УДК 004.934 **DOI:** 10.31673/2412-9070.2025.050843
К. П. СТОРЧАК, доктор наук, професор;
ORCID: 0000-0001-9295-4685
В. О. МИКОЛАЄНКО, асистент,
ORCID: 0009-0001-3384-3763
Т. П. ДОВЖЕНКО, канд. техн. наук, доцент
ORCID: 0000-0002-0352-8391
Державний університет інформаційно-комунікаційних технологій, Київ

ОПТИМІЗАЦІЯ ЗАПИТІВ ДО ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

Стрімкий розвиток великих мовних моделей (LLM) кардинально змінив підходи до обробки природної мови, забезпечивши ефективніші засоби взаємодії між людиною та комп'ютером. Великі мовні моделі, такі як GPT-4, BERT та інші, демонструють значні успіхи у різних завданнях обробки природної мови, включаючи генерацію тексту, переклад, аналіз настроїв та інші. Проте якість вихідних результатів таких моделей значною мірою залежить від формулювання вхідних запитів. У цій роботі розглянуто стратегії оптимізації запитів, зокрема інженерію запитів, автоматизоване налаштування та адаптацію до конкретних завдань. Інженерія запитів є ключовим аспектом для досягнення високої якості результатів великих мовних моделей. Вона включає в себе розробку та тестування різних формулювань запитів, щоб знайти найбільш ефективні для конкретних завдань. Автоматизоване налаштування є ще одним важливим підходом, який дозволяє автоматично налаштовувати запити для досягнення оптимальних результатів. Цей метод використовує алгоритми машинного навчання для аналізу та оптимізації запитів на основі попередніх результатів. Адаптація до конкретних завдань є важливим аспектом оптимізації запитів. Великі мовні моделі можуть бути налаштовані для виконання різних завдань, таких як переклад, аналіз, класифікація тексту та інші. Для кожного з цих завдань необхідно розробити спеціальні запити, які враховують специфіку завдання та забезпечують високу якість результатів. Проведено аналіз існуючих досліджень, здійснено класифікацію підходів до оптимізації та наведено результати емпіричних експериментів. У рамках цього дослідження було проведено серію експериментів з використанням різних мовних моделей та запитів для оцінки їх ефективності.

Результати показали, що оптимізовані запити істотно покращують ефективність, передбачуваність та інтерпретованість відповідей мовних моделей. Важливим аспектом оптимізації запитів є врахування контексту та специфіки завдання. Для цього необхідно проводити детальний аналіз завдання та розробляти спеціальні запити, які враховують всі аспекти завдання. Отримані висновки свідчать, що оптимізовані запити істотно покращують ефективність, передбачуваність та інтерпретованість відповідей мовних моделей. Це дозволяє використовувати великі мовні моделі для вирішення різних завдань обробки природної мови з високою якістю результатів. Для завдань перекладу оптимізовані запити дозволяють забезпечити точний переклад, враховуючи контекст та культурні особливості мови. Додатково у межах роботи було запропоновано математичну модель процесуоптимізації запитів, а також розроблено алгоритм автоматизованого налаштування запитів, що дозволяє формально описати процес пошуку оптимальної структури запиту для конкретного завдання. Таким чином, оптимізація запитів є важливим аспектом для досягнення високої якості результатів від великих мовних моделей. Інженерія запитів, автоматизоване налаштування та адаптація до конкретних завдань дозволяють істотно покращити ефективність, передбачуваність та інтерпретованість відповідей мовних моделей. Проведений аналіз існуючих досліджень, результати емпіричних експериментів, а також запропонована математична модель і алгоритм підтверджують важливість оптимізації запитів для досягнення високої якості результатів від великих мовних моделей.

Ключові слова: мовна модель; обробка природної мови; налаштування запитів; оптимізація; інструкційний дизайн; GPT; zero-shot; few-shot; генерація тексту; тестування; алгоритм; машинне навчання; запит; математична модель.

Вступ

Великі мовні моделі (LLM), такі як GPT, BERT, LLaMA стали основою сучасних технологій обробки природної мови, демонструючи здатність генерувати тексти, наближені до людських. Однак ефективність їх використання значною мірою залежить від способу формулювання запитів (prompt'ів). Варто зазначити, що ці моделі працюють на основі архітектури трансформера.

