

DOI: <https://doi.org/10.36910/6775-2524-0560-2025-58-15>

УДК 004.89

Проніна Ольга Ігорівна, к.т.н., доцент

<https://orcid.org/0000-0001-7085-8027>

Синицін Руслан Валерійович, студент

<https://orcid.org/0009-0007-1825-1587>

Державний вищий навчальний заклад «Приазовський державний технічний університет», м. Дніпро/Маріуполь, Україна

РОЗРОБКА AI АСИСТЕНТА ДЛЯ ЛОКАЛЬНОГО КОРИСТУВАЧА НА МОВІ ПРОГРАМУВАННЯ PYTHON

Проніна О.І., Синицін Р.В. Розробка AI асистента для локального користувача на мові програмування Python. У статті розглядається система розробки багатofункціонального AI-асистента, який здатний виконувати широкий спектр завдань. Представлено модель AI-асистента, розроблену на основі нейронних мереж, яка забезпечує високу точність у виконанні команд та здатна працювати в реальному часі. Запропонована модель досягла високої точності в розпізнаванні голосу, ефективно працюючи навіть коли використовується без підключення до інтернету. В моделі є інтеграція системи розпізнавання мови Vosk та Speech Recognition вони дозволяють точно й ефективно обробляти голосові команди. Виконується розпізнавання голосових команд у режимі реального часу, незалежно від того, чи йдеться про прості команди для окремих дій, чи про сценарії, які включають отримання інформації. Також реалізовано обробка текстового введення для виконання команд. Особлива увага в роботі приділялась розробці ефективної математичної моделі, яка мінімізує затримки у виконанні запитів і забезпечує стабільну роботу.

Ключові слова: AI-асистент, нейронні мережі, Vosk, Speech Recognition, голосові команди, локальна обробка.

Pronina O., Synytsin R. Development of AI assistant for local user in Python programming language. The article considers a system for developing a multifunctional AI assistant capable of performing a wide range of tasks. An AI assistant model developed based on neural networks is presented, which provides high accuracy in executing commands and is capable of working in real time. The proposed model has achieved high accuracy in voice recognition, working effectively even when used without an Internet connection. The model integrates the Vosk and Speech Recognition speech recognition systems, which allow for accurate and efficient processing of voice commands. Voice commands are recognized in real time, regardless of whether they are simple commands for individual actions or scenarios that include obtaining information. Text input processing for executing commands is also implemented. Particular attention in the work was paid to the development of an effective mathematical model that minimizes delays in executing requests and ensures stable operation.

Keywords: AI assistant, neural networks, Vosk, Speech Recognition, voice commands, local processing.

Постановка наукової проблеми. В сучасному світі інформаційних технологій штучний інтелект є трендовим напрямком розробки. Сучасні моделі штучного інтелекту дозволяють виконувати безліч цікавих завдань. Одним з напрямків використання штучного інтелекту є спілкування з чатами та помічниками на природній мові – текстом чи голосом. Однією з важливих проблем при розробці AI-асистента є забезпечення високої точності та ефективності обробки саме голосових команд. Vosk [1] є найбільш ефективний інструмент для локального розпізнавання мови, який забезпечує точність навіть в умовах не підключеного інтернету. Speech Recognition [2] дозволяє швидко обробляти великий обсяг запитів, але вимагає підключення до інтернету. Після розпізнавання голосу надсилається запит до ChatGPT [3]. Завдяки можливості обробляти запити та генерувати відповіді на різні запитання, дозволяє надавати корисну інформацію в реальному часі. Згенеровано відповідь надсилається до інтерфейсу PyQt6 [4] де текст виводиться на екран за допомогою QTextEdit [5]. Також є важливою складовою для побудови AI-асистента модель, яка підтримує синтез голосу Silero Models що дозволяє створювати двосторонні взаємодії між користувачем.

Завдяки PyQt6 можна комбінувати технології в інтерфейсі, що допомагає ефективно обробляти голосові команди, інтегрувати різні функціональні можливості та надавати користувачам точну та швидку відповідь. PyQt6 допомагає зробити інтерфейс більш чуйним та зручним у використанні.

