



DOI 10.28925/2663-4023.2026.32.1063

УДК 004.056.5:004.75+004.738.5

Онищук Оксана Олександрівна

кандидат технічних наук, доцент

Волинський національний університет, Луцьк, Україна

ORCID: 0000-0002-8342-3011

oksanaoo2024@gmail.com

ПОРІВНЯННЯ TELEGRAM- БОТА ДЛЯ ГЕНЕРАЦІЇ ТЕКСТУ, СТВОРЕНОГО НА ОСНОВІ OPENAI і GEMINAI

Анотація. У роботі розглянуто проблему створення та порівняльного аналізу сучасних генеративних мовних моделей на прикладі OpenAI GPT та Gemini 2.5 Pro, інтегрованих у Telegram-бот для автоматичної генерації текстів. Актуальність дослідження зумовлена зростаючою потребою у швидких, гнучких та якісних інструментах створення контенту, здатних працювати у режимі реального часу. У межах роботи було сформульовано вимоги до програмного засобу, розроблено архітектуру Telegram-бота, визначено принципи його роботи та реалізовано взаємодію з API обох моделей. Особливу увагу приділено тестуванню якості згенерованих текстів, швидкодії, стабільності роботи моделей та їх поведінці під навантаженням. Під час проведення експериментів встановлено, що Gemini 2.5 Pro демонструє значно вищу швидкість генерації та точність інтерпретації запитів, тоді як OpenAI GPT переважає у творчих завданнях, стилістичній природності та гнучкості формулювань. Обидві моделі показали високу здатність до роботи з багатьма жанрами текстів, проте відмінні за структурністю, логічністю та якістю підтримки контексту. Результати дослідження свідчать, що жодна модель не має абсолютної переваги: вибір залежить від конкретної задачі та вимог до стилю, точності чи швидкості відповіді. Розроблений Telegram-бот підтвердив ефективність інтеграції генеративних моделей у практичні системи та продемонстрував можливість адаптації до різних сценаріїв використання. Отримані висновки можуть бути використані для подальшої оптимізації генеративних систем, вибору моделей для конкретних застосунків та розробки інтелектуальних сервісів на основі штучного інтелекту.

Ключові слова: чат-бот; генеративні моделі; штучний інтелект; OpenAI GPT; Gemini 2.5 Pro; трансформери; Telegram-бот; генерація текстів; обробка природної мови; порівняльний аналіз.

ВСТУП

Сучасні цифрові технології стрімко змінюють способи взаємодії людини з інформаційними системами, а одним із найбільш динамічних напрямів розвитку є створення чат-ботів, здатних генерувати тексти. Їх використання сприяє автоматизації процесів, підвищенню ефективності комунікації та формуванню контенту в різних сферах – від освіти й журналістики до бізнесу та державного управління. Швидкий розвиток штучного інтелекту, поява трансформерних моделей і зростання обчислювальних можливостей відкривають нові перспективи побудови систем, здатних створювати зв'язні, логічні та стилістично коректні тексти.

Постановка проблеми. Актуальність дослідження полягає в необхідності теоретичного та практичного обґрунтування процесу створення чат-бота для генерації текстів та визначення його можливостей у сучасних інформаційних екосистемах. Стрімкий розвиток штучного інтелекту та поява високопродуктивних мовних моделей зумовили зростання інтересу до автоматизованої генерації текстів. Проте, попри



активне впровадження генеративних систем у різні сфери, залишається недостатньо дослідженим питання ефективності застосування різних моделей штучного інтелекту в реальних прикладних сервісах, зокрема у чат-ботах. Важливо визначити, яка модель забезпечує кращу якість, швидкодію, варіативність та стабільність відповідей у контексті користувацької взаємодії. Проблема полягає не лише у створенні функціонального інструмента генерації текстів, а й у порівнянні моделей Gemini 2.5 Pro та OpenAI GPT, які мають різні архітектурні та алгоритмічні підходи. Наявність таких відмінностей потребує ґрунтовного аналізу для виявлення оптимального рішення для розгортання застосунків, орієнтованих на реального користувача. Таким чином, постає завдання створення Telegram-бота, здатного працювати з обома моделями, а також проведення їхнього системного порівняння за якісними та кількісними показниками, що дозволить сформулювати об'єктивні висновки щодо їхньої ефективності.

