

УДК 004.8:027.7

ПЕРЕВАГИ ТА НЕДОЛІКИ РОБОТИ З API CHATGPT ЧЕРЕЗ МОВУ C++

Сіциліцин Ю.О., Ph.D.

e-mail: yurii.sitsilitsin@tsatu.edu.ua

Таврійський державний агротехнологічний університет імені Дмитра Моторного

Актуальність та постановка проблеми. На сьогоднішній день C++ залишається однією з найпопулярніших мов програмування для високопродуктивних додатків завдяки ефективному управлінню пам'яттю та швидкодії. Велика кількість систем, включаючи ігрові рушії, вбудовані системи, програмне забезпечення реального часу, написані саме на C++. У контексті штучного інтелекту (ШІ) взаємодія з API ChatGPT через C++ є актуальним питанням для спеціалістів, які вже працюють з цією мовою. Зважаючи на наявність великої спільноти C++-розробників, які володіють глибокими знаннями, постає питання: чи є сенс використовувати C++ для взаємодії з API OpenAI? Наскільки це зручно у порівнянні з іншими мовами, такими як Python чи JavaScript, що є стандартами у сфері ШІ? У тезах розглянуто переваги та обмеження інтеграції ChatGPT API з C++.

Основні матеріали дослідження. Основні вимоги для роботи з API ChatGPT через мову C++ полягають у дотриманні кількох ключових аспектів. По-перше, доступ до API надається за допомогою ключа автентифікації, який необхідно отримати на платформі OpenAI. Цей ключ забезпечує авторизацію користувача та перевірку прав доступу до API. Наступним кроком є налаштування взаємодії з вебсервісом за допомогою HTTP-запитів. Для цього в C++ можна використовувати бібліотеки, що підтримують HTTP-протокол, такі як libcurl, Boost.Beast або інші рішення, які дозволяють відправляти та отримувати дані через HTTPS-з'єднання. Важливим аспектом є правильне налаштування заголовків запиту: зокрема, обов'язковим є додавання заголовків Authorization: Bearer <YOUR_API_KEY> для передачі ключа автентифікації та Content-Type: application/json для вказівки формату даних. Формат даних, які передаються на сервер, відповідає JSON-структурі. У запиті обов'язково потрібно вказати модель ШІ (наприклад, gpt-4) і структуру повідомлень, яка включає роль ("user", "system" чи "assistant") і текстове повідомлення. Робота з JSON у C++ вимагає додаткових бібліотек, таких як nlohmann/json або RapidJSON, оскільки стандартна бібліотека мови не має вбудованих засобів для роботи з цим форматом. Після налаштування запиту його відправляють на сервер OpenAI за URL-адресою <https://api.openai.com/v1/chat/completions>. Сервер обробляє запит та повертає JSON-відповідь, яка містить результати роботи моделі: згенерований текст, використані токени та іншу супутню інформацію. Важливо передбачити обробку відповіді та можливих помилок, зокрема помилок автентифікації, некоректного формату запиту чи перевищення лімітів використання API. Таким чином, робота з API ChatGPT через C++ вимагає точного налаштування HTTP-запитів, використання JSON-парсерів та обробки відповідей сервера для ефективної взаємодії з моделлю.

Використання C++ для інтеграції з API ChatGPT має свої особливості, які полягають у необхідності ручної конфігурації ключових компонентів. C++ не має готових SDK для взаємодії з API OpenAI, як це є у Python, що вимагає використання сторонніх бібліотек для реалізації HTTP-запитів і обробки JSON-

даних. Для передачі даних до API можна використовувати бібліотеку libcurl, яка забезпечує роботу з HTTP та HTTPS-протоколами, або сучасні рішення, такі як Boost.Beast, що підтримують асинхронну взаємодію.

Обробка даних у форматі JSON є важливим етапом інтеграції, оскільки API ChatGPT вимагає передачі структурованої інформації. C++ не має вбудованої підтримки JSON, тому розробнику потрібно підключати додаткові бібліотеки, як-от nlohmann/json або RapidJSON, для створення, модифікації та парсингу JSON-об'єктів.

