НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

«На правах рукопису»
УДК 004.8+004.5

До захисту допущено:
В.О. завідувача кафедри

_____
Михайло НОВОТАРСЬКИЙ
«___»_____2025 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-науковою програмою «Комп'ютерні системи та мережі» зі спеціальності 123 «Комп'ютерна інженерія»

на тему: «Метод побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформи дистанційного навчання»

Виконав:
студент ІІ курсу, групи ІО-31мн
Писарчук Ілля Олексійович _____

Керівник:
проф., д.т.н., проф.
Клименко Ірина Анатоліївна _____

Консультант з нормоконтролю:
проф. каф. ОТ, д.т.н., проф.
Юрій Кулаков _____

Рецензент:
доцент кафедри ІСТ к.т.н, доцент
Володимир Шимкович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2025 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет *Інформатики та обчислювальної техніки*
(повна назва)

Кафедра *Обчислювальної техніки*
(повна назва)

Рівень вищої освіти – другий (магістерський)

Спеціальність *123 Комп'ютерна інженерія*
(код і назва)

Освітньо-наукова програма *Комп'ютерні системи та мережі*
(код і назва)

ЗАТВЕРДЖУЮ
В.О. завідувача кафедри

_____
(підпис)
Михайло НОВОТАРСЬКИЙ
«____» _____ 2025 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Писарчуку Іллі Олексійовичу
(прізвище, ім'я, по батькові)

1. Тема практики Метод побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформи дистанційного навчання

Науковий керівник Клименко Ірина Анатоліївна, професор, доктор технічних наук
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «13» березня 2025 р. № 1128

2. Строк подання студентом дисертації 05 травня 2025 року

3. Об'єкт дослідження процеси обробки природньої мови з технологіями Natural Language Processing – NLP

4. Предмет дослідження методи машинного навчання для обробки природньої мови з технологіями Natural Language Processing – NLP

5. Перелік завдань, які потрібно розробити:

1). Аналіз принципів побудови та організації інформаційної підтримки тих, хто навчається на відомих платформах дистанційного навчання;

2). Аналіз відомих методологічних та технологічних підходів до обробки природньої мови за технологіями NLP та великих мовних моделей;

3). Розробка методики побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей;

4). Розробка архітектури програмно-апаратної платформ дистанційного навчання з інтелектуальною інформаційною підтримкою;

5). Розробка програмної складової підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за принципами NLP та великих мовних моделей.

6. Консультанти розділів звіту з практики:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	проф. каф. ОТ Юрій Кулаков		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання практики	Строк виконання етапів практики	Примітки
1.	Аналіз принципів побудови та організації інформаційної підтримки тих, хто навчається на відомих платформах дистанційного навчання	14.03.25-21.03.25	
2.	Аналіз відомих методологічних та технологічних підходів до обробки природньої мови за технологіями NLP та великих мовних моделей	21.03.25-28.03.25	
3.	Розробка методики побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей	28.03.25-04.04.25	
4.	Розробка архітектури програмно-апаратної платформ дистанційного навчання з інтелектуальною інформаційною підтримкою	04.04.25-11.04.25	
5.	Розробка програмної складової підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за принципами NLP та великих мовних моделей	11.04.25-18.04.25	

7.	Розробка стартап-проєкту	18.04.25-22.04.25	
8.	Підготовка звітної документації та завершення робіт	22.04.25-03.05.25	

Студент ___Ілля ПИСАРЧУК__ _____
(підпис)

Науковий керівник дисертації ___ Ірина КЛИМЕНКО _____ _____
(підпис)

РЕФЕРАТ

на магістерську дисертацію

виконану на тему:

Метод побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформи дистанційного навчання

студентом: Писарчуком Іллею Олексійовичем

Актуальність теми. Реалії сьогодення – питання безпеки, пандемічні заходи – безумовно актуалізують формат дистанційного навчання. Якщо приймати це як реальний факт, що діє вже протягом декількох років то питання удосконалення платформ дистанційного навчання є і буде актуальною задачею. В класичній реалізації платформа дистанційного навчання це web-портал із сервісами комунікації, зберігання і надання доступу до різнорідної освітньої інформації. Дисертаційна робота присвячена розробці методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформи дистанційного навчання, що робить її актуальною та конче необхідною.

Метою роботи автоматизація процесу інтелектуальної інформаційної підтримки тих хто навчається на платформі дистанційного навчання.

Об'єктом дослідження процеси обробки природньої мови за технологіями Natural Language Processing (NLP).

Предмет дослідження. методи машинного навчання для обробки природньої мови за технологіями Natural Language Processing (NLP).

Методи дослідження. У роботі використовуються методи машинного навчання в аспекті обробки природньої мови (NLP): фільтрації, токенізації; лематизації інформаційних повідомлень; лексичний та контент аналіз з використанням великих мовних моделей, що підтримуються генеративним штучним інтелектом.

Наукова новизна. Новизна роботи полягає у запропонованому методі побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформи дистанційного навчання. Метод базується на послідовному

визначенні контенту інформаційного повідомлення з використанням великих мовних моделей та відрізняється від аналогів формування інтерфейсного запиту до методів доступу к різнорідної інформації, яка становить суть методичного забезпечення освітнього процесу платформи дистанційного навчання. Запит формується із впровадженням методів штучного інтелекту за технологіями обробки природньої мови. Результатом є вичерпана релевантна відповідь про інформаційні кластери методичного забезпечення, що підтримують суть відповіді на запит, сформований природною мовою.

Практична цінність досліджень. Полягає в отримання конкретних прикладних результатів: архітектури програмно-апаратної платформ дистанційного навчання з інтелектуальною інформаційною підтримкою; програмної складової підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання. Практика впровадження запропонованих рішень забезпечує автоматизацію процесу інтелектуальної інформаційної підтримки тих хто навчається на платформі дистанційного навчання, що підтверджено актом впровадження.

Особистий внесок магістранта. Усі основні результати отримані особисто здобувачем. Зокрема: розроблено методі побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформи дистанційного навчання; запропоновано архітектуру програмно-апаратної платформ дистанційного навчання з інтелектуальною інформаційною підтримкою; розроблено програмну компоненту підсистеми інтелектуальної підтримки.

Структура та обсяг роботи. Робота складається зі вступу та чотирьох розділів. Загальний обсяг роботи: 00 аркуші основного тексту, 00 ілюстрації. При підготовці використовувалася технічна та наукова література з 00 різних джерел.

Ключові слова. Інформаційна система, комп’ютерна система, дистанційне навчання, аналіз та обробка природної мови (Natural Language Processing (NLP)), машинне навчання, глибинне навчання, нейронні мережі, штучний інтелект.

ABSTRACT

for a master's thesis

made on the topic:

Method of building an intelligent support subsystem for a software and hardware platform for distance learning

Student: Pisarchuk Ilya Oleksiyovych

Relevance of the topic. The realities of today – security issues, pandemic measures – certainly actualize the format of distance learning. If we take this as a real fact that we have been working for several years, then the issue of improving distance learning platforms is and will be an urgent task. In the classic implementation, the distance learning platform is a web portal with services for communication, storage and access to heterogeneous educational information. The dissertation is devoted to the development of a method for building an intelligent support subsystem for a software and hardware platform for distance learning, which makes it relevant and extremely necessary.

The purpose of the work is to automate the process of intelligent information support for those who study on the distance learning platform.

The object of the study is the processes of natural language processing using Natural Language Processing (NLP) technologies.

Subject of research. machine learning methods for natural language processing using Natural Language Processing (NLP) technologies.

Research methods. The paper uses machine learning methods in the aspect of natural language processing (NLP): filtering, tokenization; lemmatization of information messages; Lexical and content analysis using large language models supported by generative artificial intelligence.

Scientific novelty. The novelty of the work lies in the proposed method of building an intelligent support subsystem for a software and hardware platform for distance learning. The method is based on the sequential definition of the content of an information message using large language models and differs from analogues of the

formation of an interface request to methods of access to heterogeneous information, which is the essence of the methodological support of the educational process of the distance learning platform. The request is formed with the introduction of artificial intelligence methods using natural language processing technologies. The result is an exhausted relevant answer about information clusters of methodological support that support the essence of the response to the request formed in natural language.

Practical value of research. It consists in obtaining specific applied results: architecture of software and hardware platforms for distance learning with intelligent information support; software component of the intelligent support subsystem for software and hardware platforms of distance learning. The practice of implementing the proposed solutions provides automation of the process of intelligent information support for those who study on the distance learning platform, which is confirmed by the act of implementation.

Personal contribution of the master's student. All the main results were obtained personally by the applicant. In particular: a method for building an intelligent support subsystem for a software and hardware platform for distance learning has been developed; the architecture of software and hardware platforms for distance learning with intelligent information support has been proposed; the software component of the intelligent support subsystem has been developed.

Structure and scope of work. The work consists of an introduction and three sections. Total amount of work: 00 sheets of the main text, 00 illustrations. In the preparation, technical and scientific literature from 00 different sources was used.

Keywords. Information System, Computer System, Distance Learning, Natural Language Processing (NLP), Machine Learning, Deep Learning, Neural Networks, Artificial Intelligence.

Пояснювальна записка
до магістерської дисертації

на тему: «Метод побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформи дистанційного навчання»

Київ – 2025 року

ЗМІСТ

ВСТУП 12

РОЗДІЛ 1 АНАЛІЗ ВІДОМИХ МЕТОДОЛОГІЧНИХ ПІДХОДІВ ДО ПОБУДОВИ ПІДСИСТЕМИ ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ ДЛЯ ПРОГРАМНО-АПАРАТНОЇ ПЛАТФОРМ ДИСТАНЦІЙНОГО НАВЧАННЯ ... 14

1.1. Аналіз принципів побудови та організації інформаційної підтримки тих, хто навчається на відомих платформах дистанційного навчання 15

1.2. Аналіз відомих методологічних та технологічних підходів до обробки природньої мови за технологіями NLP та великих мовних моделей 18

1.3. Формалізація задачі досліджень та постановка часткових задач досліджень 29

1.3.1. Загальна постановка задачі дослідження 29

1.3.2. Декомпозиція на часткові задачі 31

1.3.3. Обмеження та припущення 31

ВИСНОВОК ДО РОЗДІЛУ 1 33

РОЗДІЛ 2 МЕТОД ПОБУДОВИ ПІДСИСТЕМИ ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ ДЛЯ ПРОГРАМНО-АПАРАТНОЇ ПЛАТФОРМ ДИСТАНЦІЙНОГО НАВЧАННЯ ЗА ТЕХНОЛОГІЯМИ NLP ТА ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ 34

2.1. Розробка методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання 34

2.2. Архітектура програмно-апаратної платформ дистанційного навчання з інтелектуальною інформаційною підтримкою 38

2.3. Програмна компонента підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за принципами NLP та великих мовних моделей 41

2.3.1. Інженерія вимог 41

2.3.2. Архітектурне проектування програмної компоненти 43

2.3.3. Програмна реалізація 50

2.3.4. Дослідження ефективності та доналаштування моделі Gemini Flash 2.0 для програмної компоненти інтелектуального помічника 58

ВИСНОВОК ДО РОЗДІЛУ 2 63

РОЗДІЛ 3 ПРИКЛАДНІ АСПЕКТИ ЗАСТОСУВАННЯ ПІДСИСТЕМИ ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ ДЛЯ ПРОГРАМНО-АПАРАТНОЇ ПЛАТФОРМ ДИСТАНЦІЙНОГО НАВЧАННЯ ЗА ТЕХНОЛОГІЯМИ NLP ТА ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ 65

3.1. Тестування та оцінювання ефективності запропонованих рішень 65

ВИСНОВОК ДО РОЗДІЛУ 3 79

РОЗДІЛ 4 СТАРТАП: «ПІДСИСТЕМА ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ ДЛЯ ПРОГРАМНО-АПАРАТНОЇ ПЛАТФОРМ ДИСТАНЦІЙНОГО НАВЧАННЯ ЗА ТЕХНОЛОГІЯМИ NLP ТА ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ» 80

4.1. Загальна характеристика стартап-проєкту 80

4.2. Технологічний аудит ідеї проєкту 83

4.3. Аналіз ринкових можливостей запуску стартап-проєкту 84

4.4. Розроблення ринкової стратегії проєкту (в межах конкретного закладу освіти на етапі пілотування) 96

ВИСНОВОК ДО РОЗДІЛУ 4 108

ВИСНОВКИ ДО РОБОТИ 110

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .. **Помилка! Закладку не визначено.**

ДОДАТОК ..... **Помилка! Закладку не визначено.**

ВСТУП

Світова глобалізація, трансформаційні процеси в суспільстві, викликані розвитком інформаційних технологій, досвідом долання пандемічних захворювань, безпекова ситуація в Україні та досвід інших країн – безумовно вносять корективи в формат здійснення освітніх процесів різного рівня і призначення. Все це безумовно стосується національних освітніх процесів. Так багато закладів вищої, передвищої, середньої освіти здійснюють освітній процес в онлайн, дистанційній синхронно-асинхронній формі. Це є і, мабуть, буде реальність не одного року. В контексті цього особливого значення набувають інформаційні технології загалом та платформи дистанційного навчання зокрема.

