх моделей. Розробники навчали Falcon-7B на 1,5 трильйона токенів з високоякісного набору даних RefinedWeb, що очищений від дубльованих записів. Автори Falcon опублікували код моделей і їхні ваги. Також у відкритому доступі наявні SFT-версії зазначених моделей. Особливістю мовних моделей Falcon є використання в них механізму багатозапитової уваги (multiquery attention) [8], що зменшує використання оперативної пам'яті графічного процесора.

### Результати дослідження

Порівняємо якість відповідей моделі з відкритим доступом - Alrasa з ChatGPT. Поставимо 5 запитів кожній з моделей: два запити - розв'язати задачу з програмування на Python з leetcode.com, 3 запити - з теорії обробки природної мови.

Таблиця 1. Відповіді на запити до великих мовних моделей - ChatGPT і Alrasa та оцінки за відповіді

Запит	Відповідь ChatGPT	Відповідь Alrasa	Коментар/Оцінка
Explain layer of attention mechanism in simple terms	<p>Certainly! In the context of the transformer architecture, an attention mechanism is a crucial component that helps the model understand the relationships between different elements in a sequence.</p> <p>Let's say we have a sentence: "The cat sat on the mat." In this sentence, there are different words, or "tokens," such as "cat," "sat," "on," "the," and "mat." The attention mechanism allows the model to determine which tokens are most relevant to each other.</p> <p>To do this, the attention mechanism assigns weights to each token, indicating their importance or relevance. These</p>	<p>The layer of attention mechanism is the part of the model that determines which parts of the input data should be focused on. This layer is responsible for determining which parts of the input data should be focused on. This layer is responsible for determining which parts of the</p>	<p>ChatGPT : 5/5 Alrasa: 2/5</p>

	<p>weights are calculated based on the similarity or compatibility between the tokens. Tokens that are more related to each other will have higher weights.</p> <p>The attention mechanism works in three main steps:</p> <ol style="list-style-type: none"> <li>1. Query, Key, and Value: Each token in the sequence is transformed into three vectors: a query vector, a key vector, and a value vector. These vectors capture different aspects of the token's information.</li> <li>2. Similarity Calculation: The attention mechanism calculates the similarity between the query vector of a token and the key vectors of all the other tokens in the sequence. This is usually done using a mathematical operation called dot product, which measures the similarity between two vectors.</li> <li>3. Attention Weights and Aggregation: The similarity scores obtained in the previous step are transformed into attention weights using a softmax function. The softmax function normalizes the scores, ensuring that they sum up to 1. These weights represent the importance of each token in relation to the others. Finally, the attention weights are used to compute a weighted sum of the value vectors, resulting in a context vector that represents the attended information from the sequence.</li> </ol> <p>In simple terms, the attention mechanism allows the model to focus on different parts of the input sequence while considering the relationships between the tokens. It assigns weights to the tokens based on their relevance to each other and combines their information to create a meaningful representation.</p> <p>This mechanism has proven to be powerful in capturing long-range dependencies and understanding the context in NLP tasks. It has significantly contributed to the success of the</p>	<p>input data should be focused on. This layer is responsible for determining which parts of the input data should be focused on. This layer is responsible for determining which parts of the input data should be focused on. This layer is responsible for determining which parts of the input data should be focused on. This layer is responsible for determining which parts of the input data should be focused on.</p>	
--	--	---	--

	transformer architecture in various applications.		
<p>What are hallucinations in large language models?</p>	<p>Hallucinations in large language models refer to instances where the model generates outputs that are incorrect, nonsensical, or unrelated to the input or the desired task. It can produce seemingly plausible but false information that does not exist or may even contradict reality.</p> <p>Hallucinations can occur due to several reasons:</p> <ol style="list-style-type: none"> <li>1. Over-optimization: Large language models are trained to generate coherent and contextually relevant text based on the patterns and examples they have learned during training. However, they can sometimes become overly confident in their predictions, leading to the generation of plausible-sounding but incorrect information. This occurs when the model has "memorized" patterns in the training data but fails to truly understand the underlying concepts.</li> <li>2. Inference from incomplete or biased data: Language models are trained on vast amounts of text from the internet, which contains a wide range of information, including errors, biases, and unverified claims. The model may unintentionally generate responses that reflect or amplify these biases or incorporate false information from unreliable sources.</li> <li>3. Lack of contextual understanding: While large language models excel at generating text that appears coherent on a sentence level, they may struggle with deeper understanding of context, world knowledge, or logical reasoning. This can lead to responses that are contextually incorrect or nonsensical, resembling hallucinations.</li> </ol>	<p>Hallucinations in large language models refer to the phenomenon in which the model generates text that does not correspond to the input data. This can occur when the model is overfitting to the input data, or when the model is generating text that does not make sense.</p>	<p>ChatGPT : 5/5 Alpaca: 3/5</p>