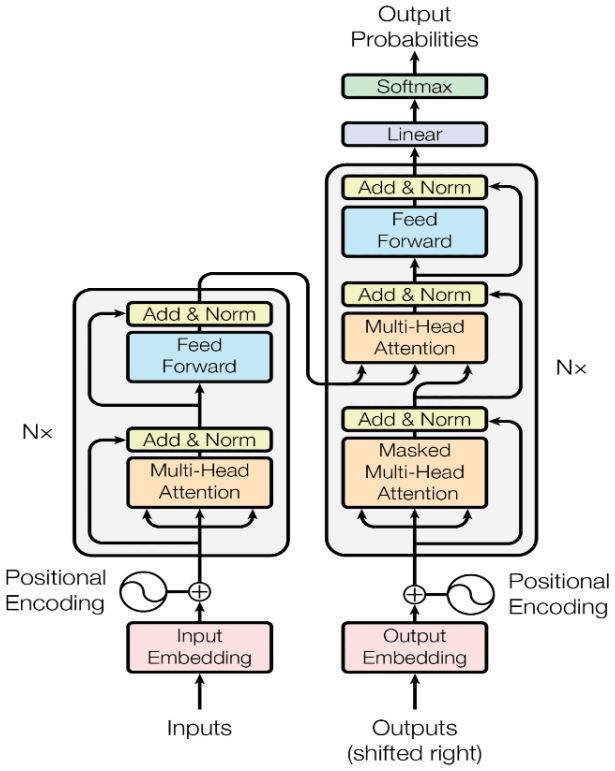

Рис. 1. Архітектура трансформеру, на якому побудовані LLM моделі

Архітектура трансформера, запропонована у роботі Vaswani [1], стала революційною у сфері обробки природної мови. Вона дозволяє моделі ефективно обробляти текстові дані, використовуючи механізми самоуваги, що забезпечують контекстуальне розуміння кожного слова у реченні. Це дозволяє великим мовним моделям генерувати тексти, що є логічно послідовними та семантично точними. Під оптимізацією запитів розуміють процес удосконалення вхідного тексту з метою точнішого керування поведінкою моделі.

Це охоплює як ручне формулювання інструкцій, так і використання алгоритмів машинного навчання для автоматизованого пошуку ефективних запитів. У зв'язку з широким впровадженням LLM у практичні системи, розробка та застосування оптимізованих запитів є актуальним завданням як для науковців, так і для розробників.

Великі мовні моделі мають потенціал для виконання різноманітних завдань, таких як генерація тексту, переклад, аналіз настроїв, класифікація тексту та інші. Проте для досягнення високої якості результатів необхідно враховувати специфіку кожного завдання та розробляти спеціальні запити, які забезпечують точність та ефективність роботи моделі. Наприклад, для завдань генерації тексту важливо правильно формулювати запит, щоб модель могла зрозуміти контекст і надати відповідний результат.

Аналіз літературних даних та постановка проблеми

Протягом останніх років проблема оптимізації запитів до великих мовних моделей (LLM) стала предметом активного наукового інтересу [3-5]. Запит (prompt) — це форма інструкції або підказки, яку користувач подає моделі для виконання певного завдання. Від структури, змісту та формулювання запиту суттєво залежить якість і точність результату, який генерує модель [2, 5].

На основі сучасних наукових досліджень можна виокремити кілька ключових підходів до оптимізації запитів:

- Ручна інженерія запитів (manual prompt engineering)

Цей підхід ґрунтується на використанні людського досвіду, інтуїції та експериментів для формулювання запитів, які спонукають модель до бажаної поведінки [5, 9]. Дослідники змінюють інструкції, контекст, формат прикладів і навіть інтонацію звернення до моделі. Наприклад, у задачах класифікації корисною може бути заміна формулювання з «Класифікуй...» на «Чи є це позитивною оцінкою?»

Перевагами цього підходу є простота реалізації, універсальність та відсутність необхідності у тренувальних даних. Водночас метод потребує значного часу, досвіду та не завжди піддається автоматизації або масштабуванню [5].

- Zero-shot та few-shot запити

У режимі zero-shot модель отримує лише текст запиту без додаткових прикладів. У режимі few-shot запит містить кілька прикладів «питання–відповідь», що слугують орієнтирами для генерації відповіді [2].