На теперішній час існує доволі широкий спектр інструментів та платформ дистанційного навчання глобального та локального рівнів, різного ступеня універсальності, інтеграції, спрямованості та функціоналу зокрема: Google Classroom; Moodle; Prometheus; Cisco. Спеціалізовані корпоративні платформи типу: Optima College; ITSTEP Academy; Sigma Software University та багато інших. Не дивлячись на їх багатогранність, відмінності та специфіку вони, безумовно, мають спільні риси, недоліки та переваги. Головне призначення таких платформ дистанційного навчання: систематизоване зберігання і надання доступу до навчально-методичного забезпечення навчальних дисциплін / освітніх компонент / курсів; підтримання зворотного зв’язку на рівні надання звітності, підтримка діалогу «питання – відповіді»; облік і систематизація успішності, активності на платформі тощо. Тобто відбувається сценарій відділеної комунікації із навігаційним процесом доступу до систематизованої інформації [1-7].

В той же час маємо пікове зростання і популяризацію у використанні технологій штучного інтелекту у формуванні відповідей на цікаві та невідомі питання. Це відображається у множині ресурсів генеративного штучного інтелекту різної спрямованості, класу та якості – типу GPT технології тощо. Ці технології, наразі доволі широко застосовуються в освітній сфері для різних

задач практики підготовки і розробки методичного забезпечення освітнього процесу та інше [8-22].

Розвиток технологій обробки природньої мови Natural Language Processing (NLP) обумовив появу інтелектуальних помічників роботів – Chat-Bot в різних прикладних сферах: банкінг; електронна комерція; туризм; медицина та інше.

З огляду на зазначений аналіз вбачається логічним сформувати метод інтеграції в освітні платформи технології інтелектуальної підтримки дистанційного навчання. Зазначене пропонується реалізувати на сучасній процесній складовій технологій обробки природної мови – NLP з використанням гнучкого механізму великих мовних моделей. Результатом має стати програмна компонента, яка в форматі інтерфейсного запиту користувача має звертатись до ресурсів навчальної платформи та надавати релевантну відповідь із переліку методичних матеріалів і тез, близьких за контентом до сформованого запиту.

Отже дисертаційна робота спрямована на **розв'язок науково-прикладного завдання:** розробка методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання.

Мета досліджень: автоматизація процесу інтелектуальної інформаційної підтримки тих хто навчається на платформі дистанційного навчання.

Предмет: методи машинного навчання для обробки природньої мови за технологіями Natural Language Processing (NLP).

Об'єкт: процеси обробки природньої мови за технологіями Natural Language Processing (NLP).

РОЗДІЛ 1
АНАЛІЗ ВІДОМИХ МЕТОДОЛОГІЧНИХ ПІДХОДІВ ДО ПОБУДОВИ ПІДСИСТЕМИ ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ ДЛЯ ПРОГРАМНО-АПАРАТНОЇ ПЛАТФОРМ ДИСТАНЦІЙНОГО НАВЧАННЯ

Дисертаційна робота спрямована на розв’язок науково-прикладного завдання – розробка методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання. Це забезпечить автоматизація процесу інтелектуальної інформаційної підтримки тих хто навчається на платформі дистанційного навчання.

Передбачається, що користувач платформ дистанційного навчання формує текстовий запит природною мовою. При цьому формат запиту вільний, тобто не встановлюється обмежень на вибір питань. За сформованим запитом має відбуватись процес пошуку релевантної інформації в базі даних платформи дистанційного навчання, які вміщує навчально-методичний комплекс навчальних дисциплін / предметів / освітніх компонент.

Навчально-методичні матеріали містять різнорідну інформації, представлену текстуально, аудіо формату природної мови, зображення та відео.

Базова мова запиту – українська, альтернативна – англійська.

Виходячи із зазначеного пропонується:

провести аналіз принципів організації та інтелектуальної інформаційної підтримки користувачів відомих платформ / ресурсів дистанційного навчання;

здійснити аналіз принципів організації пошукових систем із зверненням до глобальних інформаційних мереж з пошуку релевантної запиту інформації;

провести аналіз методологічних підходів з обробки природної мови за принципами NLP із визначення релевантних контенту інформаційних масивів.

Результати аналізу мають дати робочу гіпотезу розв’язку задачі розробка методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання та її формалізацію.

1.1. Аналіз принципів побудови та організації інформаційної підтримки тих, хто навчається на відомих платформах дистанційного навчання

Відомі платформи дистанційного навчання. Розвиток практики застосування методологій і технологій дистанційного навчання призвів до появи доволі великої кількості інструментів та платформ дистанційного навчання. Зокрема можна виділити: Google Classroom; Moodle; Prometheus; Cisco; спеціалізовані корпоративні платформи типу: Optima College; ITSTEP Academy; Sigma Software University та багато інших.

Google Classroom [4,5] – це інструмент від Google Apps for Education, за допомогою якого викладачі можуть швидко створювати й упорядковувати завдання, надавати результати перевірок і спілкуватися зі студентами.

Рисунок 1.1. Платформа Google Classroom

Платформа Google Classroom – об'єднує ксервіси Google, організовані спеціально для навчання. На платформі доступний функціонал: створити свій клас/курс; організувати запис студентів на курс; поширювати серед студентів необхідні навчальні матеріали; запропонувати завдання для студентів; оцінювати завдання студентів і стежити за їхнім прогресом; організувати спілкування із студентами.

Moodle [6,7] – (Modular Object-Oriented Dynamic Learning Environment) – Open Source навчальне середовище / система управління навчанням (LMS), яка надає викладачам, учням та адміністраторам розвинутий набір інструментів для комп'ютеризованого навчання, в тому числі дистанційного.

Рисунок 1.2. Платформа Moodle

Moodle найбільш досконала і поширена в Україні і в світі система такого призначення. На даний момент Moodle вже має близько 400 мільйонів користувачів в усьому світі й продовжує розвиватися темпами, значно швидшими, ніж її конкуренти.

Колосальні можливості Moodle, тим не менш, не включають можливості інтелектуального помічника, що черговий раз доводить певну обмеженість відомих ресурсів дистанційного навчання в цьому аспекті.

Prometheus, Cisco, Optima College; ITSTEP Academy; Sigma Software University – набір корпоративних платформ дистанційного навчання, іноді імплементація інструментів Google Classroom або Moodle з типовим функціоналом: акумуляція матеріалів навчально-методичного комплексу; розмежування доступу студентів; облік звітності / успішності; організація процесу інформаційної взаємодії із студентом.

Не дивлячись на їх багатогранність, відмінності та специфіку вони, безумовно, мають спільні риси, недоліки та переваги. Головне призначення таких платформ дистанційного навчання: систематизоване зберігання і надання доступу до навчально-методичного забезпечення навчальних дисциплін / освітніх компонент / курсів; підтримання зворотного зв'язку на рівні надання звітності, підтримка діалогу «питання – відповіді»; облік і систематизація успішності, активності на платформі тощо. Тобто відбувається сценарій відділеної комунікації із навігаційним процесом доступу до систематизованої інформації.

Жодна із розглянутих платформ поки не містить інтелектуального помічника із пошуком релевантної інформації.

Пошукові сервіси. Дана категорія програмних застосунків варта уваги через її наближення до функціоналу інтелектуального помічника дистанційних курсів – пошук інформації, релевантної запиту в форматі природної мови.

Розвиток технологій глобальної комунікації призвів до появи значної кількості таких систем (див. рис.1.3).

Рисунок 1.3. Пошукові системи

Однак, застосування принципів їх побудови – обробка регулярних виразів, статичні каркаси та іноді методи глибинного навчання можливо на рівні ідей, але таких, що потребують істотного доопрацювання для інтелектуального помічника платформи дистанційного навчання.

В той же час реальні практика сучасності має доволі багато прикладів застосування інтелектуальних чат ботів в сферах надання послуг, електронної комерції, сектор безпеки тощо.

1.2. Аналіз відомих методологічних та технологічних підходів до обробки природньої мови за технологіями NLP та великих мовних моделей

Створення інтелектуального помічника має базуватись на методах обробки природної мови. Це передбачає етап підготовки: фільтрація / токенізація / лематизація; етап лексичного та семантичного аналізу; етап контент / глибинного аналізу – фази NLP (див. рис.1.4. [8-22]).

Рис.1.4. Фази NLP

Зазначений перелік фаз обробки природньої мови застосовується для реалізації конвеєра етапів NLP, представлений на рис.1.5.

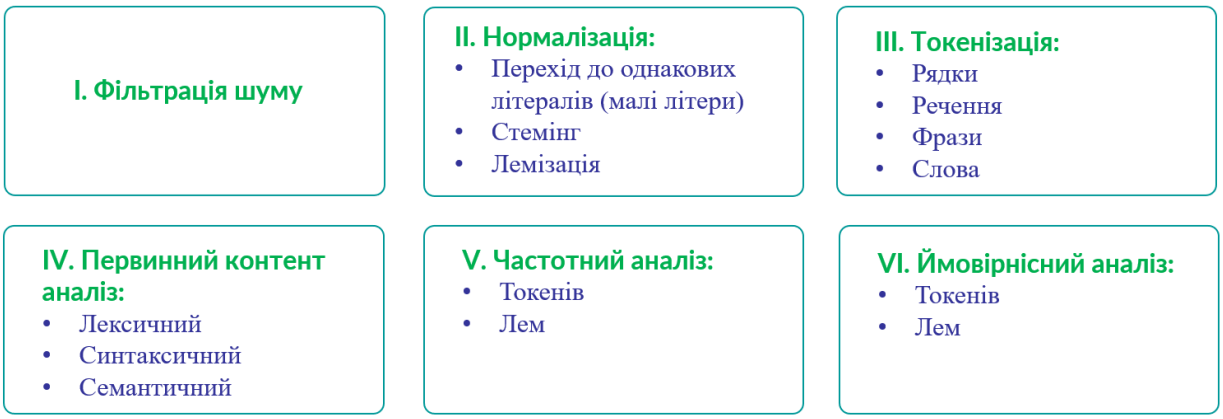


Рис.1.5. Конвеєр етапів NLP

Залежно від завдань обробки природньої мови конвеєр етапів може трансформуватись до конкретики процесів. Наприклад, якщо метою є визначення контенту, то процеси NLP можуть мати перелік, поданий на рис.1.6.

Рис.1.6. Процеси NLP для контент аналізу

Формування процесів NLP для створення інтелектуального помічника на платформі дистанційного навчання саме орієнтовано на контент-аналіз текстового запиту, тому саме цей процесний ряд слід підпорядковувати на аналіз існуючих підходів.

Практика обробки природньої мови (NLP) для визначення контенту має наступні класи методологій машинного навчання (див.рис1.7) [11-15]:

1. Класичні методи машинного навчання:

Метод опорних векторів (support vector machine - SVM);

Метод k-Means / Hard-c-means / k-середніх;

Метод «найближчого сусіда» / k-nearest neighbors algorithm / k-NN.

2. Методи глибинного навчання:

Застосування згорткових / рекурентних нейронних мереж;

3. Методи, засновані на використанні великих мовних моделей за технологіями генеративного штучного інтелекту.

Рис.1.7. Методи машинного навчання для NLP

Метод опорних векторів (support vector machine - SVM) [11-13];

У методі опорних векторів кожний об’єкт відображається як вектор у багатовимірному просторі, кожний вимір якого відповідає інтегрованій характеристиці ознак. Для текстових даних це простір речення запиту.