	<p>4. Exposure to adversarial inputs: When large language models are exposed to deliberately crafted inputs designed to deceive or exploit their weaknesses, they may produce misleading or false outputs. Adversarial inputs can exploit vulnerabilities in the model's architecture or training process, causing it to generate hallucinatory responses.</p> <p>Addressing hallucinations is an ongoing challenge in the development of language models. Researchers and engineers continuously work on improving model architectures, training methodologies, and evaluation techniques to mitigate the occurrence of hallucinations and enhance the reliability and accuracy of large language models.</p>		
<p>Explain transformer architecture in simple terms</p>	<p>Sure! The transformer architecture is a type of deep learning model that has revolutionized natural language processing (NLP) tasks. It's called a "transformer" because it transforms input data into meaningful representations.</p> <p>At its core, the transformer architecture is made up of two main components: the encoder and the decoder. Both the encoder and decoder consist of multiple layers of attention mechanisms and feed-forward neural networks.</p> <p>The encoder's job is to take an input sequence, such as a sentence, and convert it into a set of hidden representations. It processes the input sequence one token at a time and applies attention mechanisms to capture the relationships between different tokens. The attention mechanism allows the model to focus on relevant parts of the input sequence and weigh their importance in generating the representations.</p> <p>The decoder, on the other hand, takes the encoder's hidden representations and generates an output sequence. It also uses attention mechanisms, but with a slight modification called "masked attention" to ensure that the model only attends to tokens that have already been</p>	<p>A transformer is a type of neural network. It is composed of multiple layers of neurons. Each layer is connected to the previous layer, and the output of each layer is connected to the input of the next layer. The output of each layer is used as the input of the next layer. The transformer architecture is composed of multiple layers of neurons. Each layer is connected to the previous layer, and the output of each layer is connected to the input of the next layer. The output of each layer is used as the input of the next layer. The output of each layer is used as the input of the next layer.</p>	<p>ChatGPT : 5/5 Alpaca: 1/5</p>

	<p>generated, preventing it from "cheating" by looking ahead in the output sequence.</p> <p>During training, the transformer learns to optimize its parameters by minimizing the difference between its predicted output and the desired output using a technique called backpropagation. This allows the model to learn the patterns and relationships in the input data.</p> <p>One of the key advantages of the transformer architecture is its ability to handle long-range dependencies in sequences. Unlike previous models that relied on recurrent neural networks (RNNs), which process sequences sequentially, the transformer can process all tokens in parallel. This parallel processing enables faster training and inference, making it highly efficient.</p> <p>Transformers have been incredibly successful in NLP tasks such as machine translation, text summarization, and question answering. They have also been applied to other domains like computer vision and speech recognition, demonstrating their versatility and effectiveness in various applications.</p>	<p>each layer is used as the input of the next layer. The output of each layer is used as the input of the next layer. The output of each layer is used as the input of the next layer. The output of each layer is used as the input of the next layer. The output of each layer is used as the</p>	
<p>Please write a code in Python to solve the problem. Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target.</p> <p>You may assume that each input would have exactly one solution, and you may not use the same element twice.</p> <p>You can return the</p>	<pre>def twoSum(nums, target):     num_map = {}     for i, num in enumerate(nums):         complement = target - num         if complement in num_map:             return [num_map[complement], i]         num_map[num] = i     return []</pre>	<pre>def two_sum(nums, target):     for i in range(len(nums) - 1):         for j in range(i + 1, len(nums)):             if nums[i] + nums[j] == target:                 return [i, j]     return []</pre>	<p>Обидві моделі впорались із завданням, але складність розв'язку ChatGPT є меншою - <math>O(n)</math>, ніж складність розв'язку Alpaca - <math>O(n^2)</math>.</p> <p>ChatGPT : 5/5 Alpaca: 4/5</p>

answer in any order.			
<p>Please write a code in Python to solve the problem. Given a string s, find the length of the longest substring without repeating characters.</p>	<pre>def lengthOfLongestSubstring(s):     unique_chars = set()     left = right = 0     max_length = 0      while right &lt; len(s):         if s[right] not in unique_chars:             unique_chars.add(s[right])             max_length = max(max_length, right - left + 1)             right += 1         else:             unique_chars.remove(s[left])             left += 1      return max_length</pre>	<pre>def longestSubstringWithoutRepeating(s):     longestSubstring = ""     for i in range(len(s)):         for j in range(i+1, len(s)):             if s[i] != s[j] and s[i] != s[j+1] and s[i] != s[j+2] and s[i] != s[j+3] and s[i] != s[j+4] and s[i] != s[j+5] and s[i] != s[j+6] and s[i] != s[j+7] and s[i] != s[j+8] and s[i] != s[j+9] and s[i] != s[j+10] and s[i] != s[j+11] and s[i] != s[j+12] and s[i] != s[j+13] and s[i] != s[j+14] and s[i] != s[j+15] and s[i] != s[j+16] and s[i] != s[j+17] and s[i] != s[j+18] and s[i] != s[j+19] and s[i] != s[j+20] and s[i] != s[j+21] and s[i] != s[j+22] and s[i] != s[j+23] and s[i] != s[j+24] and s[i] != s[j+25] and s[i] != s[j+26] and s[i] != s[j+27] and s[i] != s[j+28] and s[i] != s[j+29] and s[i] != s[j+30] and s[i] != s[j+31] and s[i] != s[j+32] and s[i] != s[j+33] and s[i] != s[j+34] and s[i] != s[j+35] and s[i] != s[j]</pre>	<p>ChatGPT : 5/5 Alpaca: 0/5</p>