Аналіз останніх досліджень і публікацій. Поставлену проблематику активно досліджують провідні науковці, інженери та дослідницькі групи у сфері штучного інтелекту, машинного навчання та обробки природної мови. Значний внесок у розвиток генеративних систем зробили A. Vaswani, N. Shazeer, N. Parmar та їхні колеги, які у своїй фундаментальній праці «Attention Is All You Need» запропонували архітектуру трансформера – інноваційний підхід, що замінив рекурентні моделі механізмом багатоголосої уваги і забезпечив якісний стрибок у можливостях генерації текстів [1]. Саме трансформери стали базою для створення сучасних мовних моделей, таких як GPT, BERT, T5 та Gemini.

Подальший розвиток трансформерних підходів детально описано у дослідженні C. Raffel, N. Shazeer, A. Roberts та ін., де представлена модель T5 (Text-to-Text Transfer Transformer), яка уніфікувала всі завдання обробки природної мови у формат «текст → текст» і показала високу ефективність у генерації логічних та граматично правильних текстів [2].

Оглядові наукові публікації Min B., Ross H., Sulem E., Roth D., а також огляд Kalyan S. «A Survey of GPT-3 Family...» демонструють надзвичайно швидкі темпи розвитку великих мовних моделей (LLM), зокрема GPT-3, GPT-4 та моделей сімейства Gemini. У цих роботах проаналізовано архітектури, принципи навчання, обмеження та потенціал систем ШІ, що дозволяє оцінити їх можливість розв'язувати широке коло завдань, включаючи генерацію змістовних текстів, ведення діалогу та створення контенту різної складності [3, 4].

Практичний аспект застосування генеративних моделей у чат-ботах розглядають такі ресурси, як платформа Amvera, яка описує інтеграцію GPT-4 Omni у Telegram-ботів, та офіційна документація бібліотеки TeLLMgramBot (PyPI), що пропонує готові інструменти для поєднання Telegram API з мовними моделями [5,6]. Публікації Speka Media також демонструють актуальність використання ШІ в SEO, журналістиці та контент-маркетингу, що підтверджує зростання попиту на автоматизовані системи генерації текстів у практичних сценаріях [7].

Додатково, у дипломних проектах і науково-методичних роботах, представлених у відкритих електронних бібліотеках (зокрема КПІ), висвітлено приклади застосування ChatGPT та подібних моделей у процесах трансформації тексту, покращення контенту та створення навчальних матеріалів [8]. Навчальні курси з машинного навчання, наприклад курс ЗНУ, допомагають систематизувати знання про моделі, алгоритми та методи оптимізації, що формує базу для реалізації власних генеративних систем [9].

Аналіз цих джерел дозволив визначити як теоретичні основи, так і практичні підходи до створення чат-ботів на основі штучного інтелекту. Сукупність отриманих



даних підтверджує актуальність і перспективність використання трансформерних моделей у системах автоматичної генерації текстів, а також дає можливість сформувати методичну базу для розробки та порівняння моделей Gemini та OpenAI у межах даної роботи.

Метою роботи є створення, тестування чат-бота на основі моделей Gemini або OpenAI та здійснення їх порівняльного аналізу.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Теоретичні основи дослідження охоплюють комплекс наукових положень, що визначають принципи створення та функціонування систем генерації текстів на основі штучного інтелекту. Насамперед розглядається класифікація чат-ботів, яка включає шаблонні, статистичні та інтелектуальні моделі. Шаблонні чат-боти працюють на основі наперед визначених сценаріїв та шаблонів, статистичні використовують методи машинного навчання для аналізу мовних закономірностей, а інтелектуальні моделі застосовують глибокі нейронні мережі для розуміння контексту та побудови нових текстів. Така класифікація дозволяє зрозуміти еволюцію технологій і визначити місце сучасних генеративних систем у загальній структурі програмних агентів.

Важливе значення мають моделі природної мови, зокрема RNN, LSTM/GRU та трансформерні архітектури. Рекурентні мережі стали першими ефективними інструментами для роботи з послідовностями, LSTM та GRU розв'язали проблеми довготривалих залежностей, а трансформери, такі як GPT, BERT, T5, зробили можливим глибоке розуміння контексту та створення зв'язаних, логічних текстів високої якості. Саме трансформерні моделі сьогодні є основою найбільш продуктивних систем генерації природної мови.