Для визначення належності до кластера використовується скалярний добутком векторів, звідки визначається кут між вектором ознак (образом) та поточним (об'єктом кластеризації) векторами:

$$cos\,\alpha = \frac{tf_1 tf_2 + idf_1 idf_2}{\sqrt{tf_1^2 + idf_1^2}\sqrt{tf_2^2 + idf_2^2}}, \quad (1.1)$$

де tf_1та idf_1 – координати опорного вектора / образу, а tf_2 та idf_2 – координати вектора об'єкту кластеризації / ідентифікації.

Алгоритм SVM дозволяє ідентифікувати схожість тексту до запиту (фрази) за мінімальним значенням величини $cos\,\alpha$.

Метод k-Means / Hard-c-means / k-середніх [11-13].

Метод k-середніх дозволяє первинно сформувати кластери належності текстових сегментів до лінгвістичного запиту за мірою близькості. Тобто реалізувати кластеризацію інформаційних повідомлень. При цьому обєкти порівняння потребують попередньої обробки до формату векторного простору.

Етапи алгоритму k-середніх.

1. Обрати k точок, які будуть початковими центрами мас кластерів – кількість кластерів;

2. Віднести кожен об'єкт до кластеру з найближчим центром мас;

3. Перерахувати центри мас кластерів згідно з поточним членством.

4. Якщо критерій зупинки алгоритму не виконується – є не пройдені компоненти, повернутися до кроку 2.

5. В якості критерію зупинки зазвичай обирають факт відсутності переходу об'єктів з кластера в кластер на кроці 2 за досягненням точності наближення елементів кластера.

Математична модель алгоритму k-середніх.

Крок 1. Проініціювати початкове розбиття (наприклад, випадковим чином). Вибрати точність δ (використовується в умові завершення алгоритму), проініціювати номер ітерації $l = 0$

Крок 2. Визначити центри кластери за виразом:

$$c_l^{(i)} = \frac{\sum_{j=1}^{d} u_{ij}^{(l-1)} m_j}{\sum_{j=1}^{d} u_{ij}^{(l-1)}}. \quad (1.2)$$

Крок 3. Оновити матрицю розбиття з тим, щоб мінімізувати квадрати помилок, використовуючи відношення:

$$u_{ij}^{(l)} = \begin{cases} 1, \ d\left(m_j, c_i^{(l)}\right) = min\, d\left(m_j, c_i^{(l)}\right) \\ 0, \ \textit{інакше} \end{cases} \quad (1.3)$$

Крок 4. Перевірити умову $\left\|u^{(l)} - u^{(l-1)}\right\| < \delta$. Якщо умова виконується – завершити процес. Інакше – перейти до кроку 2 з ітерацією $l + 1$.

Метод «найближчого сусіда» / k-nearest neighbors algorithm / k-NN [13].

Алгоритм k-NN.

1. Алгоритм починається в довільній точці та поступово «відвідує» кожну найближчу точку, яка ще не була «відвідана».

2. Пункти обходу плану послідовно включаються до маршруту.

3. Причому кожен наступний пункт, що включається до маршруту, повинен бути найближчим до останнього вибраного пункту серед усіх інших, ще не включених до складу маршруту.

4. Алгоритм завершується, коли «відвідано» всі точки.

5. Точками маршруту є множина об’єктів кластеризації.

Математична модель k-NN.

Вхідні дані для множини точок *V* розмірністю *N*.

Вихідні дані: кластери та маршрут *T*, що складається з послідовності відвідування точок множини *V.*

Послідовність роботи алгоритму «найближчого сусіда» включає:

1. Вибрати довільну точку V_1;
2. $T_1 = V_1$;
3. Для $i = 2$ до $i = N$ виконати:
4. Вибрати точку V_i, найближчу до точки T_{i-1};
5. $T_i = V_i$;
6. $T N +1 = V_1$;
7. Кінець алгоритму та прийняття рішення про належність об’єкту до кластеру за мінімумом відстані V_i, T_{i-1}.

Методи глибинного навчання.

Методи глибинного навчання базуються на технологіях штучних нейронних мереж і мають на меті застосування гнучкого механізму використання каркасів – словників. При цьому застосовуються рекурентні або згорткові нейронні мережи у т.ч. і навчені під конкретні задачі.

Методи, засновані на використанні великих мовних моделей [15-22].

Цей підхід базується на використанні генеративних структур штучного інтелекту, призначених для генерації повідомлень природною мовою. Практика їх застосування на теперішній час є доволі розповсюдженою див. рис.1.8 та успішною.

Аналіз принципів побудови і функціонування наведених традиційних рішень дозволяє сформувати певні підсумки. З огляду на необхідність реалізації задач побудови інтелектуального помічника для платформи дистанційого навчання слід опрацювати питання контентного аналізу на методах глибинного навчання та з використанням великих мовних моделе. Це обумовлено тим, що запит на обробку має бути не формалізованим за структурою, але таким, що сприймається за контентом. Тому класичні методи машинного навчання та інші підходи не забезпечать таку гнучкість.

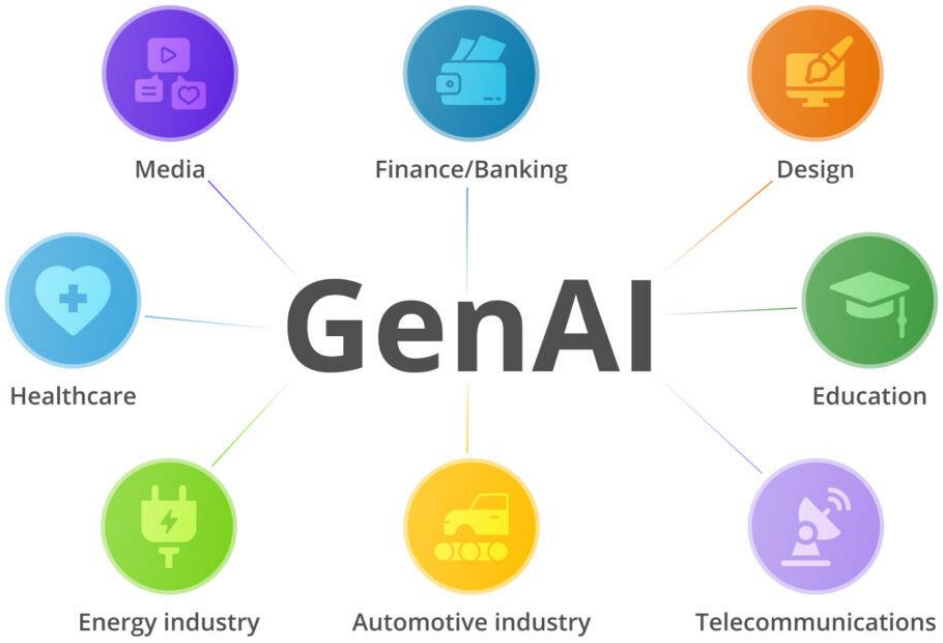


Рис.1.8. Напрямки використання генеративного штучного інтелекту ([https://blog.colobridge.net/uk/2023/11/generative-artificial-intelligence-ua/])

Отже, варто зазначити, що існує доволі поширена практика інтелектуальних помічників в багатьох сферах діяльності соціуму, але їх імплементація в структури курсів дистанційного навчання ще не знайшла широкого застосування, тож актуальним є розробка методу побудови інтелектуальних помічників для систем дистанційного навчання.

Основою для реалізації сутності підходу у формуванні рішень інтелектуальним помічником є функціонал, здатний гнучко реагувати на варіабельність запитів та формування відповідей. Тобто, користувач має отримати релевантну відповідь на питання яке однакове за контентом, але різні за формою.

Так саме ів відповідь, має бути релевантна за контентом, але різна за формою.

В практиці NLP процеси розпізнавання та генерації контенту базуються на словникових моделях – каркасах [8-22].

Відомі два типи моделей каркасів:

статичні, засновані на формуванні переліку слів, термінів та правил їх комбінаторного застосування;

методи, засновані на технологіях глибинного навчання – великі мовні моделі (Large Language Model (LLM)).

Зрозуміло, що за сформованих вимог до функціональності майбутнього програмного застосунку більш продуктивним є шлях використання LLM. Саме ця технологія буде використана для побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформи дистанційного навчання.

Принцип організаційної взаємодії етапів використання LLM демонструється схемами рис.1.9.

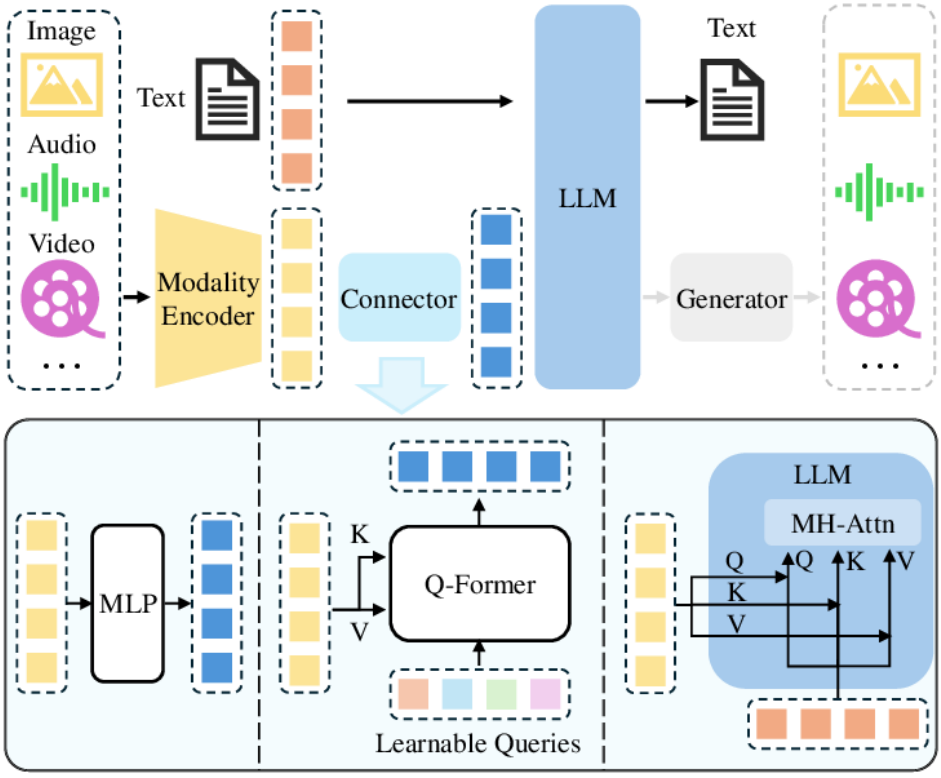


Рис.1.9. Етапів використання LLM ([https://www.researchgate.net/publication/371855457_A_Survey_on_Multimodal_Large_Language_Models])

Типова схема взаємодії з LLM полягає в формуванні запита, який через програмний інтерфейс (application programming interface (API)) потрапляє до

LLM. Запит може бути будь-якого формату, що несе в собі контент. Попередньо відбувається приведення повідомлення до формату, що вимагає структура API в LLM.

Загалом існує три типи перетворень: на основі проекції, на основі запитів і на основі злиття. Перші два типи використовують злиття на рівні маркерів, обробляючи потоки запитів до токенів, які надсилаються разом із текстовими маркерами, тоді як останній тип забезпечує злиття на рівні функцій у LLM.

За сформованими запитами формується генеративна структура розгорнутої відповіді – отримання ключових функцій LLM.

Функціонування LLM потребує потужного апаратного середовища для розгортання, що демонструється схемою, рис.1.10.