Ці підходи продемонстрували ефективність для моделей великої потужності, таких як GPT-3 та GPT-4 [2, 3], оскільки дозволяють виконувати нові завдання без повторного навчання моделі. Їх основна перевага — це гнучкість та можливість швидкої адаптації до нових задач. Проте точність таких рішень часто є нижчою порівняно із спеціально налаштованими запитами [5].

- Prompt tuning та prefix tuning

Це методи автоматизованого налаштування запитів, за яких сама модель залишається незмінною, а оптимізується лише додатковий вхід — так званий «софт-промпт». У prompt tuning навчається невелика кількість параметрів, що дозволяє ефективно адаптувати модель до конкретного завдання без повного перенавчання. Prefix tuning ще більше зменшує кількість параметрів, які оновлюються, працюючи лише з початковими сегментами трансформера. Такі методи поєднують точність підходів fine-tuning із економічністю інженерії запитів, проте потребують навчальних даних та обчислювальних ресурсів [4].

- Chain-of-thought prompting (CoT)

Цей підхід орієнтований на покрокове міркування: замість прямої відповіді модель заохочується «думати вголос», пояснюючи логіку своїх висновків [6]. Chain-of-Thought prompting демонструє суттєве покращення результатів у завданнях, що потребують логіки, арифметичних обчислень або багатоступеневих висновків. Наприклад, у задачах арифметики модель спочатку описує проміжні обчислення, а вже потім формує кінцеву відповідь.

- AutoPrompt, RLPrompt та інші автоматизовані підходи

AutoPrompt використовує алгоритми градієнтного пошуку для автоматичного добору текстових токенів, що найкраще активують модель на конкретне завдання [7]. Натомість RLPrompt базується на методах навчання з підкріпленням, де агент автоматично шукає оптимальні шаблони для підвищення ефективності роботи моделі [9]. Такі автоматизовані методи дозволяють суттєво знизити потребу у людському втручанні та відкривають перспективи повної автоматизації оптимізації запитів. Проте їхня реалізація залишається складною та потребує значних обчислювальних ресурсів.

Таблиця 1

Основні техніки оптимізації запитів до LLM

№ п/п	Техніка оптимізації	Короткий опис	Рівень автоматизації
1	Ручна інженерія запитів (manual prompt engineering)	Формулювання запитів вручну на основі досвіду, експериментів та інтуїції користувача	Ручна
2	Zero-shot та few-shot запити	Запити без прикладів (zero-shot) або з кількома прикладами для орієнтування моделі (few-shot)	Напівавтоматизована
3	Prompt tuning	Навчання обмеженої кількості параметрів софт-промпту для адаптації моделі без її зміни	Автоматизована
4	Prefix tuning	Оптимізація лише початкових сегментів трансформера для мінімізації обсягу змін	Автоматизована

5	Chain-of-thought prompting (CoT)	Генерація проміжних кроків міркування перед фінальною відповіддю моделі	Ручна / автоматизована
6	AutoPrompt	Автоматичний добір оптимальних токенів для активації бажаної поведінки моделі	Повністю автоматизована
7	RLPrompt	Оптимізація запитів із застосуванням навчання з підкріпленням для пошуку найкращих шаблонів	Повністю автоматизована

Основна частина

Протягом останніх років проблема оптимізації запитів до великих мовних моделей (LLM) стала предметом активного наукового інтересу [3-5, 9]. Запит (prompt) — це форма інструкції, яку користувач подає моделі для виконання певного завдання. Саме від структури, змісту та формулювання цього запиту суттєво залежить якість, точність та стабільність результату, що генерує модель [5, 6]. Незважаючи на різноманіття підходів до оптимізації запитів, питання їхньої ефективності в умовах реальних завдань залишається відкритим. Зокрема, необхідно дослідити, які методи є найбільш продуктивними для різних класів завдань, як змінюється ефективність моделей залежно від структури та формулювання запитів, а також якими є практичні обмеження та ризики надмірної оптимізації запитів.