Аналіз останніх досліджень та публікацій. Наукова галузь розробки AI-асистентів є достатньо популярною, існує ряд робіт, деякі з яких розглянути в роботі. В роботі [6] розглядається питання чат-боті, що є ідеальною демонстрацією того, як прості завдання та запити, такі як підтримка клієнтів або показники продажів і звітність, можна вирішити без втручання людини. В роботі представлено орієнтовану на завдання структуру чат-бота для іспанської мови на веб-сторінці точки продажу. Автори застосували методи обробки природної мови (NLP) і оцінили два методи навчання під наглядом. Для реалізації цієї діалогової системи в подальших роботах була

запропонована архітектура програмного забезпечення, що використовує обчислювальний сервіс PaaS і структуру API.

У роботі [7] розглядається пошук найліпшого варіанту для впровадження штучної нейронної мережі в інформаційній системі для технічних інструкцій студентської дипломної роботи. Автори обрали для машинного навчання послідовну модель, це означає, що ANN використовує лише 1 вхідний шар, прихований/щільний шар і 1 вихідний шар. У процесі підготовки даних застосовано метод стохастичного градієнта (SGD). Результатами цього дослідження є додатки чат-ботів і тестування моделі з використанням матриці плутанини. Робота [8] присвячена використанню генеративних помічників ШІ (GAI) для розробки програмного забезпечення. Хоча ШІ вже використовується в різних сферах розробки програмного забезпечення, технології GAI, такі як GitHub Copilot і ChatGPT.

У статті [9] автори провели масштабне дослідження, спрямоване на те, як використовуються помічники ШІ, зосереджуючись на конкретних діях і етапах розробки програмного забезпечення. Було виявлено, що використання помічників ШІ залежить від активності та стадії та активно впроваджується.

Робота [10] присвячена моделюванню нового покоління віртуальних персональних помічників як інтегрованого голосового помічника в ОС Windows. Автори в своїй роботі підтверджують, що навчання з підкріпленням може вивчати попередньо визначену модель віртуального помічника за допомогою визначеного користувачем набору даних.

Мета дослідження. Метою роботи є створення AI-асистента, який буде здатний ефективно працювати в реальному часі, виконуючи запити користувача через голосове або текстове введення.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. У цієї роботі було побудовано власного AI-асистента. Користувач надсилає запит, який спочатку обробляється системою для визначення його цілі, далі запит або надсилається до ChatGPT, або виконується на локальному комп'ютері. Результати повертаються користувачу через канал зв'язку, забезпечуючи швидкий і зрозумілий зворотний зв'язок.

Таким чином, AI асистента дозволяє реалізувати його основні функції голосових запитів, інтеграцію з API, автоматизацію завдань, взаємодію з додатками, а також забезпечує ефективну і зручну взаємодію з користувачем. На рис. 1 показується використання вбудованих команд.

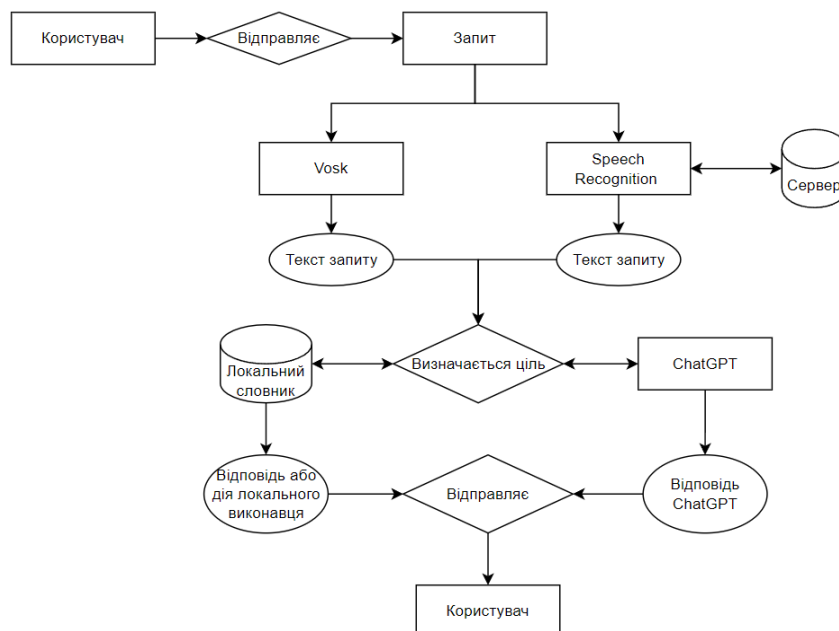


Рис. 1 – Схема роботи AI-асистента

AI-асистент ініціалізує всі необхідні ресурси для роботи з даними. Коли користувач дає запит, він перевіряє, чи є підключення до інтернету, оскільки це може вплинути на API чи онлайн сервіси. Далі йде етап обробки введених даних якщо це голосова команда, вона проходить через систему розпізнавання мови Vosk-офлайн або Speech Recognition-онлайн. AI-асистент конвертує голосові

