УДК 004.891:159.953:51
DOI https://doi.org/10.32782/2663-5941/2025.4.2/14

Дадиверін В.В.
Вінницький національний технічний університет

Бісікало О.В.
Вінницький національний технічний університет

АНАЛІЗ ТА ІМПЛЕМЕНТАЦІЯ ІДЕЇ НАВЧАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ЗА АНАЛОГІЄЮ З ДИТЯЧИМ КОГНІТИВНИМ РОЗВИТКОМ

Дослідження присвячено проблематиці недостатньої ефективності великих мовних моделей у питаннях розв'язання математичних завдань, послідовного багатокрокового аналізу та міркування щодо можливих варіантів вирішення. У статті розкрито недосконалість сучасних LLM у питаннях послідовного мислення, попри їхню здатність до генерації природного та осмисленого тексту, вони досі залишаються суттєво обмеженими при вирішенні задач, які потребують логічного міркування або обґрунтування. З'ясовано, що такі обмеження нерідко є наслідком структури навчального процесу мовної моделі, бо той не має нічого спільного з процесами, які відбуваються під час формування когнітивних та ментальних моделей людини.

У цій роботі розглянуто підхід навчання за подібністю до когнітивного розвитку людини, який базується на поєднанні наступних концепцій: Curriculum learning – навчання з поетапним збільшенням складності завдань, Scaffolding – сегментоване навчання зі зменшенням рівня опіки після засвоєння певного рівня знань, та прогресивне зміцнення ментальних структур за рахунок створення узагальнених патернів для розв'язання задач.

Основну увагу зосереджено на імплементації навчального процесу з поетапним ускладненням вхідного матеріалу – від базових арифметичних рівнянь до великих складених логіко-словесних задач. Ефективність даного підходу оцінено на прикладі моделей Phi-2 та Mistral 7B, навчених в двох режимах: стандартне випадкове навчання та запропоноване контрольоване навчання.

Результатами дослідження доведено, що запропонований підхід когнітивного навчання забезпечує кращі результати під час вирішення базових задач та складних багатокомпонентних завдань, які потребують проміжних узагальнених знань. Аналізом подібних досліджень підтверджено перспективність розвитку навчання з імітацією когнітивного розвитку для підвищення ефективності обробки вхідних даних за умов обмеженої кількості навчальної інформації.

Ключові слова: *велика мовна модель, когнітивний розвиток, адаптивне навчання, Mistral 7B, Phi-2, curriculum learning, scaffolding.*

Постановка проблеми. Великі мовні моделі досягають все нових і нових вершин у завданнях розпізнавання мовленнєвої інформації та генерації якісної відповіді чи текстового доробку, проте ми і далі стикаємося з проблемою вирішення математичних задач навіть базового рівня. Великі LLM, за типом Ghat GPT, показують вагому нестабільність у задачах математичного міркування та скромну здатність до узагальнення методології вирішення [1]. У свою чергу, це означає, що модель у більшості випадків легко справляється з задачами, що мають типовий зміст, але і так само легко допускає помилки у розумінні задач з нетиповим описом або сценарієм. Це відбувається через те, що вирішення математичних задач потребує відповідний набір навичок – від стандартного семантичного розбору тексту до логічних висновків та математичних операцій [1]. Мовні моделі, що в основі своїй здобували навички на статистичних закономірностях, не є гнучкими у питаннях обробки та виконання багатокрокових завдань.

Визначимо бар'єри, що зазвичай обмежують можливості мовної моделі успішно вирішувати прості та нестандартні математичні задачі:

1. Відсутність поступового навчання. На відміну від людей, які опановують математичні знання поступово, підвищуючи складність завдань, мовні моделі зазвичай отримують свої знання зі змішаного набору вхідних даних. Тобто, не використовується принцип Curriculum learning, який за свою суттю має мету до створення правильних причинно-наслідкових зв'язків. Останні дуже потрібні для вирішення задач з нестандартною умовою, бо вони дають можливість розібрати задачу на прості операції.

2. Брак самоконтролю. Зазвичай LLM-ки не спраглі до перевірки своїх відповідей, бо не мають створеної потреби до постійного сумніву. Вони часто наводять недостовірну інформацію або, взагалі, вдаються до «вигадувань» і подають таку інформацію як достовірну та єдино правильну.