ChatGPT отримав 25 балів з 25 можливих, тоді як Alpaca тільки 10. Відповіді ChatGPT повні і відповідають запиту. Alpaca заціклюється на повторенні послідовності слів після деякої частини згенерованого тексту, а також тривалість генерації результату значно перевищує час відповіді ChatGPT і досягає на деяких запитах 2,5 хвилин.

## Висновки

Отже, охарактеризувавши популярні великі мовні моделі, що є у відкритому доступі, і порівнявши одну з них з ChatGPT, можна підсумувати, що досягнувши вражаючих результатів, великі мовні моделі все ж не позбавлені недоліків та обмежень і все ще відстають від такої потужної мовної моделі як ChatGPT. Основні недоліки: здатність генерувати тексти неетичного і токсичного змісту, галюцинації, погано справляються із завданнями, пов'язаними з математикою чи міркуваннями. За таблицею лідерів від Hugging Face, з-поміж розглянутих великих мовних моделей у відкритому доступі, найкращою є Vicuna\* (див. таблицю 1). Але потрібно зважати на той факт, що сім'я моделей Falcon з'явилася відносно недавно і має багатообіцяючі перспективи, а техніками квантування моделей дозволяють проводити тонке налаштування мовних моделей з 7 мільярдами параметрів на графічному процесорі Tesla T4 з 16 Гб оперативної пам'яті в безкоштовній версії Google Colab. Тому варто очікувати покращення результативності моделі від учасників спільноти.

Таблиця 2. Результативність великих мовних моделей загального призначення з 7-ма мільярдами параметрів. Пояснення контрольних тестів:

AI2 Reasoning Challenge (25-shot) - набір запитань з природознавства для початкової школи;

HellaSwag (10-shot) - тест на здоровий глузд, який легкий для людей (~95%), але складний для моделей SOTA.

MMLU (5-shot) - тест для вимірювання багатозадачної точності текстової моделі. Тест охоплює 57 завдань, серед яких елементарна математика, історія США, інформатика, право тощо.

TruthfulQA (0-shot) - тест для вимірювання схильності моделі відтворювати неправду, яку зазвичай можна знайти в Інтернеті.

\*Примітка: вибрано мовні моделі загального призначення з максимальним середнім значенням результату серед усіх моделей сім'ї

Автор/Назва моделі	Середнє значення	AR C (25-s)	HellaSwag (10-s)	MM (5-s)	TruthfulQA (0-s)
eachadea/vicuna-7b-1.1	52.2	47	75.2	37.5	48.9
chavinlo/alpaca-native	49.6	48.9	76.1	36.3	37.2
tiiuae/falcon-7b-instruct	48.4	45.9	70.8	32.8	44.1

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ashish Vaswani, Noam Shazeer, Niki Parmar, and others. Attention Is All You Need. June 2017. <https://arxiv.org/abs/1706.03762>
2. Jingfeng Yang, Hongye Jin, Ruixiang Tang and others. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. April 2023. <https://arxiv.org/abs/2304.13712>
3. Open LLM Leaderboard. [https://huggingface.co/spaces/HuggingFaceH4/open\\_llm\\_leaderboard](https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard)
4. Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, and others. Alpaca: A Strong, Replicable Instruction-Following Model. March 2023. <https://crfm.stanford.edu/2023/03/13/alpaca.html>
5. Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, and others. Stanford Alpaca: An Instruction-following LLaMA Model. March 2023. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca)



6. Long Ouyang, Jeff Wu, Xu Jiang, and others. Training language models to follow instructions with human feedback. March 2022. <https://arxiv.org/abs/2203.02155>
7. Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, and others. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%\* ChatGPT Quality. March 2023. <https://lmsys.org/blog/2023-03-30-vicuna/>
8. Leandro von Werra, Younes Belkada, Sourab Mangrulkar, Lewis Tunstall. The Falcon has landed in the Hugging Face ecosystem. June 2023. <https://huggingface.co/blog/falcon>

**Левіцький Сергій Мойсейович** - аспірант кафедри системного аналізу та інформаційних технологій

**Levitskyi Serhii Moiseiovych** - graduate student of the Chair of System Analysis and Information Technologies