Теоретичні засади також включають опис алгоритмів генерації тексту, які застосовуються для вибору наступних слів або фраз у процесі створення відповіді. До основних методів належать greedy search, що забезпечує передбачувані результати; beam search, який дозволяє отримувати більш оптимальні та логічні тексти; sampling – метод із випадковим вибором слів, що підвищує різноманітність; а також temperature tuning, який дає змогу регулювати ступінь креативності моделі. Кожен із цих алгоритмів має свої переваги та слугує для різних цілей під час генерації текстів.

Окремий блок теоретичних положень пов'язаний з архітектурою чат-бота, що включає інтерфейс взаємодії, модуль обробки природної мови (NLP), модуль генерації тексту, систему управління діалоговим контекстом та серверну інфраструктуру. Взаємодія цих компонентів забезпечує правильне сприйняття запитів, адекватну інтерпретацію намірів користувача, логічність відповідей та стабільність роботи системи в реальному часі.

Важливо підкреслити міждисциплінарний характер створення чат-ботів, оскільки процес розробки поєднує програмування, лінгвістику, теорію штучного інтелекту, роботу з мовними моделями, а також знання з галузей мережевих технологій і баз даних. Таке поєднання різних напрямів дозволяє сформувати цілісне уявлення про механізми функціонування систем генерації текстів.

У сукупності наведені теоретичні положення становлять фундамент для практичної реалізації чат-бота, визначають технічні вимоги, особливості обробки природної мови та критерії якості згенерованих текстів. Саме ці основи забезпечують основу для створення стабільної, точної та ефективної системи автоматичної генерації текстів.



МЕТОДИКА ДОСЛІДЖЕННЯ

Методика дослідження передбачає послідовне проходження кількох взаємопов'язаних етапів, спрямованих на створення та оцінювання чат-бота для генерації текстів. На початковому етапі проводили аналіз теоретичних джерел, що охоплювали особливості роботи чат-ботів, архітектуру генеративних моделей та принципи обробки природної мови. Це дозволило сформувати наукове підґрунтя для подальшої практичної реалізації та уточнити технічні потреби системи.

Наступним кроком стала постановка задачі й визначення вимог до програмного засобу. Було встановлено функціональні та нефункціональні характеристики, яких повинна дотримуватися система, а також окреслено загальну концепцію роботи чат-бота та його інтеграції з мовними моделями.

Після цього здійснювався вибір інструментальних засобів. Основою розробки стала мова програмування Python, а також бібліотека python-telegram-bot для реалізації взаємодії з Telegram API. Для генерації текстів використовувалися API моделей OpenAI та Gemini 2.5 Pro, що дало змогу порівняти їх продуктивність та якість відповідей.

Практичний етап включав програмну реалізацію Telegram-бота. Було створено обробники повідомлень, які забезпечують прийом і передачу даних між користувачем та мовною моделлю. Визначено логіку діалогу, проведено інтеграцію з генеративними моделями та налаштовано режим polling, що дозволив безперервно отримувати повідомлення з Telegram без підняття веб-сервера.

Подальший етап стосувався тестування та налагодження системи. Спочатку виконувалися модульні перевірки коректності коду та імпорту бібліотек. Потім проводилося практичне тестування в Telegram, під час якого аналізувалися стабільність роботи, швидкість генерації текстів і коректність відповідей. Особливу увагу приділено обробці помилок та перевірці системи в умовах підвищеного навантаження під час стрес-тестування.

Завершальним етапом стала підготовка документації та формування рекомендацій щодо використання програмного засобу. Було описано вимоги до запуску бота, особливості його налаштування, можливості масштабування та напрями подальшого вдосконалення. Такий комплексний підхід забезпечив повне охоплення як теоретичних, так і практичних аспектів дослідження.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

У ході дослідження було розроблено Telegram-бота для генерації текстів та здійснено порівняльний аналіз двох мовних моделей – OpenAI GPT та Gemini 2.5 Pro. Результати охоплюють технічні, функціональні та якісні аспекти роботи створеного чат-бота, а також демонструють сильні та слабкі сторони обох моделей у контексті генерації текстів, швидкодії та стабільності.