3. Відсутність плану дій та некоректне збереження проміжних результатів. Досить часто мовні моделі не мають достатньої кількості кешпам'яті для збереження великого масиву проміжних результатів або не створюють логічного зв'язку для цих даних. Навіть неймовірно потужні модель, до прикладу GPT-4, припускаються арифметичних помилок через структуру плану дій під обробки [2].

4. Екстраполяція отриманих знань. Навчання моделі на текстових шаблонах не дає можливості сформувати глибинні концепції обробки вхідних даних і тому це не дозволяє моделі використовувати знання гнучко, а тільки опрацьовувати задачі з подібною умовою.

Для людей опанування математичних знань не вважається натуральним (природним) процесом психологічного розвитку, бо такого роду інформація засвоюється тільки завдяки «вчителю», який є частиною соціального контексту, що в ручному режимі керує процесом ускладнення рівня вхідного матеріалу та виступає додатковою підтримкою чи контролером якості засвоєння пройденого матеріалу [3]. Ці думки є розвитком початкової концепції Scaffolding learning [4], сформованої у 1970-х роках Д. Вудом, Дж. Брунером та Г. Россом, – дозована підтримка учня дає йому змогу вирішувати задачі, що перевищують рівень індивідуальних знань та вмінь. Аналогічний онтогенетичний підхід до накопичення знань було формалізовано в роботах [5, 6].

Отже, ми приходимо до логічного висновку, що навчання дітей математиці є виваженим та структурованим процесом соціальної взаємодії та співпраці. Великі мовні моделі здебільшого позбавлені такого методу контролю та настанови, що не дає можливість зрозуміти глибину випадково отриманих прикладів та препарувати їх для розширення кругозору знань.

Ми схильні вважати, що наведені проблеми можуть бути усунуті завдяки впровадженню принципів онтогенетичного когнітивного розвитку людини: Scanfolding learning + поступова зміна складності задач та їх тематики. Ці принципи є основою даного дослідження.

Аналіз останніх досліджень і публікацій. Останні 5 років дослідники плідно займаються питанням покращення математичної логіки великих мовних моделей і за типом ці групи досліджень можна поділити на:

- Покращення математичного мислення через спеціалізоване донавчання;
- Навчання за принципами розвитку людських когнітивних здібностей.

До першої групи досліджень належить стаття «Large Language Models for Mathematical Reasoning: Progresses and Challenges» [1], опублікована у 2024 році, і там описується застосування методу «Program of Thought» до GPT-4. Цей метод займається генеруванням послідовних програм (алгоритмів) у формі міркування, у цьому дослідженні він використовувався для прописування моделлю явних кроків розв'язку відповідної задачі та генерації коду для перевірки правильності відповіді, подібно до того, як може бути організовано в автоматизованих тестах або unit-тестах. При цьому було отримано результат у вигляді приросту ефективності на 15% при вирішенні складних математичних задач.

Інше дослідження «Minerva: Solving Quantitative Reasoning Problems with Language Models» [7] про створення системи Minerva, що має донавчати мовні моделі, використовуючи великий масив математичних завдань та прикладів можливих рішень. Цей підхід також полягав у застосування схожого принципу «Chain of Thought», де генеруються випадкові ланцюжки вирішень, потім серед створених обираються найбільш схожі за змістом і на етапі навчання порівнюються з наявними прикладами. Така синергія дій дає можливість моделі створювати якісніші ланцюги міркування і результатом такого донавчання є приріст у якості на 10% і вище в порівнянні з базовими моделями.

До другої групи досліджень належить стаття «Scaffolding learning: From specific to generic with large language models» [8], опублікована у 2024 році, в цій роботі описується застосування методу Scanfolding при двоетапному навчанні

моделі: перший етап – вирішення простих арифметичних операцій, другий етап – складні та багатоетапні задачі, в основному алгебраїчні. В цьому доробку намагалися формалізувати підхід навчання дітей базовим навичкам з переходом до більш складних викликів в момент готовності до глибшого розбору. Результат показав в середньому покращення ефективності вирішення завдань на 30% в порівнянні з моделями, що не мали алгоритму поетапної підтримки.