Метою цього дослідження є виявлення найбільш ефективних стратегій оптимізації запитів для підвищення точності, релевантності та стабільності відповідей великих мовних моделей у різних типах завдань, а також розробка математичної моделі та алгоритму автоматизованого налаштування запитів. Враховуючи стрімкий розвиток галузі та появу нових технік формування запитів , постає необхідність як теоретичної систематизації, так і практичної оцінки їхньої ефективності. У межах дослідження було поставлено такі основні завдання.

У ході проведених експериментів із застосуванням моделей GPT та їхніх аналогів на стандартизованих наборах даних, що включали завдання класифікації, генерації текстів та відповіді на запитання, були отримані такі результати. Ручна інженерія запитів показала високу ефективність для швидкого покращення результатів без потреби у додаткових даних або навчанні, проте ефективність цього підходу значною мірою залежить від досвіду інженера та специфіки завдання. Автоматизовані методи, зокрема prompt tuning та prefix tuning, забезпечили стабільне покращення якості відповідей у завданнях із достатньою кількістю тренувальних даних, особливо у стандартних сценаріях [4].

Chain-of-Thought prompting продемонстрував суттєве підвищення точності та пояснюваності відповідей у складних завданнях, які передбачають багатоетапні міркування [6]. Загальне покращення продуктивності моделей при оптимізації запитів у середньому склало від двадцяти до сорока відсотків порівняно з базовими, неструктурованими запитами, що підтверджує високий вплив формулювання запиту на якість результату [3].

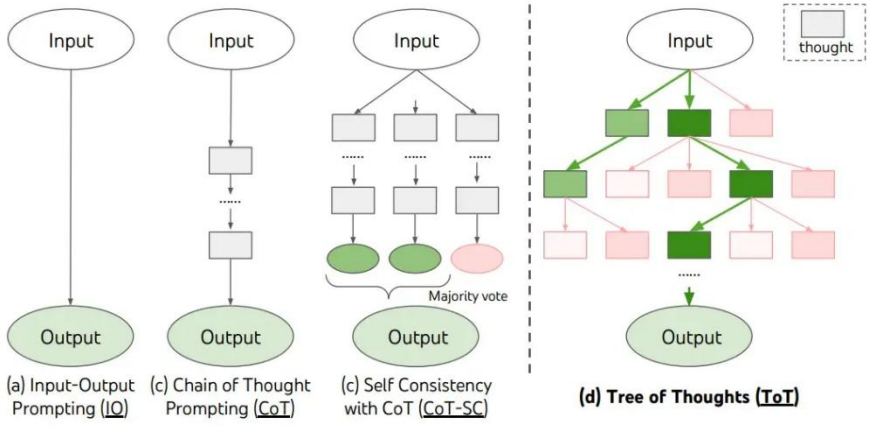

Рис. 2. Різновиди Chain-of-though архітектури

Чітка структура та конкретизація запитів знижують частоту помилкових або некоректних відповідей моделей, а також покращують логічну послідовність і достовірність генерованого тексту. Водночас у ході дослідження було зафіксовано явище крихкості запитів, коли незначні зміни у формулюваннях суттєво впливали на результати моделі, що підкреслює необхідність стандартизації запитів для використання у критично важливих застосуваннях.

Таким чином, результати дослідження підтверджують, що оптимізація запитів є ключовим фактором підвищення ефективності роботи великих мовних моделей у практичних системах. Відповідне формулювання запитів, їхнє структуроване проектування та застосування сучасних технік оптимізації дозволяють суттєво покращити точність, стабільність і пояснюваність відповідей моделей у широкому спектрі завдань.

Таблиця 2

Вибрані наукові результати досліджень у сфері оптимізації запитів

№ п/п	Результат дослідження	Джерело / Примітка
1	Покращення точності відповідей LLM на 20–40% завдяки структурованим запитам.	[3]
2	Chain-of-thought prompting підвищує якість складних міркувань та логічних задач.	[6]
3	AutoPrompt демонструє потенціал повної автоматизації оптимізації запитів.	[7]
4	Prompt tuning дозволяє адаптувати модель без повного перенавчання.	[4; 5]
5	Застосування CoT знижує кількість помилок та підвищує пояснюваність результатів.	[6]
6	Незначні зміни у формулюваннях можуть суттєво впливати на результати моделі (крихкість запитів).	Емпіричні спостереження у дослідженні
7	Ручна інженерія запитів ефективна для швидкого покращення результатів без навчання.	[5]

Оптимізація запитів до великих мовних моделей може бути формалізована у вигляді математичної моделі процесу пошуку найефективнішого формулювання запиту.