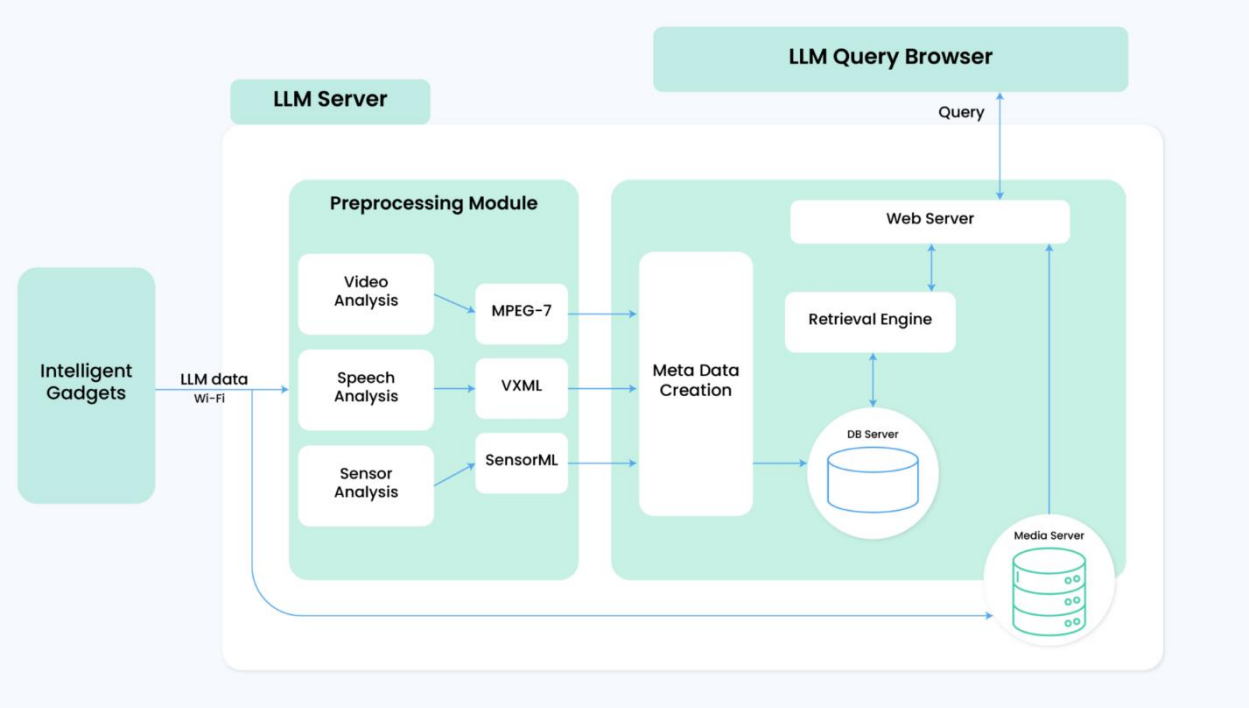


Рис.1.10. Апаратна підтримка LLM, ([https://coralogix.com/ai-blog/exploring-architectures-and-capabilities-of-foundational-llms/])

Як правило LLM розгортається на зовнішньому серверному обладнанні. Запити та відповіді надаються клієнтській частині процесу, що підтримується великою мовною моделлю.

Отже для організації побудови інтелектуального помічника необхідно опрацювати питання я к програмної так і апаратної складових.

Варто також висвітлити основні аспекти архітектурної побудови нейронних мереж з LLM функціоналом. Це дозволить побудувати сттратегію вибору конкретної LLM для імплементації у проект, що реалізується.

На теперішній час найбільш відома трансформаторна архітектура нейронних мереж, що підтримує LLM. Трансформаторна модель є нейронною мережею, яка фіксує контекст і значення шляхом аналізу зв'язків у послідовних даних, таких як слова в реченні. Моделі трансформаторів розрізняють тонкі зв'язки між навіть віддаленими елементами в послідовності за допомогою еволюційних математичних методів, відомих як увага або самоувага [8-10].

Основна ідея роботи моделей трансформаторів включає деталізацію кроків, загальної схеми рис.1.9.

Вбудовування вхідних даних. Передбачає перетворення вхідного речення на числові уявлення, що представляють семантичне значення лексем у реченні.

Позиційне кодування. Кодування токенів на основі їх позицій в реченні, дозволяючи моделі зрозуміти контекстуальні зв'язки між словами.

Самоувага. Передбачає вагове масштабування значення окремих слів у вхідній послідовності. Цей механізм дозволяє моделі зосередитися на релевантних словах і вловити складні зв'язки між ними, а не на «шумових термінах».

Нейронні мережі прямого зв'язку. Використання нейронних мереж прямого зв'язку для покращення інформації, що міститься в представленнях.

Рівень виводу. Кінцевий результат генерується на основі трансформованих уявлень, отриманих за допомогою попередніх кроків.

Наразі найпоширенішими є наступні моделі трансформаторів: GPT; BERT; RAG; Falcon; OPT.

Архітектура GPT

GPT (Generative Pre-trained Transformer) – авторегресійна мовна модель, яка використовує глибоке навчання для створення повідомлень природною мовою.

GPT, являє собою категорію великих мовних моделей, які «вміють» генерувати текст, схожий на розмову людини, пропонуючи можливості для створення вмісту та персоналізовані рекомендації. Архітектура моделі GPT ґрунтується на принципах трансформатора та проходить навчання за допомогою значного текстового корпусу. Завдяки трьом лінійним проекціям, застосованим до вбудовування послідовності, модель ефективно обробляє 1024 токени.

Модель ChatGPT

ChatGPT налаштована спеціально для участі в розмовах природною мовою. Вона чудово генерує контекстуально відповідні та узгоджені відповіді, що робить його особливо вправним у імітації людської взаємодії.

Архітектура BERT

BERT (Bidirectional Encoder Representations Transformers) – це архітектура на основі трансформатора, яка широко використовується в задачах обробки природної мови. BERT складається з кількох рівнів, включаючи нейронні мережі прямого зв’язку та самоконтролю, розроблений для розуміння контексту слова в реченні, враховуючи попередні та наступні слова.

RAG LLMs

RAG (Retrieval-augmented Generation) – архітектурна стратегія, яка розширює можливості великих мовних моделей шляхом плавної інтеграції зовнішніх знань у режимі реального часу у відповіді LLM. Цей інноваційний підхід дозволяє мовним моделям отримувати доступ до найновішої інформації без необхідності перенавчання, використовуючи методи пошуку для створення надійних результатів.

Архітектура Falcon LLM

Архітектура Falcon LLM відноситься до великих мовних моделей домену, які піддаються адаптації або тонкому налаштуванню відповідно до конкретних вимог предметної галузі.

Архітектура OPT

Архітектура OPT включає в себе використання спеціалізованих або специфічних для конкретної програми LLM, ретельно розроблених для конкретних сфер, таких як фінанси, охорона здоров'я, юридичний або технічний сектори.

Технологічні аспекти реалізації NLP процесів.

На теперішній час NLP процеси доволі широко представлені технологічним базисом бібліотек pytnon. Тому реалізація завдань роботи буде здійснено мовою програмування високого рівня pytnon.

Серед бібліотек pytnon, спрямованих на підтримку NLP процесів слід виділити наступні інструменти:

1. Підтримка попередньої обробки повідомлень – препроцесінг за допомогою бібліотек: регулярних вирізів re; NLTK; Spacy; Gensim;

2. Підтримка методів машинного навчання з використанням бібліотек: scikit-learn; SciPy;

3. Підтримка процесів глибинного навчання з використанням бібліотек: tensorflow; keras; PyTorch.

Конкретика імплементації тої чи іншої технології / пакету / бібліотеки / фреймворку буде представлена на етапі розробки підсистеми інтелектуальної підтримки для програмно-апаратної платформи дистанційного навчання.

1.3. Формалізація задачі досліджень та постановка часткових задач досліджень

1.3.1. Загальна постановка задачі дослідження

Таким чином, розробці підлягає метод побудови підсистеми інтелектуальної інформаційної підтримки тих, хто навчається для програмно-апаратної платформ дистанційного навчання, побудована за технологіями NLP та великих мовних моделей.

Прикладним результатом має стати:

архітектура програмно-апаратної платформ дистанційного навчання з інтелектуальною інформаційною підтримкою;

програмна складова підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання.

Робоча гіпотеза про головну ідею роботи полягає у створенні інтелектуального помічника, який буде надавати тим, хто навчається на платформі дистанційного навчання релевантну відповідь на питання, що охоплюють орієнтовні кластери: заняття; оцінювання; навчально-методичні матеріалів; організаційні питання; поточний стан тих, хто навчається; інші питання:

$$Artefact_{BD} = \{Artefact_{BD_i}\}, i = 1 \dots N\ , \tag{1.4}$$

де: N – кількість інформаційних кластерів де вміщуються інформація – артефакти – $Artefact_{BD}$ для релевантної відповіді на пошуковий запит.

Запитання мають формуватися в лінгвістиці природної мови в текстовому форматі:

$$Text_S \tag{1.5}$$

Лінгвістика роботи помічника має охоплювати україномовний сегмент та мати можливість розширення іншими мовами.

Метод, що розробляється має забезпечити формування релевантного запиту (1.5) переліку – підмножини артефактів із генеральної сукупності (1.4), як результат інтелектуального пошуку за мінімальною відстанню за встановленою мірою близькості:

$$Text_R = f\{Text_S \in Artefact_{BD_i}\}, i = 1 \dots N\ , \tag{1.6}$$

Методологічною основою процесу інтелектуальної інформаційної підтримки мають стати підходи в Natural Language Processing – NLP. Передбачається розглянути можливість застосування класичних каркасних моделей та методи глибинного навчання із великими мовними моделями.

1.3.2. Декомпозиція на часткові задачі

Розв’язок науково-прикладного завдання із розробки методу побудови підсистеми інтелектуальної інформаційної підтримки тих, хто навчається для програмно-апаратної платформ дистанційного навчання передбачає реалізацію комплексу взаємопов’язаних часткових задач досліджень.

1. Аналіз принципів побудови та організації інформаційної підтримки тих, хто навчається на відомих платформах дистанційного навчання.

2. Аналіз відомих методологічних та технологічних підходів до обробки природньої мови за технологіями NLP та великих мовних моделей.

3. Розробка методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей.

4. Розробка архітектури програмно-апаратної платформ дистанційного навчання з інтелектуальною інформаційною підтримкою.

5. Розробка програмної складової підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за принципами NLP та великих мовних моделей.

1.3.3. Обмеження та припущення

Вимоги до апаратної платформи – не висуваються. Вимоги до оперативності обробки – відповідає реальному масштабу часу. Точність обробки має забезпечити прийнятний функціонал розробки на рівні релевантних відповідей. Вимоги до рівня реалізації програмної компоненти – демонстраційний прототип. Вимоги до апаратно-програмної побудови – архітектурні рішення. Вимоги до

структурної побудови, рівня інтерфейсу програмного застосунку – не висуваються.

Реалізацію програмної компоненти передбачається здійснити на мові програмування Python та відповідних бібліотек.

Програмний код передбачається оформити з урахуванням PEP – стандартів, супроводжувати документацією (описом сутностей), logging-ом та Unit - тестами. За умов забезпечення прав інтелектуальної власності / конфеденційності буде розглянута можливість доступ до розробленої платформи на ресурсах GitHub.

Буде розглянуто можливість впровадження результати магістерської дисертації на одній з платформ дистанційного навчання.

ВИСНОВОК ДО РОЗДІЛУ 1

Аналіз показав, що існує доволі поширена практика інтелектуальних помічників в багатьох сферах діяльності соціуму, але їх імплементація в структури курсів дистанційного навчання ще не знайшла широкого застосування. Тож актуальним є розробка методу побудови інтелектуальних помічників для систем дистанційного навчання.

Аналіз принципів побудови і функціонування традиційних рішень дозволяє сформувати наступне. З огляду на необхідність реалізації задач побудови інтелектуального помічника для платформи дистанційого навчання слід опрацювати питання контентного аналізу на методах глибинного навчання та з використанням великих мовних моделе. Це обумовлено тим, що запит на обробку має бути не формалізованим за структурою, але таким, що сприймається за контентом. Тому класичні методи машинного навчання та інші підходи не забезпечать таку гнучкість. В тренді задач і рішень слід передбачити програмну імплементації продуктивних моделей з потужною апаратною підтримкою їх функціонування.

Таким чином, розробці підлягає метод побудови підсистеми інтелектуальної інформаційної підтримки тих, хто навчається для програмно-апаратної платформ дистанційного навчання, побудованої за технологіями NLP та великих мовних моделей. Прикладним результатом має стати: архітектура програмно-апаратної платформ; програмна складова підсистеми інтелектуальної підтримки. Робоча гіпотеза головної ідеї роботи полягає у створенні інтелектуального помічника, який буде надавати тим, хто навчається на платформі дистанційного навчання релевантну відповідь на питання, що охоплюють орієнтовні кластери: заняття; оцінювання; навчально-методичні матеріалів; організаційні питання; поточний стан тих, хто навчається (прогрес, рейтинг).