Друге дослідження «Curriculum learning» [9] представляє вперше однойменну концепцію навчального процесу з поетапним логічним зростанням складності, бо такий тип навчання дозволяє досягти швидшої збіжності та показати кращу здатність до узагальнення алгоритму вирішення завдань в порівнянні з випадково отриманими навчальними даними. У ході експерименту навчені моделі показали вищу стабільність вирішення однотипних та модифікованих за текстом задач, зокрема, моделі, що навчалися для вирішення задач класифікації послідовностей показали приріст у точності на 7–10%.

Постановка завдання. Метою даної роботи є експериментальна перевірка методу поетапного адаптивного навчання LLM, яка буде спиратися на принципи онтогенетичного когнітивного розвитку дитини задля покращення ефективності вирішення математичних задач. Очікується, що впровадження такого типу навчання сприятиме підвищенню точності та можливості розв'язання задач різної складності моделями Mistral 7B [10] та Phi-2 [11]. Порівняння проводитиметься між моделями, що пройшли хаотичне та запропоноване адаптивне навчання.

Виклад основного матеріалу. Прогресивне адаптивне навчання великої мовної моделі досягається через поетапний виклад знань, тобто перехід на наступний рівень тренування відбувається тільки після успішного завершення минулого кроку, головною основою даного підходу є допоміжна підказка або проміжна дія, що має ключове значення для розв'язку, тобто спочатку надається значна підтримка для моделі у вигляді легких прикладів та розбиттю завдання на підзадачі, але поступово ця підтримка згасає і модель має самостійно виконувати таку роботу. Саме ця частина навчання відповідає принципу покращення когнітивних здібностей дитини в процесі нормального онтогенезу, бо дитина (модель) отримує інформацію для спостереження і сприймає підказки від учителя, щоб сформувати методологію та самостійне вміння до такого роботи.

Моделі та їхні налаштування для навчання

Для навчання було обрано дві моделі з різною кількістю параметрів:

- Mistral 7B – 7 мільярдів параметрів [10].

Відносно сучасна модель, яка відрізняється своєю ефективністю, бо за якістю в багатьох задачах перевищує більші моделі, що мають 13 млрд. параметрів (наприклад, Llama-2 показує себе гірше в роздумах, математиці та генерації коду [10]).

- Phi-2 – 2.7 мільярдів параметрів [11].

Досить невелика та компактна модель від Microsoft, яка круто себе проявляє у задачах пов'язаних з міркуванням та трактуванням природної мови. Як зазначають розробники, можливості моделі вражають, бо результати замірів розуміння мови та математичних здібностей як мінімум паритетні, а як максимум кращі за більшість 7 та 13 млрд. моделей [11].

Оскільки навчання моделей з багатьма мільярдами параметрів вимагає велику кількість ресурсів, то було вирішено займатися саме донавчанням або fine-tuning на заготовленому наборі задач. Для цього було обрано методику Low-Rank Adaptation або LoRa [12], бо вона дозволяє навчати лише частину додаткових параметрів, що у свою чергу суттєво знижує вимоги до кількості пам'яті та обчислень. На практиці це дозволяє навчати такі моделі як Mistral 7B, використовуючи один процесор та відеокарту середнього рівня продуктивності (наприклад, RTX 3060) [13].

Системні ресурси ПК, які були використанні для донавчання обох моделей:

- Процесор: AMD Ryzen 7 7700X 8-Core, 4.5 GHz;
- Оперативна пам'ять: DDR5 32 GB, 6000 MHz;
- Відеокарта: NVIDIA GeForce RTX 3060, 12 GB;
- Операційна система: Windows 11 Pro, 64-bit, x64-based.

Для досягнення оптимальних параметрів навчання було застосовано техніку акумуляції градієнтів з наступними параметрами: gradient accumulation steps = 4 та batch size = 1, це дозволяє досягти ефективний розмір партії – 5 і не завищувати споживання пам'яті. Hyper-параметри можна побачити на рисунку 1, де наведено приклад конфігурації для моделі Phi-2.