У процесі розробки також потрібно враховувати обробку помилок, оскільки будь-які несправності у форматі запиту або відсутність ключа автентифікації призведуть до некоректної відповіді від API. На відміну від Python, де є мінімальний вхідний бар'єр для роботи з OpenAI, у C++ розробник стикається з додатковою складністю налаштування середовища та більшою кількістю коду для реалізації базової взаємодії.

Попри це, C++ надає високу продуктивність і гнучкість, що робить його ефективним інструментом для інтеграції API у великі проєкти. Це особливо актуально для систем, що потребують максимальної швидкодії або працюють в умовах обмежених ресурсів, як-от вбудовані системи чи ігрові рушії. Таким чином, використання C++ для інтеграції з API ChatGPT є виправданим у випадках, коли необхідна оптимізація продуктивності та контроль за використанням ресурсів.

Переваги використання C++ з API ChatGPT полягають у його продуктивності, надійності та гнучкості. Завдяки ефективному управлінню пам'яттю та низькорівневій оптимізації, C++ забезпечує високу швидкість при обробці великих обсягів даних та виконанні складних обчислювальних задач. Це робить його особливо корисним для інтеграції API ChatGPT у системи, що потребують мінімальних затримок або працюють у реальному часі, наприклад, вбудовані пристрої, серверні рішення або програми для обробки природної мови. Крім продуктивності, C++ є надійною мовою програмування, що дозволяє створювати стабільні та масштабовані рішення. Це важливо для великих проєктів, де інтеграція з API ChatGPT може розширити функціонал вже існуючих систем без необхідності переходу на інші мови програмування. Також гнучкість C++ дає змогу розробникам повністю контролювати процес передачі HTTP-запитів, обробки JSON-даних і керування ресурсами, що забезпечує оптимальну взаємодію з API OpenAI. Використання C++ є особливо актуальним у сферах, де необхідна максимальна ефективність, наприклад, у високонавантажених додатках, ігрових рушіях або системах автоматизованого виробництва. Це дозволяє поєднати можливості передових моделей штучного інтелекту з високошвидкісними алгоритмами і надійною архітектурою C++.

Недоліки використання C++ з API ChatGPT пов'язані зі складністю реалізації, відсутністю готових інструментів і необхідністю значного обсягу коду для досягнення базової функціональності. На відміну від мов програмування, таких як Python чи JavaScript, де доступні готові SDK і бібліотеки для роботи з API OpenAI, у C++ розробнику доводиться налаштовувати взаємодію вручну. Це включає роботу з HTTP-запитами, обробку JSON-структур і управління помилками, що ускладнює розробку та потребує додаткових знань. Ще одним недоліком є відсутність вбудованих засобів для роботи з JSON, через що потрібно підключати сторонні бібліотеки, такі як RapidJSON або nlohmann/json. Це додає кроки до процесу розробки та вимагає ретельного налаштування залежностей проєкту. Складність інтеграції також проявляється у великому обсязі коду, який

необхідно написати для реалізації простих задач, на кшталт відправки запиту та обробки відповіді від API. Такий підхід значно програє у зручності розробки мовам вищого рівня, де аналогічні операції виконуються простіше і швидше. Крім того, у C++ відсутня гнучкість роботи з динамічними об'єктами та строками, що характерна для мов із вбудованими засобами для обробки текстових даних. Це може створювати додаткові труднощі під час інтеграції API, що активно працює з текстовою інформацією. Загалом, використання C++ вимагає більше ресурсів і часу на розробку, що робить його менш ефективним вибором для швидкої інтеграції API ChatGPT, особливо у порівнянні з мовами, орієнтованими на штучний інтелект.

Порівняємо C++ та Python для роботи з API ChatGPT за допомогою таблиці (Таблиця 1).

Таблиця 1 Порівняння C++ та Python для роботи з API

Критерій	C++	Python
Швидкодія	Висока	Помірна
Наявність SDK	Відсутнє	Доступне (openai)
Налаштування запитів	Складніше, але гнучкіше	Просте та інтуїтивне
Обробка JSON	Додаткові бібліотеки	Вбудована підтримка
Розмір коду	Більший	Коротший

Практичні кейси використання C++ для роботи з API ChatGPT демонструють його ефективність у специфічних сферах, де потрібна висока продуктивність і контроль за ресурсами. Одним із таких прикладів є інтеграція в ігрові рушії, де C++ є основною мовою розробки завдяки своїй швидкодії та надійності. Використання API ChatGPT у таких системах дозволяє створювати інтерактивних персонажів з природною мовною взаємодією, що значно підвищує ігровий досвід.