РОЗДІЛ 2
МЕТОД ПОБУДОВИ ПІДСИСТЕМИ ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ ДЛЯ ПРОГРАМНО-АПАРАТНОЇ ПЛАТФОРМ ДИСТАНЦІЙНОГО НАВЧАННЯ ЗА ТЕХНОЛОГІЯМИ NLP ТА ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

2.1. Розробка методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання

Головна ідея методу полягає у наступному. Інформаційне повідомлення, як запит на отримання релевантної інформації із сховища даних – елементи навчально-методичного комплексу платформи дистанційного навчання формується природньою українською мовою у текстовому форматі. Надалі, за необхідності, здійснюється переклад повідомлення на англійську мову, оскільки окремий технологічний базис NLP процесів може не мати підтримки українського словника. Надалі здійснюється комплекс підготовчих операції NLP процесів: токенізація; видалення стоп-слів; лематизація. Із залученням каркасних моделей (у т.ч. організованими за методами глибинного навчання) здійснюється контент аналіз повідомлення. Надалі ключовою процедурою є формування коротенького запиту, вільного від зайвих забруднюючих елементів початкового повідомлення. В результаті формується інтерфейс запиту, якій подається до бази даних, де зберігаються навчальні матеріали. На підставі сформованих запитів надаються релевантні відповіді з пошуку міри найближчої близькості анотованої інформації на кластери: тексу; зображень; відео. Реалізація зазначених процесів може бути описана послідовністю трансформаційних перетворень, що відображає суть запропонованого методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей

Інформаційний запит від користувача платформою дистанційного навчання являє собою текстове повідомлення, як можна позначити як змінна

$$Text_S = \{Words,\ Phrases\}. \quad (2.1)$$

Над вхідним повідомленням (2.1) здійснюється процесний ряд первинних перетворень NLP процесів – f_{NLP}: переклад (за необхідності); токенізація; видалення стоп-слів; лематизація. Результатом є множина компонент, утворених із початкового повідомлення (2.1)

$$Text_S \rightarrow f_{NLP}[Text_S] = Text_{NLP} = \{Token, Lemma\}. \quad (2.2)$$

З використанням великих мовних моделей із множини компонент (2.2) конструюється короткий запит, що за контентом тотожний первинному (2.1):

$$Text_k = f_{LLM}[Text_{NLP}], \quad (2.3)$$

де літералом f_{LLM} – позначено суть процесу конструювання короткого запиту (інтерфейсний запит), отриманий за пітримки технологій великих мовних моделей.

Надалі інтерфейсна фраза (2.3) перетворюється в абстрактне векторне уявлення класичними методами / технологіями векторизації природної мови, що позначено як f_V:

$$Text_V = f_V[Text_k]. \quad (2.4)$$

Об’єктами релевантного пошуку є множина елементів бази даних платформи дистанційного навчання, яка містить артефакти: текстового формату $Artifact_T$; зображення $Artifact_I$; відео $Artifact_V$.

$$BD = \{Artifact_T, Artifact_I, Artifact_V\}. \quad (2.5)$$

Пошук здійснюється як процес оцінювання міри мінімальної відстані між векторними уявленнями пошукової інтерфейсної фрази та анотацій артефактів бази даних навчально-методичного комплексу навчальних дисциплін та інших інформаційних кластерів системи дистанційного навчання. Для цього, наприклад, може бути використана технологія двошарової нейронної мережі Word2Vec [8-22]. Результатом пошуку є перелік релевантних запиту артефактів:

$$Result_{Artifact} = f_{\text{Word2Vec}}[Artifact_T, Artifact_I, Artifact_V]. \quad (2.7)$$

Множина релевантних відповідей може також інтерпретуватись в мовну / текстову коротку відповідь з використанням великої мовної моделі – LLM (2.8).

$$Result_{text} = f_{LLM}\left[Result_{Artifact}\right]. \quad (2.8)$$

За потреби на кожному етапі можливо організувати протокол спілкування користувач (студент / учень) – інтелектуальний помічник. При цьому ітераційно повторюються етапи із залученням моделей (2.3) – (2.8).

Порядок організаційної взаємодії етапів, представлених виразами (2.1) – (2.8) можна подати структурною схемою рис.2.1.

Таким чином, вирази та позначення (2.1) – (2.8) та схема рис.2.1 – розкривають суть методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей.

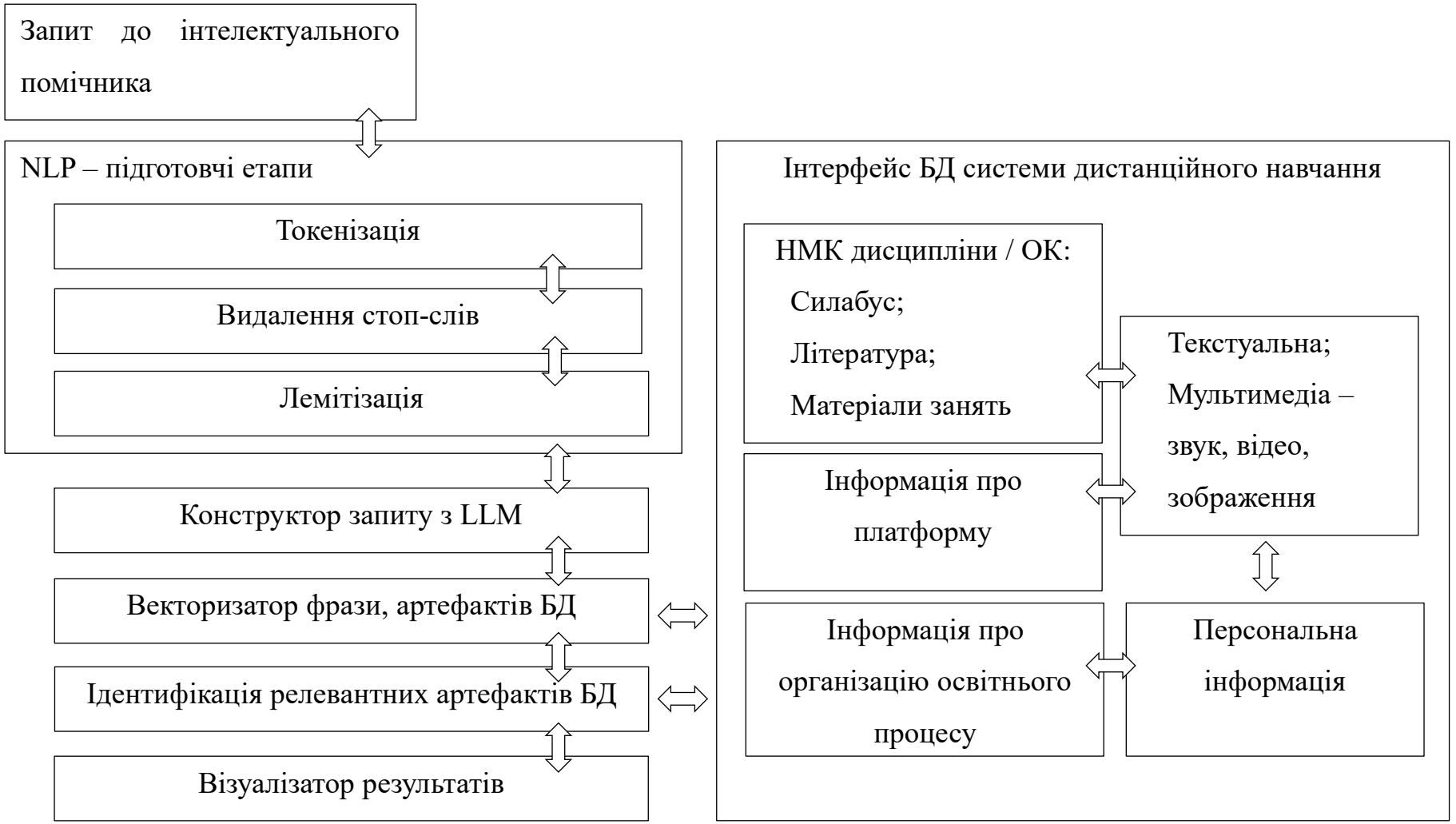


Рис.2.1. Метод побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання

Метод базується на послідовному визначенні контенту інформаційного повідомлення з використанням великих мовних моделей та відрізняється від аналогів формування інтерфейсного (короткого) запиту до методів доступу к різнорідної інформації, яка становить суть методичного забезпечення освітнього процесу платформи дистанційного навчання. Короткий / інтерфейсний запит формується із застосуванням великої мовної моделі.

Результатом є вичерпана релевантна відповідь про інформаційні кластери методичного забезпечення, що підтримують суть відповіді на запит, сформований природною мовою.

2.2. Архітектура програмно-апаратної платформ дистанційного навчання з інтелектуальною інформаційною підтримкою

Апаратно-програмна компонента системи повинна забезпечити фізичну підтримку розрахункових процесів для інтелектуального помічника. Разом із тим, структура апаратної компоненти інтелектуальної підтримки має інтегруватись в апаратно-програмне середовище компанії, що надає освітні послуги у дистанційному форматі. Тому до архітектури програмно-апаратної платформ дистанційного навчання з інтелектуальною інформаційною підтримкою можна сформувати таки вимоги:

1. Надійність та захищеність збереження корпоративних інформаційних кластерів;
2. Інтеграція помічника до корпоративних сегментів через інтерфейс програмних запитів з апаратною підтримкою;
3. Підтримка високопродуктивних обчислень апаратною компонентною, що забезпечує роботу інтелектуального помічника;
4. Можливості резервування та зберігання усіх інформаційних кластерів освітньої платформи;
5. Документування усіх подій освітньої платформи та запитів до інтелектуального помічника.

Виходячи із зазначеного можливо сформувати архітектуру програмно-апаратної платформи дистанційного навчання з інтелектуальною інформаційною підтримкою, яка подана на рис.2.2.

Кожна компонента має інформаційне і функціональне навантаження, яке спрямоване на комплекс функцій надання освітніх послуг в дистанційному форматі.

Серверна (back-end) компонента інтелектуального помічника виконує головні його функції за рахунок імплементації розробленого методу методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей. Апаратна складова має підтримувати адекватне та оперативне функціонування LLM моделі.

Інтерфейс back-end компонента інтелектуального помічника має віддалене (за клієнт-серверною) архітектурою з користувачами front-end компоненти. На схемі рис.2.2 front-end компонента інтелектуального помічника представлена на множині автоматизованих робочих місць (АРМ) користувачів. Організація розподіленого інформаційного обміну здійснюється з використанням технічного обладнання, протоколів та ресурсів глобальних інформаційних мереж (WAN) – Internet мережі.

Для пошуку релевантних відповідей на запит користувача back-end компонента інтелектуального помічника взаємодіє з інформаційними ресурсами платформи дистанційного навчання. Ці ресурси, як правило, розташовані на корпоративних серверах для забезпечення конфіденційності та / або як головна складова чи резервна на серверах зовнішньої інформаційної підтримки.

Взаємодія між інтелектуальним помічником та зовнішніми до нього ресурсами здійснюється локальними (LAN) та глобальними комп’ютерними мережами.