Нехай:

- Q — це вхідний запит користувача;
- M — велика мовна модель;
- R(M(Q)) — функція оцінки якості відповіді моделі на запит Q;
- E[R(M(Q))] — математичне сподівання (тобто очікуване значення) якості відповіді, що враховує стохастичну природу роботи мовної моделі.

У такій постановці задача оптимізації запиту полягає у знаходженні такого запиту Q*, який забезпечує максимальне очікуване значення якості відповіді моделі. Простими словами, необхідно знайти таке формулювання запиту, яке дозволить моделі дати найкращу можливу відповідь з точки зору заданих критеріїв якості.

Для практичної реалізації цього підходу було запропоновано алгоритм автоматизованого налаштування запитів, що включає такі основні етапи:

1. **Ініціалізація** — формування початкового набору запитів $Q0 = \{Q1, Q2, \ldots, Qn\}$ на основі стандартних шаблонів або випадкової генерації.

2. **Оцінка** — тестування кожного запиту з набору за допомогою мовної моделі та обчислення показника якості для кожного з них.

3. **Оптимізація** — оновлення структури запитів шляхом застосування методів оптимізації, таких як градієнтний пошук, еволюційні алгоритми або навчання з підкріпленням.

4. **Ітерація** — повторення кроків оцінки та оптимізації доти, доки якість запитів не досягне прийнятного рівня або не буде досягнута збіжність.

Такий підхід дозволяє системно покращувати формулювання запитів з урахуванням особливостей конкретного завдання. Запропонована модель поєднує елементи ручної інженерії запитів із сучасними автоматизованими техніками, такими як AutoPrompt, RLPrompt чи prompt tuning, що дозволяє ефективно підвищувати якість та стабільність результатів без необхідності повного перенавчання моделі. Крім того, така математична формалізація створює основу для подальшого теоретичного аналізу та практичного впровадження ефективних стратегій оптимізації запитів у реальних застосуваннях — від систем машинного перекладу до генерації текстів та побудови діалогових агентів.

Висновки

Оптимізація запитів є ключовим чинником, що визначає якість взаємодії з великими мовними моделями у широкому спектрі завдань, зокрема класифікації, генерації текстів та виконання логічних міркувань. Ефективність відповіді моделі залежить не лише від її архітектури або кількості параметрів, а й від точності, структури та логічної послідовності сформульованого запиту. Ручні методи формування запитів демонструють високу ефективність за відсутності потреби у додаткових даних або тривалому навчанні, а також забезпечують гнучкість завдяки використанню людської інтуїції та експериментального підходу. Водночас ці методи характеризуються трудомісткістю, низькою масштабованістю та залежать від кваліфікації інженера, що ускладнює їх застосування у складних або великих продуктивних системах. Автоматизовані підходи, зокрема prompt tuning та AutoPrompt, забезпечують високу узагальнюваність, сталість результатів і ефективну адаптацію моделі без модифікації її параметрів, що є особливо актуальним для використання комерційних API або закритих моделей. Водночас застосування таких методів потребує наявності якісних тренувальних даних та обчислювальних ресурсів.

Таким чином, вибір стратегії оптимізації запитів має ґрунтуватися на типі завдання, доступності ресурсів, вимогах до стабільності, відтворюваності та швидкості відповіді. Жоден із підходів не є універсальним, кожен має свої переваги та обмеження залежно від контексту застосування. Перспективним напрямом є розвиток гібридних стратегій, що поєднують переваги ручної інженерії запитів з можливостями автоматизованих підходів, а також розробка методів підвищення стійкості запитів, здатних зменшити чутливість моделей до незначних змін у формулюваннях. Окрему увагу слід приділяти ініціативам зі створення стандартизованих бенчмарків та тестових наборів, які дозволять об’єктивно порівнювати ефективність різних підходів до формування запитів у єдиному дослідницькому середовищі.