дані в текст. Після того, як команда була перетворена у текст, асистент розпізнає ключові слова та фрази, щоб зрозуміти, яку задачу потрібно виконати.

Після аналізу тексту асистент перевіряє, чи є в його базі даних необхідні скрипти для виконання цієї задачі, якщо відповідь знайдена, AI асистент виконує команду або надає відповідь на запит користувача.

Якщо команда вимагає доступу до зовнішніх сервісів, AI асистент звертається до потрібних API або посилань. Після того, як задача виконана, асистент надає користувачу результат. На рисунку 2 наведено інтерфейс додатку, та запит на побудову графіка по коду. На рисунку 3 виконано вже обробку запита користувача та наведено результат виконання. Були проведені експериментальні дослідження для виявлення часу відповіді та точності виконання. В таблиці 1 наведено фрагмент експерименту у вигляді різних запитів до AI-асистента.

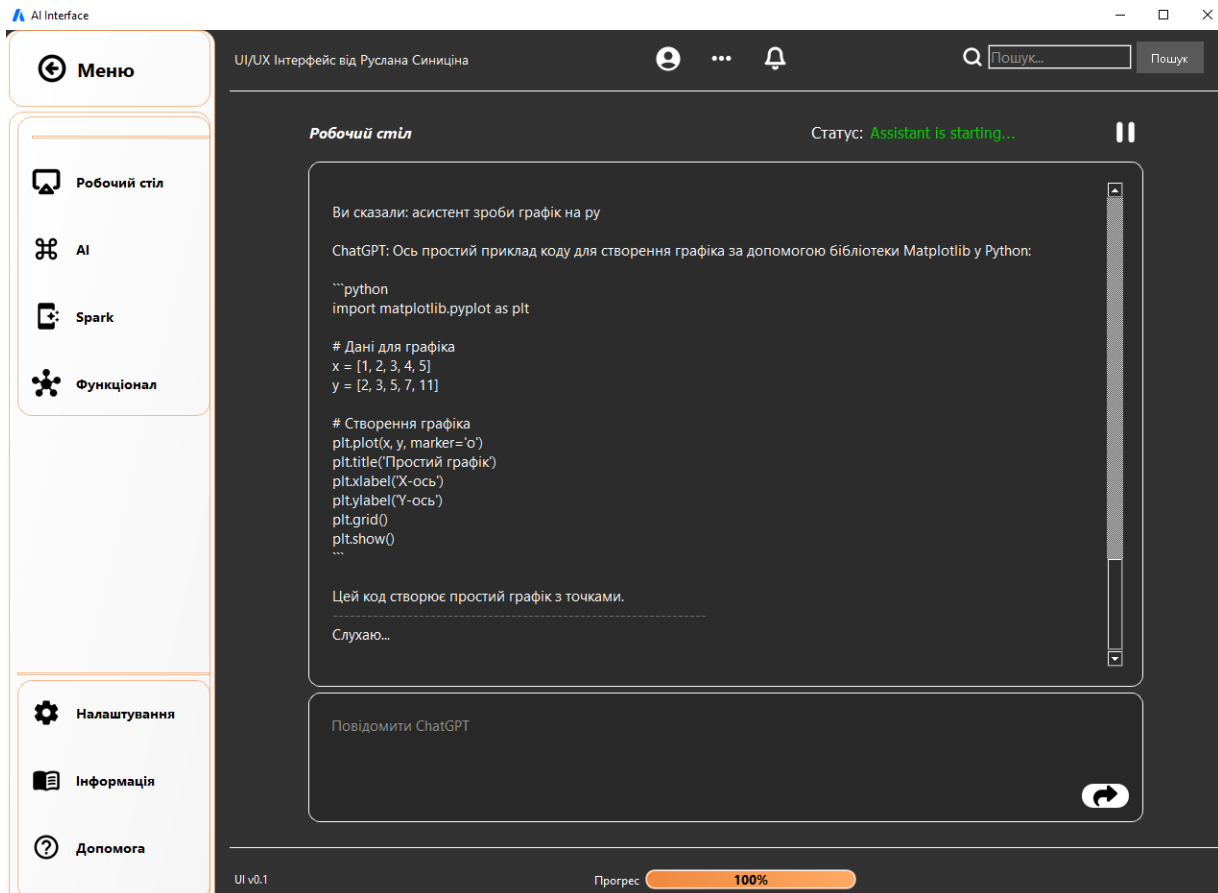


Рис. 2 – Перевірка працездатності та правильності відповіді, запит до AI – побудова графіка

