

НОВИЙ ПІДХІД КООПЕРАЦІЇ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ТА ТЕХНІК КОНСТРУЮВАННЯ ПІДКАЗОК ДО СНАТ-GPT ДЛЯ АВТОМАТИЧНОГО РОЗВ'ЯЗАННЯ ПРИРОДНОМОВНИХ ЗАДАЧ З МАТЕМАТИКИ

Вінницький національний технічний університет

Анотація

Досліджено проблему розв'язання математичних задач, сформульованих природною мовою, за допомогою великих мовних моделей (LLM). Проаналізовано та порівняно ефективність існуючих підходів. Розроблено новий підхід для оптимізації підказок на основі навчання з підкріпленням, який використовує динамічну побудову підказок та комбінування відомих технік "prompt engineering", в залежності від категорії задачі. Наведено приклад, який довів ефективність запропонованого підходу на практиці.

Ключові слова: великі мовні моделі, кооперація моделей, конструювання підказок, оброблення природної мови, математичні задачі

Abstract

The issue of solving mathematical problems formulated in natural language using large language models (LLM) was investigated. The efficiency of existing approaches was analyzed and compared. A new approach for optimizing prompts based on reinforcement learning was developed, which uses dynamic construction of prompts and combines known "prompt engineering" techniques, depending on the category of the problem. An example that proved the effectiveness of the proposed approach in practice is provided.

Keywords: Large Language Models, models cooperation, prompt engineering, natural language processing, mathematical problems

Вступ

Стрімкий розвиток інформаційних технологій та штучного інтелекту за останні роки суттєво трансформував багато сфер людської діяльності та відкрив нові горизонти для розв'язання складних задач, які раніше вважались недосяжними для автоматизації.

Великі мовні моделі (англ. Large Language Models - LLM) - мовні моделі з великою кількістю параметрів, що переднавчені на великому обсязі текстів для моделювання контекстуальної семантики тексту [1], дозволяють за допомогою запиту, сформульованого природною мовою (так звана "підказка", від англ. "prompt"), виконувати складні операції та перетворення тексту. Проте, великі мовні моделі не позбавлені обмежень і недоліків, що ускладнює загальний підхід до розв'язання задач і створює потребу у розробленні спеціальних підходів для покращення якості згенерованих моделлю результатів. Це стосується, зокрема, розв'язання математичних задач, сформульованих природною мовою, за допомогою LLM.

Метою даного дослідження є розроблення підходу для покращення ефективності використання LLM для розв'язання математичних задач, сформульованих природною мовою.

Результати дослідження

Розв'язання математичних задач за допомогою LLM є важливою задачею, оскільки може значно спростити процес навчання та наукових досліджень, допомогти в автоматизації аналізу даних, підвищити ефективність вирішення складних проблем, зробити математичні знання більш доступними для широкого кола людей тощо.

Аналіз сучасних підходів до розв'язання математичних задач за допомогою LLM показує, що, незважаючи на обмеження моделей, такі як неточності у розумінні контексту та обмеженість знань тренувальними даними, існують підходи, які можуть покращити результативність цих моделей [2].

Серед них можна виділити наступні популярні техніки, які можна віднести до категорії технік конструювання підказок (англ. “prompt engineering”):

1. Chain-of-Thought (CoT) [2]: за цією технікою підказка формулюється таким чином, що спонукає LLM не тільки до розв’язання основної задачі, але також до декомпозиції її на підзадачі і - на окреме виконання кожної з підзадач, утворюючи тим самим “ланцюг думок” (послідовність підказок). Часто для цього слід вказати LLM приклад правильного “ланцюга думок”, щоб мінімізувати логічні помилки у розв’язку, тоді такий підхід називається few-shot CoT. Якщо приклади в підказці не наведені, то підхід має назву zero-shot CoT.
2. Program-Aided Language (PAL) Models [3]: в цьому підході LLM генерує не відповідь на сам запит, а - програму (частіше, мовою Python), яка вирішує поставлену задачу. Це дозволяє уникнути помилок в обчисленнях або складних формальних процедурах, які простіше виконуються безпосередньо комп’ютером, ніж LLM. В поєднанні з технікою Chain-of-Thought, PAL дозволяє генерувати програмний код для окремих підзадач, на які була декомпована основна задача.