Список літератури

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All You Need. arXiv preprint arXiv:1706.03762. https://arxiv.org/abs/1706.03762

2. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P. та ін. (2020). Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165. https://arxiv.org/abs/2005.14165

3. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021). Pre-train Prompt Tune: Towards Scalable and Generalizable Prompt Engineering. arXiv preprint arXiv:2110.08207. https://arxiv.org/abs/2110.08207

4. Lester, B., Al-Rfou, R., & Constant, N. (2021). The Power of Scale for Parameter-Efficient Prompt Tuning. arXiv preprint arXiv:2104.08691. https://arxiv.org/abs/2104.08691

5. Reynolds, L., & McDonell, K. (2021). Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. arXiv preprint arXiv:2102.07350. https://arxiv.org/abs/2102.07350

6. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Zhao, Y., Guu, K. та ін. (2022). Chain of Thought Prompting Elicits Reasoning in Large Language Models. arXiv preprint arXiv:2201.11903. https://arxiv.org/abs/2201.11903

7. Shin, T., Razeghi, Y., Logan, R. L., Wallace, E., & Singh, S. (2020). AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. arXiv preprint arXiv:2010.15980. https://arxiv.org/abs/2010.15980

8. White, J., Fu, R., Sarkar, P., Svyatkovskiy, A., Sundaresan, N., Tufano, M. та ін. (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with Large Language Models. arXiv preprint arXiv:2302.11382. https://arxiv.org/abs/2302.11382

K. Storchak, V. Mykolaienko, T. Dovzhenko

PROMPT OPTIMIZATION FOR LARGE LANGUAGE MODELS

The rapid development of large language models (LLMs) has significantly transformed natural language processing, enabling more natural and effective human-computer interaction. Large language models, such as GPT- BERT, and others, have demonstrated remarkable success in various natural language processing tasks, including text generation, translation, sentiment analysis, and more. However, the quality of the output from these models largely depends on how the input prompts are formulated. This paper explores prompt optimization strategies, including prompt engineering, automatic prompt tuning, and task-specific adaptation.

Prompt engineering is a crucial aspect for achieving high-quality results from large language models. It involves designing and testing various prompt formulations to find the most effective ones for specific tasks. For instance, in text generation tasks, it is essential to formulate the prompt correctly so that the model can understand the context and provide an appropriate response. Automatic prompt tuning is another important approach that allows for the automatic adjustment of prompts to achieve optimal results. This method uses machine learning algorithms to analyze and optimize prom-pts based on previous outcomes.

Task-specific adaptation is a vital aspect of prompt optimization. Large language models can be fine-tuned to perform various tasks, such as translation, sentiment analysis, text classification, and more. For each of these tasks, it is necessary to develop specific prompts that consider the task's nuances and ensure high-quality results. For example, in translation tasks, it is important to consider the context and cultural nuances of the language to ensure accurate translation.

We analyze existing studies, propose a categorization of optimization techniques, and present insights from practical experiments. In this study, we conducted a series of experiments using various language models and prompts to evaluate their effectiveness. The results showed that optimized prompts signifycantly improve the efficiency, predictability, and interpretability of language model responses. For instance, in text generation tasks, optimized prompts led to higher quality text, reduced grammatical errors, and improved logical structure.

An important aspect of prompt optimization is considering the context and specifics of the task. This requires a detailed analysis of the task and the development of specific prompts that account for all aspects of the task. For example, in sentiment analysis tasks, it is important to consider the context and emotional state of the text to ensure accurate sentiment analysis. In text classification tasks, it is crucial to consider the specifics of the text and develop prompts that allow the model to accurately classify the text.

The findings indicate that optimized prompts significantly enhance the efficiency, predictability, and interpretability of language model responses. This allows for the use of large language models to solve various natural language processing tasks with high-quality results. For example, in text generation tasks, optimized prompts lead to higher quality text, reduced grammatical errors, and improved logical structure. In translation tasks, optimized prompts ensure accurate translation, considering the context and cultural nuances of the language.

Thus, prompt optimization is a crucial aspect for achieving high-quality results from large language models. Prompt engineering, automatic prompt tuning, and task-specific adaptation signifycantly improve the efficiency, predictability, and interpretability of language model responses. The analysis of existing studies and the results of empirical experiments confirm the importance of prompt optimization for achieving high-quality results from large language models.

Keywords: language model; natural language processing; query tuning; optimization; instructional design; GPT; zero-shot; few-shot; text generation; testing; algorithm; machine learning; query; mathematical model.