На етапі постановки задачі були визначені основні вимоги до програмного засобу, що включали коректне розуміння текстових запитів, здатність створювати змістовні та стилістично різноманітні тексти, а також стабільну роботу в режимі реального часу. Під час реалізації Telegram-бота було створено систему взаємодії з користувачем через polling, налаштовано передачу запитів до мовних моделей, а також реалізовано механізми обробки помилок, що забезпечило стабільний цикл запит → генерація → відповідь.

Файл bot.py є центральним елементом програмної системи, що забезпечує роботу Telegram-бота для генерації текстів. Реалізація починається з підключення необхідних



бібліотек, зокрема `python-telegram-bot` для роботи з Telegram API та офіційної бібліотеки для взаємодії з моделлю Gemini 2.5 Pro. У цьому ж модулі виконуються завантаження API-ключів та створення клієнтів для обох сервісів (лістинг 1).

```
python-telegram-bot==21.4
openai>=1.50
pyTelegramBotAPI
google-generativeai
python-dotenv
requests
from telegram.ext import Updater, MessageHandler, Filters
from google import genai
GEMINI_API_KEY = "AIzaSyAwHZQjX0Nj87BhT0RTP9HEsq9-0wegBag"
TELEGRAM_TOKEN = "7663464326:AAGFzYglEvc0Y32Hf-IVzOaHybCuLs75bec"
client = genai.Client(api_key=GEMINI_API_KEY)
```

Лістинг 1

Особливо важливою частиною реалізації є система обробників подій (handlers). Використовується `MessageHandler`, який реагує на всі текстові повідомлення користувача. Кожне отримане повідомлення передається у функцію, що виконує генерацію тексту за допомогою моделі (лістинг 2).

```
def handle_message(update, context):
    user_text = update.message.text
    response = client.models.generate(
        model="gemini-2.5-pro",
        prompt=user_text
    )
    answer = response.text
    update.message.reply_text(answer)
```

Лістинг 2

Функція генерації тексту працює за стандартною схемою: вона отримує повідомлення від користувача, надсилає запит до моделі Gemini, отримує сформовану відповідь та передає її назад у Telegram через API. Таким чином забезпечується повний цикл взаємодії між користувачем, Telegram і мовною моделлю. Для запуску бота використовується механізм `polling`, що не вимагає окремого сервера чи налаштування веб-куків. Це дозволяє легко запускати програму в локальному середовищі (лістинг 3).

```
updater = Updater(TELEGRAM_TOKEN, use_context=True)
dispatcher = updater.dispatcher
dispatcher.add_handler(MessageHandler(Filters.text &
~Filters.command, handle_message))
updater.start_polling()
updater.idle()
```

Лістинг 3

Проект організовано таким чином, що запуск виконується через команду PowerShell: `python bot.py`

При цьому основна логіка роботи відображається не у консолі, а безпосередньо в Telegram: користувач надсилає повідомлення боту – бот генерує відповідь і повертає її



в чат. Саме такий спосіб реалізації дозволяє забезпечити стабільність роботи, чітку структуру проєкту та зручність подальшого розширення. Уся програма займає порівняно невелику кількість коду, але за рахунок правильної організації структури проєкту, налаштування API-ключів і коректної інтеграції з мовною моделлю, Telegram-бот працює швидко, стабільно та забезпечує якісну генерацію текстів.

Розроблени чат-бот@GeneratedA_bot (#ГенераціяЗображеньітекстів) підтвердив свою ефективність у автоматизації процесів текстогенерації та продемонстрував можливість використання різних мовних моделей у межах одного інструмента. Проведене порівняння стало необхідним для формування висновків щодо доцільності застосування кожної моделі в залежності від потреб користувача (рисунок 1).

Значна увага приділялася порівнянню моделей Gemini 2.5 Pro та OpenAI. Виявлено, що Gemini має перевагу у швидкодії та контекстній точності, тоді як OpenAI демонструє більш стабільну стилістичну якість і природність мови (таблиця 1).

Тестування показало, що обидві моделі здатні генерувати коректні наукові, публіцистичні та художні тексти, проте відрізняються за обсягом, варіативністю та логічністю побудови відповідей. Для об'єктивної оцінки було проведено модульне тестування, інтеграційне тестування та стрес-тестування. Система коректно витримала велику кількість послідовних запитів і продемонструвала достатню стабільність роботи обох моделей у межах Telegram API. Стрес-тестування підтвердило, що Gemini 2.5 Pro забезпечує майже миттєву генерацію коротких відповідей, тоді як OpenAI потребує дещо більше часу, але повертає більш відшліфовані тексти (таблиця 2).