Для завантаження моделей було використано бібліотеку Hugging Face Transformers [14]. На рисунку 2 можна побачити яким чином запускається цикл автоматизації та застосовується LoRA.

```
{
    "model_name": "microsoft/phi-2",
    "train_file": "finetune/train_progressive.jsonl",
    "val_file": "finetune/val.jsonl",
    "output_dir": "checkpoints/phi2_finetune",
    "per_device_train_batch_size": 1,
    "gradient_accumulation_steps": 4,
    "learning_rate": 5e-5,
    "num_train_epochs": 3,
    "fp16": false
}
```

Рис. 1. Параметри конфігурації для моделі Phi-2

```
import json
from transformers import ( AutoModelForCausalLM, AutoTokenizer, Trainer, TrainingArguments, DataCollatorForLanguageModeling)
from peft import LoraConfig, get_peft_model, TaskType
from datasets import load_dataset
# Завантаження конфігурації, токенізатора та моделі
with open("phi2_config.json") as f:
    cfg = json.load(f)
model_name = cfg["model_name"]
tokenizer = AutoTokenizer.from_pretrained(model_name, trust_remote_code=True)
model = AutoModelForCausalLM.from_pretrained(model_name, trust_remote_code=True)
# Підключення LoRA-адаптера до моделі
lora_config = LoraConfig(r=8, lora_alpha=32, target_modules=["q_proj", "v_proj"], lora_dropout=0.1, bias="none", task_type=TaskType.CAUSAL_LM)
model = get_peft_model(model, lora_config)
# Підготовка датасетів
train_dataset = load_dataset("json", data_files=cfg["train_file"], split="train")
val_dataset   = load_dataset("json", data_files=cfg["val_file"], split="train")
# Токенізація (додавання відповіді до запитання через роздільник)
def tokenize(example):
    full_text = str(example["input"]) + "\n" + str(example["output"])
    tokenized = tokenizer(full_text, padding="max_length", truncation=True, max_length=512)
    tokenized["labels"] = tokenized["input_ids"].copy()
    return tokenized
train_dataset = train_dataset.map(tokenize, remove_columns=train_dataset.column_names)
val_dataset   = val_dataset.map(tokenize, remove_columns=val_dataset.column_names)
# Встановлення аргументів тренування
training_args = TrainingArguments(
    output_dir=cfg["output_dir"],
    per_device_train_batch_size=cfg["per_device_train_batch_size"],
    gradient_accumulation_steps=cfg["gradient_accumulation_steps"],
    learning_rate=cfg["learning_rate"],
    num_train_epochs=cfg["num_train_epochs"],
    save_strategy="epoch",
    evaluation_strategy="epoch",
    logging_steps=10,
    fp16=cfg.get("fp16", False),
    remove_unused_columns=False
)
# Тренер
trainer = Trainer(model=model, args=training_args,train_dataset=train_dataset, eval_dataset=val_dataset, tokenizer=tokenizer,
                  data_collator=DataCollatorForLanguageModeling(tokenizer, mlm=False))
# Запуск навчання
trainer.train()
```

Рис. 2. Фрагмент коду, де ініціалізуються моделі та запускається процес навчання

Структура навчання та приклади задач

Структурно навчальний план складався з ієрархії за типом та складністю самих задач. Перелік задач за складністю наступний:

- Арифметика: додавання, віднімання, множення та ділення.
- Базові логічні задачі: задачі з висновком, порівняння, головоломки (по типу, хто старший, швидший, вищий).
- Задачі на побудову логічного ланцюжку.
- Геометричні задачі: обчислення площ, кутів, задачі з формулами.
- Багатокрокові алгебраїчні задачі: системи рівнянь, текстові задачі, що мають декілька пов'язаних паралельних дій.

Приклад тренувальних задач можна побачити на рисунку 3.

У ході навчання було застосовано трохи менше 500 задач, які у різних долях охопили весь перелік задач за типами, розподіл був наступний: дода-