Пропонується загальний підхід, розроблений на основі ідей машинного навчання в поєднанні з техніками конструювання підказок, суть якого полягає в тому, щоб навчити просту модель навчання з підкріпленням підбирати ефективні підказки та використовувати CoT або PAL, в залежності від категорії, до якої можна віднести задачу. Процес навчання такої моделі використовує кооперацію великих мовних моделей для виділення ознак та перевірки правильності згенерованої на тренувальному датасеті відповіді. На рисунку 1 зображена спрощена схема роботи розробленого підходу.

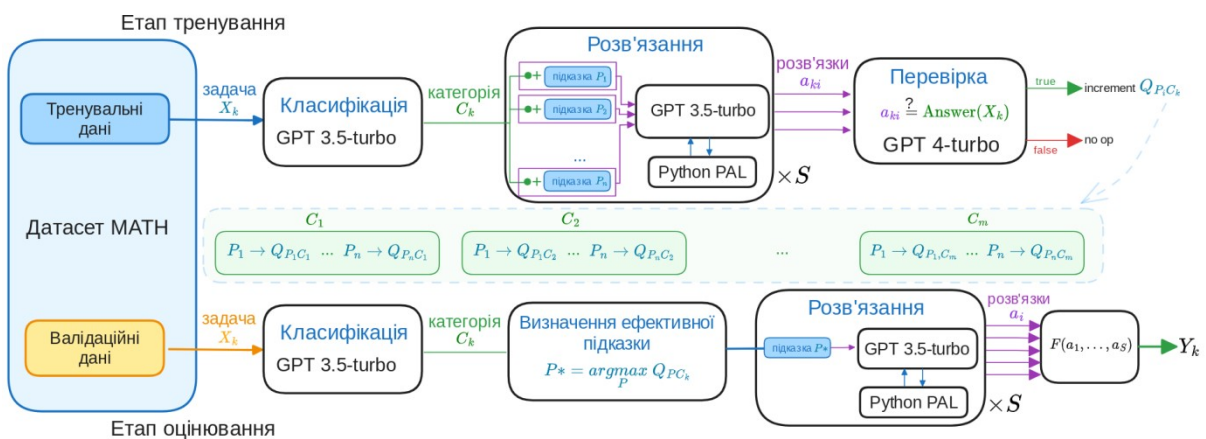


Рисунок 1 – Розроблений підхід для розв’язання математичних задач

На рис. 1 використано такі позначення: X_k — текстове формулювання чергової задачі з датасету; C_k — категорія, до якої можна віднести задачу k (наприклад, “лінійні рівняння”, “теорія ймовірності” тощо); P_i — i -й шаблон підказки для моделі; S - кількість повторів операції розв’язання на одну задачу (дозволяє збільшити консистентність генерування правильних розв’язків); a_{ki} — розв’язок задачі k , згенерований LLM за підказкою i ; $Answer(X_k)$ — правильний розв’язок задачі k з тренувального датасету; Q_{PC} — таблична функція, значенням якої є кількість правильних розв’язків, згенерованих за підказкою P для категорії задач C ; $P^* = \operatorname{argmax}_P(Q_{PC})$ — підказка, що має найбільшу кількість правильних розв’язків для категорії задач C ; $F(a_1, \dots, a_S)$ — функція вибору остаточного розв’язку з набору розв’язків a_1, \dots, a_S , згенерованих моделлю для задачі k за підказкою P^* (наприклад, мода); Y_k — остаточний розв’язок задачі k .