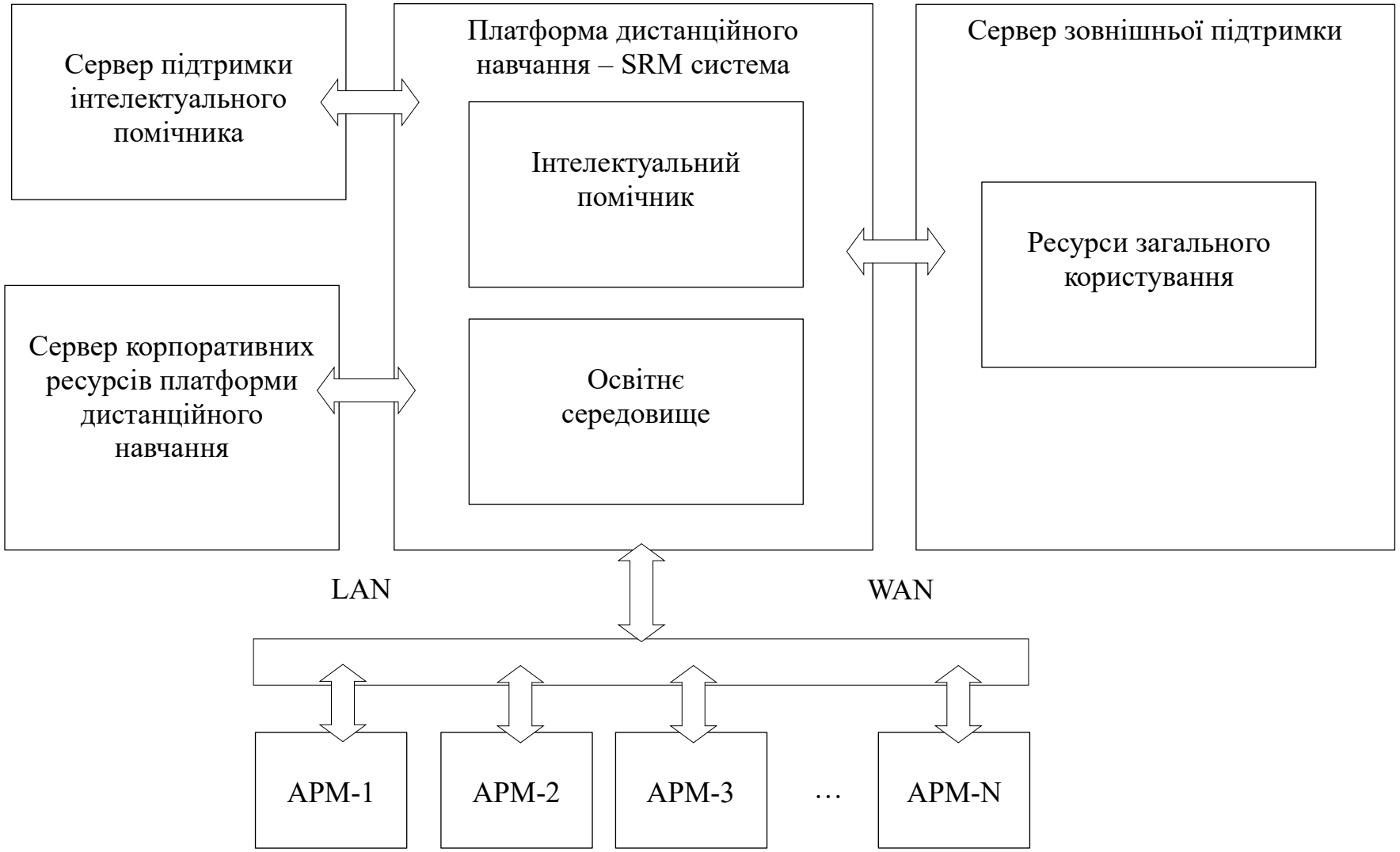


Рис.2.2. Архітектура програмно-апаратної платформ дистанційного навчання з інтелектуальною інформаційною підтримкою

Таким чином, архітектура програмно-апаратної платформи дистанційного навчання з інтелектуальною інформаційною підтримкою відображає класичні принципи розподілених комп'ютерних мереж, адаптовані з урахуванням запропонованого методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання.

2.3. Програмна компонента підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за принципами NLP та великих мовних моделей

Програмна компонента відображає суть запропонованого методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання і розгортається на програмно-апаратної платформ з архітектурою рис.2.2.

Головний функціонал програмної компоненти полягає у наданні тим, хто навчається на платформі дистанційного навчання релевантну відповідь на питання, що охоплюють орієнтовні кластери: заняття; оцінювання; навчально-методичні матеріалів; організаційні питання; поточний стан тих, хто навчається; складно-формалізовані питання.

Запитання мають формуватися в лінгвістиці природньої мови в друкованому форматі повідомлень.

Лінгвістика роботи помічника має охоплювати україномовний сегмент та мати можливість розширення іншими мовами.

2.3.1. Інженерія вимог

Підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання розроблено з метою впровадження принципів інноваційної людино-машинної взаємодії користувачів в межах реалізації освітнього процесу. Підсистема інтелектуальної пітримки, дефакто є

інтелектуальним помічником, призначеним забезпечити оперативний, релевантний та персоналізований доступ до інформації для студентів / учнів, батьків та викладачів. В основі проєкту покладені технології обробки природної мови (Natural Language Processing, NLP), зокрема великі мовні моделі (LLM) та потужні векторні бази даних, що дозволяє досягти високої ефективності та релевантності / точності відповідей. Ключовою особливістю є інтеграція інтелектуального помічника з комплексною базою даних платформи дистанційного навчання, яка містить навчальні матеріали, розклади занять, інформацію про успішність та іншу контекстну інформацію. Ця інтеграція, разом з механізмами ретриверу та реранкінгу, забезпечує швидкий та релевантний пошук відповідей на запити користувачів.

Головні функції інтелектуального помічника підсистеми інтелектуальної підтримки платформи дистанційного навчання.

 Забезпечення оперативного доступу до навчальних матеріалів. Слухачі / студенти / учні та викладачі / вчителі – об’єкти і суб’єкти освітнього процесу мають отримати оперативний доступ до контекстних занять, презентацій, відеоматеріалів та інших ресурсів.

 Спрощення вирішення організаційних питань. Інтелектуальний помічник має забезпечити надання інформації про розклад занять, дати контрольних робіт, зміни в розкладі та іншу інформацію організаційного характеру, зменшуючи навантаження на адміністрацію закладу освіти, що здійснює професійну діяльність на платформі дистанційного навчання.

 Автоматизація відповіді на типові запитання. Більшість типових запитів від суб’єктів та об’єктів освітнього процесу мають оброблятись автоматично, звільняючи час структур адміністративного управління.

 Надання підтримки суб’єктам освітнього процесу. Інтелектуальний помічник має надавати допомогу у розборі складних, надавати додаткові матеріали та направляти їх на відповідні ресурси платформи дистанційного навчання.

 Персоналізований доступ до інформації. Суб’єкти та об’єкти освітнього процесу мають отримати доступ до персоналізованої інформації: поточний рейтинг, розклад занять, дедлайни звітності, індивідуальні повідомлення тощо.

 Адаптація до української освітньої системи. Функціонал підсистеми інтелектуальної підтримки має враховувати принципи, особливості, законодавчу базу, стандарти національної – Української системи освіти.

2.3.2. Архітектурне проектування програмної компоненти

У цьому розділі детально описано процес розробки, налаштування середовища та розгортання програмного проекту інтелектуального помічника. Програмна компонента забезпечує автоматизацію взаємодії з користувачами платформи дистанційного навчання, надаючи релевантну інформацію на основі запитів.

В основу програмної компоненти покладено архітектура RAG (Retrieval-Augmented Generation), що поєднує пошук інформації у спеціалізованій базі знань та генерацію відповідей за допомогою великої мовної моделі (LLM).

Ключовими технологічними компонентами програмної реалізації є наступне.

1. **Векторна база даних Milvus.** Використовується для зберігання векторних представлень (ембедингів) навчальних матеріалів та ефективного пошуку семантично близької інформації до запиту користувача.
2. **Реляційна база даних MariaDB.** Слугує первинним сховищем структурованих платформи дистанційного навчання, зокрема занять та інших навчальних матеріалів, які потім векторизуються та завантажуються у Milvus.
3. **Велика мовна модель Gemini (Google).** Відповідає за генерацію кінцевої відповіді користувачеві на основі контексту, отриманого з Milvus.
4. **Модуль пошуку та ранжування (Retriever & Reranker).** Компонент, що обробляє запит користувача, здійснює пошук релевантних даних у Milvus та потенційно оптимізує порядок знайдених результатів перед передачею їх до LLM.

Архітектура взаємодії компонент програмної реалізації підсистеми інтелектуальної підтримки для платформи дистанційного навчання подана структурною схемою рис.2.3.

Функціонування інтелектуального помічника здійснюється за функціоналом програмної компоненти інтелектуальної підтримки платформи дистанційного навчання. Активацію роботи інтелектуального помічника здійснює користувач в організації процесу формування запиту в текстовій формі.

Процес обробки запиту відбувається наступним чином:

1. Запит користувача надходить через API до Обробника запитів.

2. Обробник запитів передає запит до Модуля пошуку та ранжування.

3. Модуль пошуку формує векторне представлення запиту та здійснює пошук найбільш релевантних векторів у векторній базі даних Milvus. Дані в Milvus попередньо завантажуються та оновлюються з бази знань закладу освіти (MariaDB) після відповідної векторизації.

4. Модуль пошуку отримує релевантні фрагменти тексту (контекст) з Milvus та передає їх разом із початковим запитом до великої мовної моделі Gemini.

5. LLM Gemini генерує відповідь на основі отриманого запиту та контексту.

6. Згенерована відповідь повертається до обробника запитів, який надсилає її користувачеві через API.

Таким чином результатом є релевантна запиту і вмісту бази даних навчальних матеріалів відповідь.

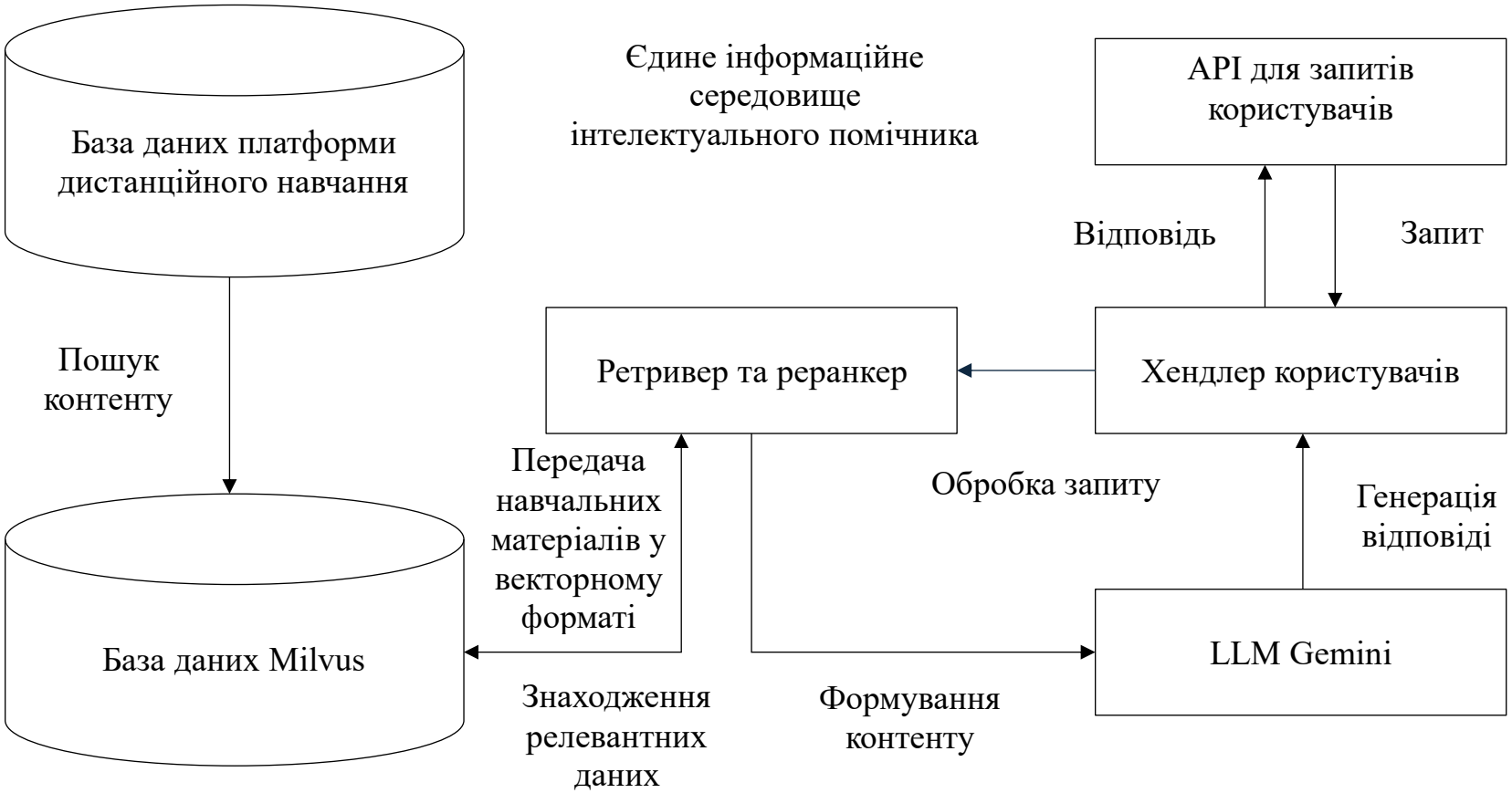


Рис.2.3. Архітектура взаємодії компонент програмної реалізації підсистеми інтелектуальної підтримки

Ключові архітектурні та технологічні рішення проекту

Реалізація програмної компоненти інтелектуального помічника базується на низці ключових архітектурних та технологічних рішень, спрямованих на забезпечення високої релевантності відповідей, ефективності роботи та адаптації до специфіки платформи дистанційного навчання. До основних особливостей проекту належать:

1. **Застосування архітектури RAG (Retrieval-Augmented Generation).** Це фундаментальне рішення, що дозволяє подолати обмеження стандартних LLM. Замість генерації відповіді виключно на основі внутрішніх знань моделі, система спочатку здійснює пошук (Retrieval) найбільш релевантної інформації у власній, спеціалізованій базі знань (Milvus), а потім використовує цю інформацію як контекст для генерації (Generation) відповіді за допомогою LLM (Gemini). Такий підхід значно підвищує точність, актуальність та фактологічну обґрунтованість відповідей, мінімізуючи ризик надання некоректної або нерелевантної інформації.