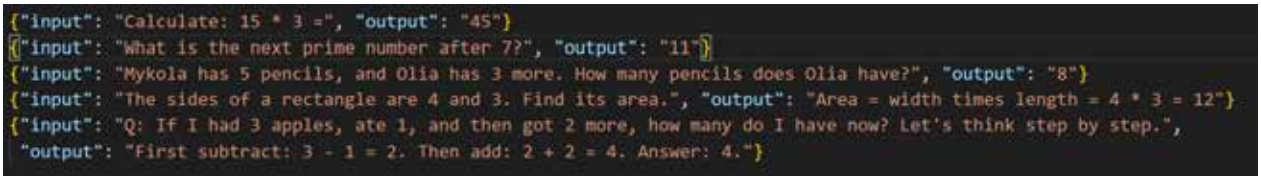

Рис. 3. Приклад навчальних задач

Рис. 4. Консольне логування процесу донавчання моделі Mistral 7B

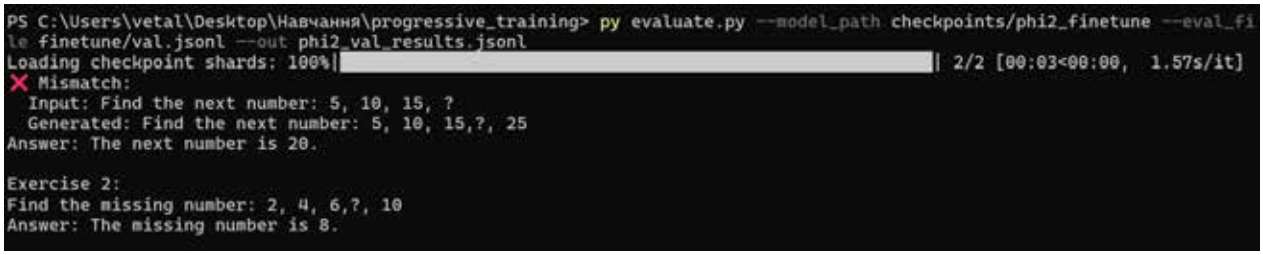

Рис. 5. Приклад вирішення задач моделлю Mistral 7B

Таблиця 1

Результати точності вирішення задач для обох моделей

Модель	Без донавчання	Хаотичне донавчання	Адаптивне донавчання
Phi-2	45.7%	62.8%	74.2%
Mistral 7B	51.4%	65.7%	82.8%

вання/віднімання – 17%, множення/ділення – 15%, базові текстові задачі – 16%, істинність/порівняння – 12%, ланцюгові міркування – 18%, геометричні задачі – 12%, багатокрокові задачі – 10%.

Контроль прогресу проходив через перевірочний набір на етапі кожного типу задач, перехід до наступного етапу відбування при досягненні порогової точності у 80±3%. Кожний наступний етап – це змішаний набір задач наступного типу і легке вкраплення задач, на яких модель навчалась на попередньому етапи, щоб досягти кумулятивного ефекту. Така схема дозволяє уникнути перевантаження моделі та ресурсів ПК на якому проводилось навчання. Фрагмент навчання моделі Mistral 7B можна побачити на рисунку 4.

Оцінка результатів

Оцінювання результатів проводилось на трьох типах моделей: без донавчання, з адаптивним навчання, з хаотичним навчанням (добірка задача та сама, що і для адаптивного). Метрикою оцінки виступало кількість задач, на яку модель дала правильну відповідь, тобто так звана точність виконання або accuracy. Тестовий набір складався з 35 задач та вони відрізнялись від тих, що були використані під час тренувань моделей. Приклад логування успішності вирішення задач можна побачити на рисунку 5.

Результати замірів представлені у Таблиці 1.

Висновки. В роботі було проведено аналіз відомих досліджень у сфері покращення «мате-

матичного мислення» LLM та вироблено методику навчання, яка ефективно покращує можливості наявної моделі при вирішенні математичних задач. В обох випадках адаптивного тренування навчена модель показала приріст на майже 30% в порівнянні з базовим рішенням. Різниця приросту між адаптивним та хаотичним навчанням довела, що правильна послідовність тренувальних задач, поступове ускладнення та контроль засвоєння сприяють утворенню більш ефективнішої моделі для вирішення математичних задач. Варто зазначити, що філософія моделі Phi-2 чітко корелюється з отриманими результатами, бо правильно підібрані дані та стратегія навчання дозволяють конкурувати з моделями, які мають значно більшу кількість параметрів. Підсумовуючи, можна сказати, що підхід навчання великих мовних моделей за аналогією з дитячим когнітивним розвитком продемонстрував свою дієвість і має подальший потенціал для розвитку.

Список літератури:

1. Ahn J., Verma R., Lou R., Liu D., Zhang R., Yin W. Large Language Models for Mathematical Reasoning: Progresses and Challenges. *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics:* Student Research Workshop, St. Julian's, Malta, 2024. – Association for Computational Linguistics. – P. 225–237.