Для перевірки ефективності запропонованого підходу було використано датасет MATH [4], з якого було обрано випадковим чином по 4 задачі з категорій "prealgebra", "algebra", "number_theory", "counting_and_probability", "intermediate_algebra", "precalculus" для навчання і по 20 задач для оцінювання. На рис. 2 наведено приклади успішного (рис. 2а) та неуспішного (рис. 2б) передбачення розв’язку математичної задачі.

1/120. Suppose that we have an 8-sided die with 4 red faces, 3 yellow faces, and a blue face. What is the probability of rolling a yellow face?

GPT SOLUTION: The probability of rolling a yellow face can be calculated by dividing the number of yellow faces by the total number of faces on the die. Number of yellow faces = 3 Total number of faces = 8 Probability of rolling a yellow face = Number of yellow faces / Total number of faces Probability of rolling a yellow face = $3 / 8$ Probability of rolling a yellow face = 0.375 or 37.5%

CORRECT SOLUTION: There are 3 yellow faces and 8 faces total, so the probability of rolling a yellow face is $\frac{3}{8}$.

45/120. How many perfect squares are factors of 180?

GPT SOLUTION: 18

CORRECT SOLUTION: The prime factorization of 180 is $2^2 \cdot 3^2 \cdot 5$. An integer is a divisor of 180 if and only if each exponent in its prime factorization is less than or equal to the corresponding exponent in the prime factorization of 180. An integer is a perfect square if and only if every exponent in its prime factorization is even. Therefore, to form the prime factorization of a perfect square divisor of 180, we may take either 0 or 2 as the exponent of 2 and we may take either 0 or 2 as the exponent of 3. Therefore, there are $\boxed{4}$ perfect square divisors of 180: $2^0 \cdot 3^0$, $2^0 \cdot 3^2$, $2^2 \cdot 3^0$, and $2^2 \cdot 3^2$.

а)

б)

Рисунок 2 – Приклади розв’язання математичних задач: а) успішний; б) неуспішний

Проведено 120 таких оцінювань, за результатами яких визначено кількість правильних передбачень як оцінка ефективності запропонованого підходу (таблиці 1).

Таблиця 1. Частка правильних відповідей на обрану підмножину задач з датасета [4] з використанням різних підходів

| Без використання технік “prompt engineering” | zero-shot CoT | PAL | Розроблений підхід |
|--|---------------|-----|--------------------|
| 43% | 41% | 45% | 55% |

Висновки

Показано актуальність проблеми розв’язання математичних задач, сформульованих природною мовою, за допомогою великих мовних моделей. Проведено дослідження ефективності існуючих підходів покращення ефективності LLM для розв’язання таких задач. Запропоновано підхід для динамічної побудови підказок та автоматичного комбінування існуючих підходів prompt engineering в залежності від типу конкретної задачі, що дозволяє досягти кращої точності розв’язання математичних задач.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, “Survey on Large Language Model (LLM) Security and Privacy: The Good, the Bad, and the Ugly”, arXiv [cs.CR]. 2024.
2. J. Wei et al., “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”, arXiv [cs.CL]. 2023.
3. L. Gao et al., “PAL: Program-aided Language Models”, arXiv [cs.CL]. 2023.
4. D. Hendrycks et al., “Measuring Mathematical Problem Solving With the MATH Dataset”, arXiv preprint arXiv:2103.03874, 2021.

Варер Борис Юхимович — аспірант кафедри системного аналізу та інформаційних технологій; e-mail: androbor17@gmail.com;

Мокін Віталій Борисович — д-р техн. наук, професор, завідувач кафедри системного аналізу та інформаційних технологій; e-mail: vbmokin@vntu.edu.ua;

Varer Borys Yu. — Post-graduate student of the Chair of System Analysis and Information Technology, e-mail: androbor17@gmail.com;

Mokin Vitalii B. — Dr. Sc. (Eng.), Professor, Head of the Chair of System Analysis and Information Technology, e-mail: vbmokin@vntu.edu.ua;