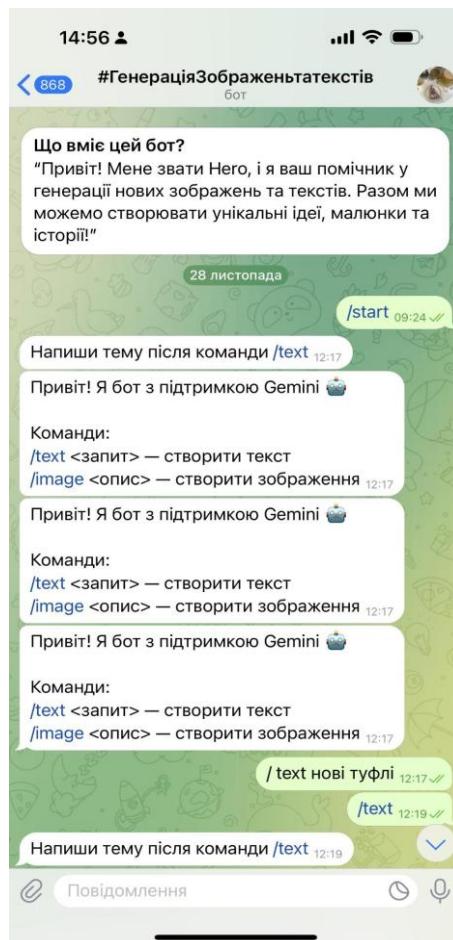


Рисунок 1



Таблиця 1

Порівняння моделей OpenAI та Gemini 2.5 Pro за якісними характеристиками

Параметр	OpenAI (GPT)	Gemini 2.5 Pro
Якість тексту	Висока природність, стилістична узгодженість	Висока логічність, точність формулювань
Розуміння контексту	Добре справляється з довгими діалогами	Чудово працює з великими обсягами інформації
Швидкість відповіді	Середня (1–3 с)	Дуже висока (до 1 с)
Варіативність стилів	Висока	Середня–висока
Креативність	Дуже висока	Висока
Схильність до повторів	Інколи присутня	Помітно нижча
Здатність до узагальнення	Висока	Дуже висока
Стійкість роботи при навантаженні	Висока	Дуже висока
Оптимальне застосування	Творчі та довгі тексти	Точні, фактологічні та структуровані тексти

Таблиця 2

Кількісні результати тестування моделей

Метрика	OpenAI	Gemini 2.5 Pro	Коментар
Середній час генерації	2.4 с	0.9 с	Gemini швидший майже втричі
Максимальне навантаження (запитів/хв)	55	72	Gemini стабільніший під навантаженням
Кількість помилок на 100 запитів	3	2	Обидві працюють стабільно
Середній обсяг відповіді	120–150 слів	110–140 слів	Відмінності незначні
Кількість логічних помилок (оцінено вручну)	5/100	3/100	Gemini точніше інтерпретує факти
Ступінь креативності (1–10)	9	7	OpenAI більш креативний
Стандарт відхилення у стилістиці	8%	5%	Gemini стабільніший у тональності

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У результаті проведеної роботи отримано висновок про те, що OpenAI краще підходить для творчих і художніх текстів, тоді як Gemini 2.5 Pro – для логічних, структурованих та фактологічних відповідей. У результаті проведеного дослідження було продемонстровано сучасні генеративні моделі та технології Telegram API дають можливість створити ефективний інтелектуальний інструмент для генерації текстів. Theoretical analysis and practical implementation confirm the доцільність інтеграції великих мовних моделей у практичні системи автоматизації контенту. Перспективи подальших досліджень включають: оптимізацію витрат на API моделей; розробку адаптивних діалогових сценаріїв; покращення контекстного аналізу; впровадження моделей наступного покоління; використання вебхуків для підвищення продуктивності; розширення функціоналу: збереження історії, персоналізація, підтримка мультимодальності. Отримані результати підтверджують актуальність дослідження та відкривають можливості для подальшого розвитку систем генерації текстів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS 2017)*.



2. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., et al. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv*. <https://arxiv.org/abs/1910.10683>
3. Min, B., Ross, H., Sulem, E., Pourn Ben Veysseh, A., Nguyen, T. H., Sainz, O., Agirre, E., Heinz, I., & Roth, D. (2021). Recent advances in natural language processing via large pre-trained language models: A survey. *arXiv*. <https://arxiv.org/abs/2111.01243>
4. Kalyan, S. S. (2023). A survey of GPT-3 family large language models including ChatGPT and GPT-4. *arXiv*. <https://arxiv.org/abs/2301.06627>
5. Amvera. (2024). Integration of GPT-4 Omni model into a Telegram bot in Python. *Habr*. <https://habr.com>
6. TeLLMgramBot. (n.d.). *Library for building Telegram bots with AI model support*. PyPI. <https://pypi.org>
7. Speka Media. (2023). How artificial intelligence is used in SEO and content marketing. <https://speka.media>
8. National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”. (2024). Context-aware text transformation using ChatGPT [Bachelor’s/Master’s thesis]. Electronic Library KPI. Zaporizhzhia National University. (n.d.). *Machine learning course materials*. <https://moodle.znu.edu.ua>

**Oksana Onyshchuk**

candidate of technical sciences, associate professor
Volyn National University, Lutsk, Ukraine
ORCID: 0000-0002-8342-3011
oksanaoo2024@gmail.com

COMPARISON OF TELEGRAM BOT FOR TEXT GENERATION BUILT ON OPENAI AND GEMINAI

Abstract. The study examines the problem of developing and conducting a comparative analysis of modern generative language models, using OpenAI GPT and Gemini 2.5 Pro integrated into a Telegram bot for automated text generation. The relevance of the research is driven by the increasing demand for fast, flexible, and high-quality content creation tools capable of operating in real time. Within the work, the requirements for the software tool were defined, the architecture of the Telegram bot was developed, the principles of its operation were determined, and the interaction with the APIs of both models was implemented. Special attention is given to testing the quality of generated texts, response speed, model stability, and their behavior under load. Experimental results indicate that Gemini 2.5 Pro demonstrates significantly higher generation speed and more precise interpretation of user queries, whereas OpenAI GPT excels in creative tasks, stylistic naturalness, and flexibility of formulations. Both models show strong performance across multiple text genres, though they differ in structural coherence, logic, and context retention. The results of the study show that neither model has an absolute advantage; the choice depends on specific tasks and requirements for style, accuracy, or speed of response. The developed Telegram bot confirms the effectiveness of integrating generative models into practical systems and demonstrates adaptability to various usage scenarios. The findings can be applied to further optimization of generative systems, model selection for specific applications, and the development of intelligent AI-based services.

Keywords: chatbot; generative models; artificial intelligence; OpenAI GPT; Gemini 2.5 Pro; transformers; Telegram bot; text generation; natural language processing; comparative analysis.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS 2017)*.
2. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., et al. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv*. <https://arxiv.org/abs/1910.10683>
3. Min, B., Ross, H., Sulem, E., Pouran Ben Veyseh, A., Nguyen, T. H., Sainz, O., Agirre, E., Heinz, I., & Roth, D. (2021). Recent advances in natural language processing via large pre-trained language models: A survey. *arXiv*. <https://arxiv.org/abs/2111.01243>
4. Kalyan, S. S. (2023). A survey of GPT-3 family large language models including ChatGPT and GPT-4. *arXiv*. <https://arxiv.org/abs/2301.06627>
5. Amvera. (2024). Integration of GPT-4 Omni model into a Telegram bot in Python. *Habr*. <https://habr.com>
6. TeLLMgramBot. (n.d.). *Library for building Telegram bots with AI model support*. PyPI. <https://pypi.org>
7. Speka Media. (2023). How artificial intelligence is used in SEO and content marketing. <https://speka.media>
8. National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". (2024). Context-aware text transformation using ChatGPT [Bachelor's/Master's thesis]. Electronic Library KPI. Zaporizhzhia National University. (n.d.). *Machine learning course materials*. <https://moodle.znu.edu.ua>

Отримано редакцією журналу / Received: 27.01.26

Прорецензовано / Revised: 18.02.26

Схвалено до друку / Accepted: 26.03.26



This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.