2. Chang F.-C., Lin Y.-C., Wu P.-Y. Unraveling Arithmetic in Large Language Models: The Role of Algebraic Structures. 2024. URL: https://arxiv.org/abs/2411.16260.

3. Rogoff B. Cultural Nature of Human Development. – Oxford University Press, 2003. – 248 p.

4. Wood D., Bruner J. S., Ross G. The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*. Vol. 17, no. 2. 1976. P. 89–100. URL: https://doi.org/10.1111/j.1469-7610.1976.tb00381.x.

5. Бісікало О.В. Онтогенетичний метод побудови нечіткого відношення сенсу / О.В. Бісікало // Штучний інтелект. – 2011. – № 1. – С. 134–140.

6. Бісікало О.В. Формальні методи образного аналізу та синтезу природно-мовних конструкцій : монографія / О. В. Бісікало. – Вінниця : ВНТУ, 2013. – 316 с. – ISBN 978-966-641-528-1.

7. Lewkowycz A., Andreassen A., Dohan D., Dyer E. and others. Solving Quantitative Reasoning Problems with Language Models. *Proceedings of the 36th Conference on Neural Information Processing Systems*. 2022. P. 1-15. URL: https://surli.cc/rwwgap

8. Yin D. S., Yin X. Scaffolding learning: From specific to generic with large language models. *PLOS ONE*. 2024. Vol. 19, no. 9. P. e0310409. URL: https://doi.org/10.1371/journal.pone.0310409.

9. Bengio Y., Louradour J., Collobert R., Weston J. Curriculum learning. *Proceedings of the 26th Annual International Conference (ICML 2009)*, Montreal, Quebec, Canada, 14–18 June 2009. – ACM International Conference Proceeding Series, 2009. – P. 41-48. URL: https://doi.org/10.1145/1553374.1553380.

10. Mistral AI. Announcing Mistral 7B. 2023. URL: https://mistral.ai/news/announcing-mistral-7b.

11. Microsoft Research. Phi-2: The surprising power of small language models. 2023. URL: https://surl.li/rftczk

12. E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu and others. LoRA: Low-Rank Adaptation of Large Language Models. 2021. URL: https://arxiv.org/abs/2106.09685.

13. Raschka S. Practical Tips for Finetuning LLMs Using LoRA (Low-Rank Adaptation). 2023. URL: https://surl.li/pwzmdd

14. Unite.AI. Complete beginner's guide to Hugging Face LLM tools. 2023. – URL: https://surl.li/xshhpt

Dadyverin V.V., Bisikalo O.V. ANALYSIS AND IMPLEMENTATION OF IDEAS FOR TEACHING LARGE LANGUAGE MODELS BY ANALOGY WITH CHILDREN'S COGNITIVE DEVELOPMENT

The study is devoted to the problem of not enough efficiency of large language models in solving mathematical problems, sequential multi-step analysis and reasoning about possible solutions. The article reveals the imperfection of modern LLMs in the issues of sequential thinking, despite their ability to generate natural and meaningful text, they still remain significantly limited in solving problems that require logical reasoning or justification. It was found that such limitations are often a consequence of the structure of the learning process of the language model, because it has nothing to do with the processes that occur during the formation of cognitive and mental models of a person.

This work considers the approach of learning by similarity to human cognitive development, which is based on a combination of the following concepts: Curriculum learning – learning with a gradual increase in the complexity of tasks, Scaffolding – segmented learning with a decrease in the level of care after mastering a

certain level of knowledge, and progressive strengthening of mental structures by creating generalized patterns for solving problems.

The main attention is focused on the implementation of the learning process with a gradual complication of the input material – from basic arithmetic equations to large complex logical-verbal problems. The effectiveness of this approach is evaluated on the example of Phi-2 and Mistral 7B models, trained in two modes: standard random learning and the proposed supervised learning.

The results of the study prove that the proposed cognitive learning approach provides better results when solving basic tasks and complex multi-component tasks that require intermediate generalized knowledge. The analysis of similar studies confirms the promising development of training with imitation of cognitive development to increase the efficiency of processing input data under conditions of a limited amount of training information.

Key words: *large language model, cognitive development, adaptive learning, Mistral 7B, Phi-2, curriculum learning, scaffolding.*

Дата надходження статті: 26.07.2025
Дата прийняття статті: 04.08.2025
Опубліковано: 27.10.2025