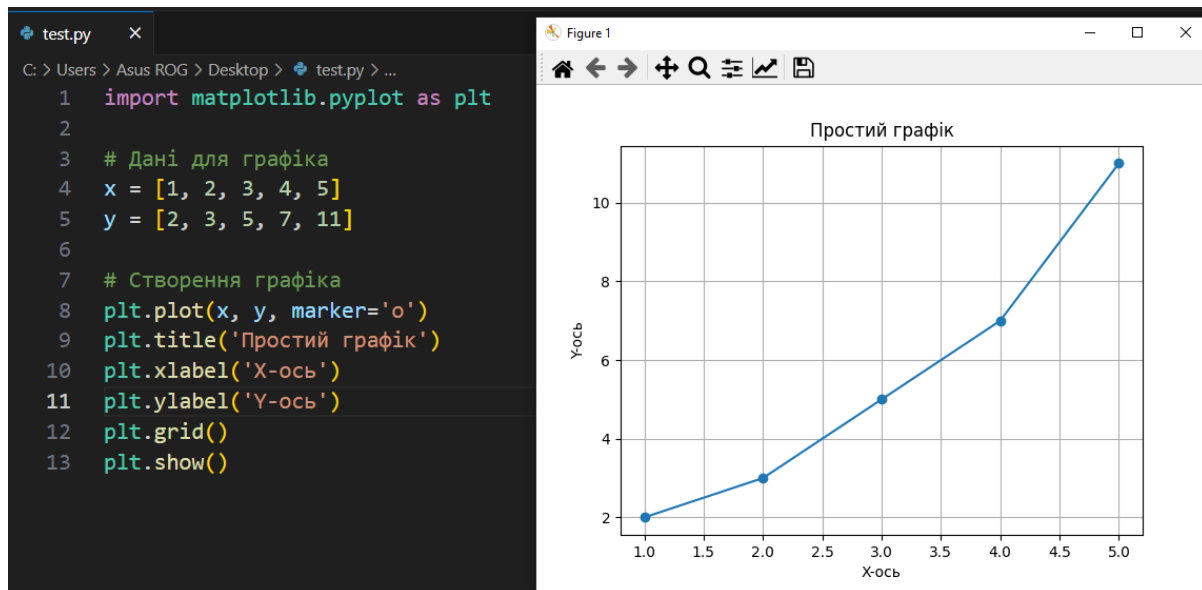


Рис. 3 – Результати запиту користувача на побудову графіка

Таблиця 1 – Результат проведення експерименту

Запит	Точність виконання	Швидкість виконання
Скільки буде п'ять помножити на сім і поділити на три?	1.0	0.421 секунд
Що таке штучний інтелект і як він впливає на сучасний світ?	0.89	2.18 секунд
How do you say hello world in Ukrainian?	1.0	0.514 секунд
Що таке мова Python і яка відмінність від C++?	0.71	2.35 секунд
Створи мені таблицю мов програмування та опиши для чого вони	0.83	1.67 секунд

Аналізуючи отримані результати тестування AI асистента можна зробити висновок, що результати вийшли цілком позитивними. AI асистент розуміє як українську так і англійську мову, завдяки тренуванню моделі. Основною перевагою є те, що асистент може обробляти команди користувачів швидко та точно, забезпечуючи користувачам високу ефективність у виконанні різноманітних завдань. Також, значним плюсом є інтеграція з іншими додатками та програмами, що дозволяє безперешкодно виконувати завдання.