2. **Інтеграція структурованої бази даних (MariaDB) та векторної бази знань (Milvus).** Унікальність рішення полягає у використанні MariaDB як первинного, авторитетного джерела структурованих даних платформи дистанційного навчання у форматі методичного забезпечення освітнього процесу. Ці інформаційні потоки / дані проходять процес векторизації та завантажуються у Milvus. Це забезпечує те, що векторна база знань, яка використовується для семантичного пошуку, формується безпосередньо на основі офіційних та актуальних матеріалів платформи. Такий двоступеневий підхід до управління знаннями і даними гарантує високу достовірність інформаційного підґрунтя інтелектуального помічника.

3. **Використання спеціалізованої векторної бази даних (Milvus).** Вибір Milvus обумовлений її високою продуктивністю та масштабованістю при роботі з великими обсягами векторних даних. Milvus надає оптимізовані механізми індексації та пошуку високовимірних векторів, що є критично важливим для

забезпечення низької затримки відповіді інтелектуального помічника при виконанні семантичного пошуку. Підтримка різних метрик подібності та типів індексів дозволяє гнучко налаштовувати процес пошуку для досягнення оптимальної релевантності.

4. **Залучення сучасної LLM (Gemini) з можливістю доналаштування.** Використання потужної моделі Gemini забезпечує високу якість генерації тексту – відповіді є природними, зв'язними та стилістично відповідними. Важливою особливістю є проведений процес доналаштування (fine-tuning) моделі на специфічних для конкретної платформи дистанційного навчання. Це дозволяє адаптувати поведінку LLM до конкретної предметної області та підвищити точність відповідей на специфічні запити користувачів.

5. **Контейнеризація за допомогою Docker та Docker Compose.** Застосування технологій контейнеризації для розгортання всіх компонент системи (Milvus, Attu, Nginx, основний застосунок) є сучасною інженерною практикою. Це забезпечує ізоляцію залежностей, відтворюваність середовища на різних машинах, спрощує процес розгортання, оновлення та масштабування системи. Рекомендований спосіб запуску через docker-compose значно знижує складність адміністрування системи.

6. **Структурований конвеєр підготовки даних.** Наявність окремого скрипта для вилучення даних з `MariaDB`, їх векторизації та завантаження в `Milvus` свідчить про системний підхід до управління базою знань та даних. Це дозволяє автоматизувати процес оновлення знань інтелектуального помічника при змінах у першоджерелі (наприклад, при оновленні навчальних програм чи матеріалів), забезпечуючи актуальність інформації, що надається.

Ці рішення в комплексі дозволяють створити ефективний, високоточний та адаптований до потреб конкретного навчального закладу інструмент для взаємодії з користувачами, що базується на передових технологіях обробки природної мови та управління даними.

Використані технології та інструменти.

Для реалізації програмної компоненти інтелектуального помічника було використано комплекс сучасних технологій та програмних інструментів, кожен з яких відіграє важливу роль у функціонуванні та архітектурі системи. Нижче наведено опис ключових технологій.

1**. Мовна модель Google Gemini.** У даному проєкті Gemini використовується як основний компонент для генерації відповідей користувачам. Модель отримує запит користувача та релевантний контекст, знайдений у векторній базі даних, і на основі цих даних формулює природну, змістовну та стилістично відповідну відповідь. Модель також пройшла процес доналаштування для кращої адаптації до специфіки запитів конкретної платформи дистанційного навчання.

2. **Векторна база даних Milvus.** Високопродуктивна система керування векторними базами даних з відкритим кодом. Milvus застосовується для зберігання, індексації та ефективного пошуку високовимірних векторних представлень (ембедингів) навчальних матеріалів платформи дистанційного навчання. Це ядро механізму пошуку (Retrieval) в архітектурі RAG, що дозволяє швидко знаходити семантично близькі до запиту користувача фрагменти тексту для формування контексту.

3. **Мова програмування Python.** Високорівнева інтерпретована мова програмування загального призначення. Python обрано як основну мову для розробки бекенд-частини програмної компоненти інтелектуальної підтримки. На Python реалізовано логіку обробки запитів користувачів (API хендлери), взаємодію з базами даних (Milvus, MariaDB), інтеграцію з API моделі Gemini, а також скрипти для підготовки та завантаження даних у Milvus.

5. **Система контейнеризації Docker та Docker Compose.** Docker – це платформа для розробки, доставки та запуску додатків у контейнерах. Docker Compose – інструмент для визначення та запуску багатоконтейнерних Docker-додатків. У цьому проєкті Docker та Docker Compose використовуються для пакування всіх компонентів системи (Milvus, Attu, Nginx, Python-застосунок) та

їх залежностей в ізольовані контейнери. Це забезпечує легкість розгортання, відтворюваність середовища та спрощує управління всією системою як єдиним цілим.

6. **Платформа NVIDIA CUDA та драйвери NVIDIA.** CUDA (Compute Unified Device Architecture) – це платформа паралельних обчислень та програмна модель, розроблена NVIDIA для графічних процесорів (GPU). Разом із відповідними драйверами NVIDIA, CUDA дозволяє використовувати обчислювальні потужності GPU для прискорення задач, що потребують значних обчислень, таких як операції з векторами у Milvus (індексація, пошук) та потенційно робота з моделями машинного навчання.

7. **Інтерфейс командного рядка Google Cloud (gcloud CLI).** Інструмент командного рядка для взаємодії з сервісами Google Cloud Platform. У даному проєкті `gcloud CLI` використовується переважно для автентифікації серверного застосунку (`gcloud auth application-default login`), що дозволяє йому безпечно звертатися до API сервісів Google, зокрема до API моделі Gemini.

8**. Веб-інтерфейс Attu.** Графічний інструмент з відкритим кодом для управління та візуалізації даних у Milvus. Attu використовується під час розробки та адміністрування системи для перегляду стану колекцій даних, управління індексами, виконання тестових запитів та моніторингу роботи екземпляра Milvus.

9. **Веб-сервер Nginx.** Високопродуктивний веб-сервер та зворотний проксі-сервер. У рекомендованій конфігурації розгортання за допомогою Docker Compose, Nginx може використовуватися як зворотний проксі перед основним компонентом інтелектуального помічника, для обробки статичних файлів або для інших завдань, пов'язаних із мережевою взаємодією.

Вибір цих технологій обумовлений їхньою ефективністю, масштабованістю, доступністю (багато з них мають відкритий код) та відповідністю сучасним підходам до розробки систем на основі штучного інтелекту та великих даних.

2.3.3. Програмна реалізація

Програмна реалізація підсистеми інтелектуальної підтримки здійснено мовою високого рівня python із відповідними бібліотеками [23-37]. Текст програмного коду наведено у додатку.

Архітектура проекту програмного коду підсистеми інтелектуальної підтримки для платформи дистанційного навчання подана на рис.2.4.

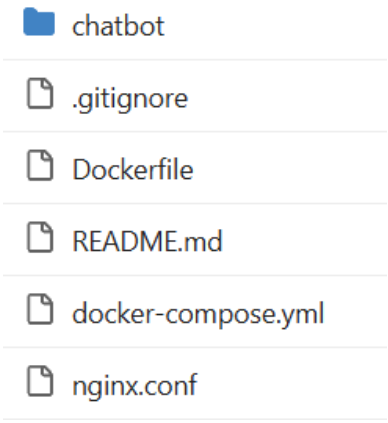


Рис.2.4. Архітектура проекту програмного коду – загальна

Структура проекту розроблена з урахуванням принципів модульності, повторного використання коду та відокремлення відповідальностей (Separation of Concerns). Зазначене сприяє гнучкості проекту за рахунок впровадження відкритої архітектури, підтримці та подальшому розвитку.

Архітектуру проекту можливо розглядати на двох рівнях:

загальна організація проекту на рівні файлової системи;

внутрішня структура основного застосунку.

Загальна структура проекту.

На кореневому рівні проекту розміщені файли та директорії, що відповідають за конфігурацію, документацію, контейнеризацію та запуск системи:

1. `chatbot/`: Ключова директорія, що містить весь вихідний код Python основного еомпонента інтелектуального помічника.

2. `.gitignore`: Стандартний файл для системи контролю версій Git, що визначає файли та директорії, які слід ігнорувати (наприклад, кеш, логи, віртуальні середовища).

3. `Dockerfile`: Специфікація для автоматичної збірки Docker-образу застосунку `chatbot`. Включає інструкції для копіювання коду, встановлення залежностей з `requirements.txt` та визначення точки входу для запуску контейнера.

4. `README.md`: Файл документації у форматі Markdown, що містить загальний опис проекту, інструкції зі встановлення, налаштування та використання.

5. `docker-compose.yml`: Файл конфігурації Docker Compose. Визначає набір сервісів, що складають систему (застосунок `chatbot`, база даних `Milvus`, інтерфейс `Attu`, веб-сервер `Nginx` та інші можливі залежності), їхні взаємозв'язки, мережеві налаштування та налаштування томів для зберігання даних. Є центральним елементом для спрощеного розгортання та управління всією системою.

6. `nginx.conf`: Конфігураційний файл для веб-сервера `Nginx`. Використовується в рамках Docker Compose для налаштування поведінки Nginx, найімовірніше, як зворотного проксі (reverse proxy) перед основним Python-застосунком, що обробляє HTTP-запити.

Внутрішня структура програмної компоненти(`chatbot/`).

Внутрішня архітектура містить основний код програмної компоненти та відповідає архітектурним поділам, поданим на рис.2.5.

Основний код програмної компоненти вміщується в директорії `chatbot/` та має модульну структуру, що відображає логічні компоненти системи.

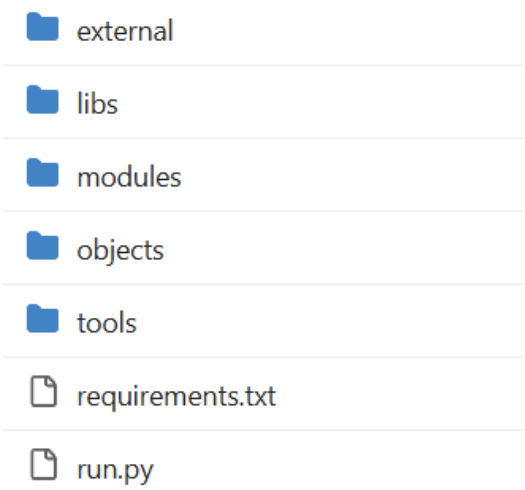


Рис.2.5. Архітектура проекту програмного коду – головна

1. `run.py`: Головний скрипт, точка входу для запуску Python-застосунку.

2. `requirements.txt`: Стандартний файл, що перелічує всі зовнішні Python-бібліотеки та їх версії, необхідні для коректної роботи проекту.

3. `external/`: Директорія для зовнішніх файлів конфігурації та даних.

3.1. `config.json`: Основний конфігураційний файл застосунку у форматі JSON, що містить параметри підключення до баз даних (`Milvus`, `MariaDB`), API-ключі, налаштування логування та інші параметри роботи системи.

3.2. `logs/`: Директорія для зберігання файлів логів роботи застосунку.

4. `modules/`: Містить основні функціональні модулі, що реалізують бізнес-логіку застосунку:

4.1. `database/`: Модулі для взаємодії з реляційною базою даних (`MariaDB`), де зберігаються вихідні дані закладу освіти.

4.2. `milvus/`: Модулі, що інкапсулюють логіку роботи з векторною базою даних `Milvus` (підключення, пошук векторів, вставка даних).