Іншим важливим напрямом є системи реального часу та вбудовані пристрої, де обмежені обчислювальні ресурси не дозволяють використовувати мови з вищими накладними витратами. У таких випадках C++ забезпечує оптимальну роботу з API, дозволяючи інтегрувати можливості штучного інтелекту в IoT-пристрої, автономні системи чи програми автоматизації.

Окрім цього, C++ часто застосовується у великих корпоративних програмних рішеннях, де існує необхідність у стабільній інтеграції додаткових функцій без зміни основної архітектури. Використання API ChatGPT у таких проєктах дозволяє розширити функціонал систем для обробки природної мови, генерації звітів або автоматизації відповіді на запити користувачів. У результаті, C++ стає важливим інструментом у високонавантажених середовищах, які вимагають ефективної роботи зі штучним інтелектом без шкоди для продуктивності.

Висновки. Використання C++ для взаємодії з API ChatGPT є доцільним у випадках, коли критичними є продуктивність, ефективне управління ресурсами та інтеграція у вже існуючі системи, розроблені на цій мові. Завдяки своїм характеристикам C++ забезпечує високу швидкодію та надійність роботи, що особливо актуально для вбудованих систем, ігрових рушіїв, серверних додатків і програм реального часу. У таких середовищах можливості API ChatGPT, як-от генерація тексту та діалогова взаємодія, дозволяють створювати інноваційні

рішення з використанням штучного інтелекту без значних компромісів щодо продуктивності.

Водночас використання C++ має і свої недоліки. Відсутність готових інструментів та SDK для роботи з OpenAI API призводить до необхідності ручної реалізації HTTP-запитів і обробки JSON-структур, що збільшує обсяг коду та ускладнює розробку. У порівнянні з мовами програмування, які активно використовуються у сфері штучного інтелекту, такими як Python, робота з API у C++ потребує більше часу, ресурсів і навичок.

Однак, незважаючи на ці обмеження, C++ залишається незамінною мовою у сферах, де важлива швидкодія та низькорівнева оптимізація. Таким чином, інтеграція API ChatGPT у C++ є перспективним напрямом для розробників, які працюють над високопродуктивними та критичними до ресурсів проєктами. Це дозволяє поєднати можливості сучасних мовних моделей із перевагами C++ для створення масштабованих, надійних і ефективних рішень у різних галузях.

Список використаних джерел:

1. Brown T., Mann B., Ryder N., Subbiah M., Kaplan J. D., Dhariwal P., Neelakantan A., et al. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*. 2020. Vol. 33. P. 1877–1901.
2. Purdy A., Large Language Models and Their Applications: ChatGPT and Beyond. *Journal of Artificial Intelligence Research*. 2023. Vol. 67. P. 45–68.

УДК 004.89

ОСОБЛИВОСТІ ТА НАПРЯМКИ ВИКОРИСТАННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

Скрипченко С.С., здобувач вищої освіти *e-mail:senia200176@gmail.com*
Науковий керівник: к.т.н., доцент Мірошніченко М.Ю.
Таврійський державний агротехнологічний університет імені Дмитра Моторного

Актуальність та постановка проблеми. Глибинні процеси інформатизації суспільства призвели до широкого використання різноманітного програмного забезпечення для реалізації безлічі прикладних задач. Одним з перспективних типів програмного забезпечення є інтелектуальні системи (ІС), в основу яких покладено функціонування елементів штучного інтелекту (ШІ). ІС набули значної популярності, оскільки забезпечують вирішення складних неформалізованих задач, які раніше традиційним програмним забезпеченням вирішити було неможливо.

З урахуванням актуальності штучного інтелекту та інтелектуальних систем зокрема, мета дослідження полягає у висвітленні значення та ключових особливостей інтелектуальних систем, а також напрямків та перспектив практичного застосування. Окремо розглянемо прикладне значення таких систем у різноманітних галузях діяльності, що підкреслює їхню актуальність у сучасному світі та значний потенціал для майбутнього розвитку.

Основні матеріали дослідження. Зростання продуктивності обчислювальних пристроїв та розробка сучасних алгоритмів обробки інформації