Висновки та перспективи подальшого дослідження. Було реалізовано AI асистента, який здатний значно покращити виконання різноманітних завдань локального користувачів. Застосування цього програмного забезпечення дозволяє знизити рівень надмірної інформації, підвищити швидкість обробки запитів і забезпечити точність виконання команд. Однією з головних переваг цієї системи є зменшення ймовірності помилок при виконанні рутинних завдань, що дозволяє заощадити час і зусилля користувачів. Розроблена система допомагає з оптимізацією алгоритмів різнопланових завдань, зокрема в автоматизації повсякденних процесів, таких як відкриття файлів, запуск скриптів, управління комп'ютером та взаємодія з іншими додатками. Програмне забезпечення також має великий потенціал для розвитку в якості консультативного помічника, допомагаючи студентам та професіоналам працювати з новими технологіями. Що в свою чергу дозволяє AI асистенту адаптуватися до змінюваних потреб користувачів, підвищуючи точність і ефективність виконання завдань у будь-якому середовищі.

Список бібліографічного опису

1. Install Vosk – Режим доступу: <https://alphacephei.com/vosk/>
2. Speech Recognition – Режим доступу: <https://pypi.org/project/SpeechRecognition/>
3. API ChatGPT – Режим доступу: <https://platform.openai.com/docs/overview>
4. Install PyQt6 – Режим доступу: <https://doc.qt.io/qtforpython-6/>
5. QTextEdit from PyQt6 – Режим доступу: <https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/QTextEdit.html>
6. Conversational AI Assistant Using Artificial Neural Networks. – Режим доступу: <https://run.unl.pt/bitstream/10362/127803/1/TAA0106.pdf>

7. Sanjaya, Ovi & Bukhori, Saiful & Furqon, Muhammad. (2023). Virtual Assistant for Thesis Technical Guide Using Artificial Neural Network. Indonesian Journal of Artificial Intelligence and Data Mining. 6. 188-196. 10.24014/ijaidm.v6i2.23473.
8. Bull, Christopher & Kharrufa, Ahmed. (2023). Generative AI Assistants in Software Development Education. 10.48550/arXiv.2303.13936.
9. Sergeyuk, Agnia & Golubev, Yaroslav & Bryksin, Timofey & Ahmed, Iftekhar. (2024). Using AI-Based Coding Assistants in Practice: State of Affairs, Perceptions, and Ways Forward. 10.2139/ssrn.4900362.
10. Baskaran, G & Raj, Harrish & Kumar, S & Anand, R. (2021). To Build A Virtual Assistant By Using Artificial Intelligence. The Open Artificial Intelligence Journal. 2. 1134. 10.6084/m9.figshare.14446467.

References

1. Install Vosk – Link: <https://alphacephei.com/vosk/>
2. Speech Recognition – Link: <https://pypi.org/project/SpeechRecognition/>
3. API ChatGPT – Link: <https://platform.openai.com/docs/overview>
4. Install PyQt6 – Link: <https://doc.qt.io/qtforpython-6/>
5. QTextEdit from PyQt6 – Link: <https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/QTextEdit.html>
6. Conversational AI Assistant Using Artificial Neural Networks. – Link: <https://run.unl.pt/bitstream/10362/127803/1/TAA0106.pdf>
7. Sanjaya, Ovi & Bukhori, Saiful & Furqon, Muhammad. (2023). Virtual Assistant for Thesis Technical Guide Using Artificial Neural Network. Indonesian Journal of Artificial Intelligence and Data Mining. 6. 188-196. 10.24014/ijaidm.v6i2.23473.
8. Bull, Christopher & Kharrufa, Ahmed. (2023). Generative AI Assistants in Software Development Education. 10.48550/arXiv.2303.13936.
9. Sergeyuk, Agnia & Golubev, Yaroslav & Bryksin, Timofey & Ahmed, Iftekhar. (2024). Using AI-Based Coding Assistants in Practice: State of Affairs, Perceptions, and Ways Forward. 10.2139/ssrn.4900362.
10. Baskaran, G & Raj, Harrish & Kumar, S & Anand, R. (2021). To Build A Virtual Assistant By Using Artificial Intelligence. The Open Artificial Intelligence Journal. 2. 1134. 10.6084/m9.figshare.14446467.