4.3. `router/`: Модулі, що відповідають за маршрутизацію вхідних API-запитів до відповідних обробників (контролерів) з використанням веб-фреймворку FastAPI.

5. `tools/`: Директорія з інструментами та реалізацією ключових компонентів RAG-системи:

5.1. `gemini/`: Модулі для інтеграції з великою мовною моделлю Google Gemini (формування запитів до API, обробка та парсинг відповідей).

5.2. `retriever/`: Реалізація логіки пошуку релевантного контексту. Включає модулі для генерації векторних представлень (`embedding.py`) та безпосередньо виконання пошуку (`retriever.py`) у `Milvus`.

5.3. `reranker/`: Модулі для реалізації логіки переранжування знайдених результатів для підвищення релевантності контексту, що подається в LLM.

6. `objects/`: Містить визначення класів та структур даних, що моделюють сутності системи:

6.1. `responses/`: Класи для стандартизації та структурування відповідей, що повертаються через API.

6.2. `prompts/`: Класи або утиліти для конструювання та управління запитами (промптами) до LLM Gemini.

6.3. `configs/`: Класи для представлення та валідації конфігураційних параметрів.

6.4. Інші файли (`queries`, `status_messages`, `ukrainian_date_names`, `user_session_objects`): Представляють різні об'єкти даних, специфічні для логіки застосунку.

7. `sessions/`: Модулі, відповідальні за управління станом діалогу та сесіями користувачів.

8. `libs/`: Директорія для допоміжних бібліотек або утиліт загального призначення, наприклад, кастомний модуль для налаштування та ведення логів (`logger/`).

9. Наявність файлів `__init__.py` у піддиректоріях вказує на використання стандартної практики Python для визначення пакетів, що дозволяє імпортувати модулі між різними частинами проекту.

Запропонована та реалізована структурована організація програмного коду відповідає вимогам і властивостям відкритої архітектури програмної компоненти. Це забезпечує його високу читабельність, полегшує тестування

окремих компонентів, спрощує командну роботу над проектом та створює основу для подальшого розширення функціональності системи.

Налаштування середовища та залежностей

Для успішного розгортання програмної компонента підсистеми інтелектуальної підтримки та її функціонування необхідно забезпечити відповідне програмне та апаратне середовище. Нижче наведено основні вимоги та кроки для налаштування.

Апаратні та програмні вимоги.

1. **Операційна система.** Рекомендовано Ubuntu 22.04 LTS або сумісний дистрибутив Linux.

2. **Графічний процесор (GPU).** Наявність GPU виробництва NVIDIA з підтримкою CUDA є бажаною для прискорення обчислень, пов'язаних із векторними базами даних та моделями машинного навчання.

3. **NVIDIA Drivers.** Встановлені актуальні драйвери NVIDIA (наприклад, версія 565 або новіша). Перевірка наявності та встановлення здійснюється стандартними засобами ОС (наприклад, `sudo apt search nvidia-driver`, `sudo apt install nvidia-driver-<version>`, `nvidia-smi`).

4. **CUDA Toolkit.** Встановлена платформа CUDA (рекомендована версія 12.6). Інсталяція виконується за допомогою офіційного інсталятора від NVIDIA (`.run` файл). Важливо під час інсталяції вимкнути опцію встановлення драйвера, якщо він вже був встановлений раніше.

5. **Docker та Docker Compose.** Необхідні для контейнеризації та оркестрації сервісів (Milvus, Attu, застосунок чат-бота). Рекомендовані версії: Docker 28.0.1+, Docker Compose 1.29.2+. Встановлення здійснюється згідно з офіційною документацією Docker для відповідної ОС.

6. **Python.** Встановлений інтерпретатор Python (рекомендована версія 3.10).

7. **Google Cloud CLI.** Необхідний для автентифікації програмної компоненти при використанні API сервісів Google Cloud (зокрема, Gemini).

Кроки налаштування середовища.

1. **Встановлення драйверів NVIDIA.** Виконати пошук та встановлення сумісної версії драйвера за допомогою пакетного менеджера ОС. Після встановлення рекомендовано перезавантажити систему та перевірити коректність встановлення командою `nvidia-smi`.

2. **Встановлення CUDA Toolkit.** Завантажити інсталяційний `.run` файл для потрібної версії CUDA (12.6) з офіційного сайту NVIDIA. Запустити інсталятор з правами суперкористувача (`sudo sh <cuda_installer_file.run>`) та дотримуватись інструкцій, не встановлюючи драйвер повторно.

3. **Налаштування змінних середовища для CUDA.** Додати шляхи до встановленої CUDA до системних змінних `PATH` та `LD_LIBRARY_PATH`. Це зазвичай робиться шляхом редагування файлу конфігурації оболонки (наприклад, `~/.bashrc`) та додавання наступних рядків (шлях може відрізнятися):

```bash
export CUDA_HOME="/usr/local/cuda-12.6"
export LD_LIBRARY_PATH=${CUDA_HOME}/lib64:$LD_LIBRARY_PATH
export PATH="$PATH:$CUDA_HOME/bin"
```

Після збереження змін необхідно оновити поточну сесію оболонки (`source ~/.bashrc`).

4. **Встановлення Google Cloud CLI.** Дотримуючись офіційної документації Google Cloud, додати репозиторій `gcloud`, оновити список пакетів та встановити `google-cloud-cli`. Після встановлення необхідно виконати ініціалізацію (`gcloud init`) та автентифікацію додатку (`gcloud auth application-default login`) для доступу до API Google.

5. **Встановлення Docker та Docker Compose.** Встановити Docker Engine та Docker Compose, слідуючи офіційним інструкціям для Ubuntu на сайті Docker.

Розгортання системи.

Рекомендований спосіб розгортання системи передбачає використання `Docker Compose` для управління всіма необхідними сервісами (Milvus, Attu, Nginx, застосунок чат-бота) у вигляді контейнерів.

Процедура розгортання за допомогою Docker Compose.

1. **Клонування репозиторію.** Отримати вихідний код проєкту з репозиторію.

2. **Налаштування конфігурації.** Переконатися, що у файлі `docker-compose.yml` у секції `volumes` для сервісу `chatbot` вказано коректний абсолютний шлях до файлу `application_default_credentials.json`, отриманого після автентифікації `gcloud`.

3. **Збірка та запуск контейнерів.** Запустити процес збірки образів (якщо вони не існують) та запуск усіх сервісів у фоновому режимі за допомогою команди:

```shell
sudo docker-compose up --build -d
```

Після успішного виконання цих кроків, компоненти системи будуть розгорнуті та доступні за відповідними мережевими портами, вказаними в конфігурації `docker-compose.yml` (за замовчуванням: Milvus - 19530, Attu - 8000, веб-інтерфейс моніторингу Milvus - 9091).

Альтернативний спосіб розгортання.

Існує також можливість ручного розгортання компонентів.

Milvus та Attu. Можна встановити за допомогою офіційних скриптів (`standalone_embed.sh`) або Docker-команд, наданих у документації Milvus та Attu. Застосунок інтелектуального помічника: Можна запустити безпосередньо на хост-машині:

1. Клонувати репозиторій застосунку.

2. Створити та активувати віртуальне середовище Python (`python3 -m venv venv`, `source venv/bin/activate`).

3. Встановити необхідні залежності (`pip install -r requirements.txt`).

4. Запустити основний скрипт застосунку, вказавши шлях до конфігураційного файлу (`python main.py --config external/config.json`).

Наповнення бази знань.

Ключовим етапом підготовки системи є наповнення векторної бази даних Milvus даними з MariaDB. Для цього передбачено окремий скрипт.

Процедура наповнення бази даних.

1. **Клонування репозиторію.** Отримати репозиторій зі скриптом наповнення.

2. **Встановлення залежностей.** Перейти до директорії репозиторію, створити та активувати віртуальне середовище Python, після чого встановити необхідні пакети:

```shell
pip install networkx==2.8.8
pip install -r requirements.txt
pip install milvus-model==0.2.5
```

3. Запуск скрипта. Виконати скрипт `run.py`, вказавши діапазон ідентифікаторів уроків для обробки та шлях до конфігураційного файлу:

```shell
python run.py --min <min_id> --max <max_id> --config <path_to_config.json>
```

Скрипт підключається до `MariaDB`, отримує текстові дані уроків, генерує для них векторні представлення за допомогою відповідної моделі (`milvus-model`) та завантажує отримані вектори разом із метаданими до `Milvus`.

Тестування та документація API

Після успішного розгортання системи, доступ до інтерактивної документації API (Swagger UI) та можливість тестування ендпоінтів надається через веб-браузер за адресою:

`http://<хост_сервера>:<порт_застосунку>/docs`

де `<хост_сервера>` - IP-адреса або доменне ім'я сервера, де розгорнуто застосунок, а `<порт_застосунку>` - порт, вказаний у конфігурації застосунку.

Для візуального перегляду та управління даними у векторній базі даних можна використовувати веб-інтерфейс Attu, доступний за адресою `http://<хост_сервера>:8000/`.

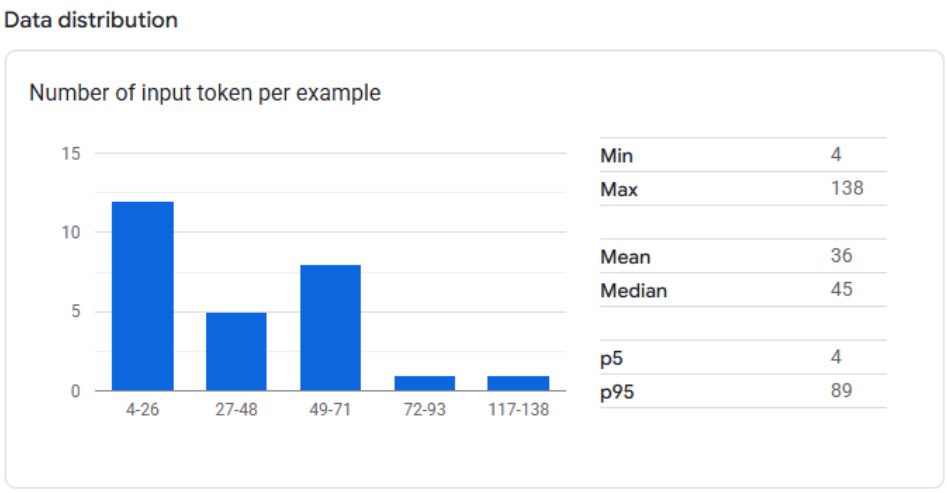
2.3.4. Дослідження ефективності та доналаштування моделі Gemini Flash 2.0 для програмної компоненти інтелектуального помічника

Підготовка даних та процес доналаштування.

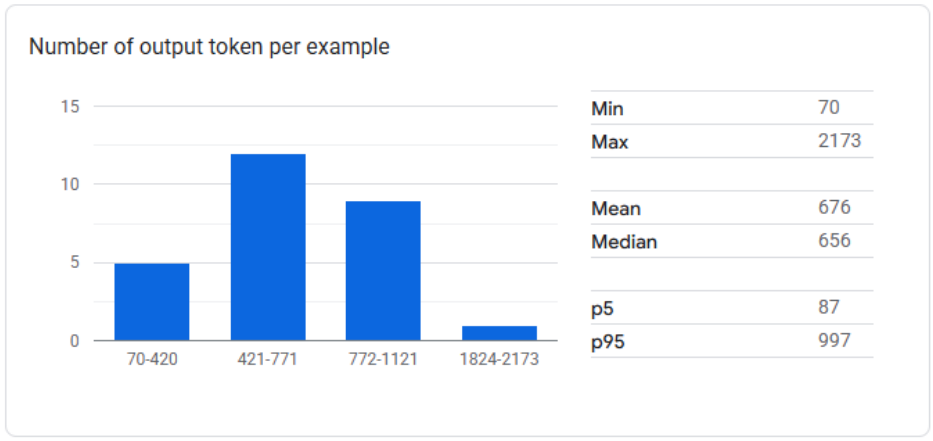
В рамках даної роботи було проведено експериментальне доналаштування (fine-tuning) великої мовної моделі Gemini Flash 2.0. Метою цього процесу є адаптація базової моделі для специфічного завдання – створення інтелектуального помічника, здатного надавати релевантну та структуровану інформацію щодо діяльності української платформи дистанційного навчання.

Для доналаштування було підготовлено спеціалізований набір даних (dataset), характеристики якого представлені на рис.2.6.

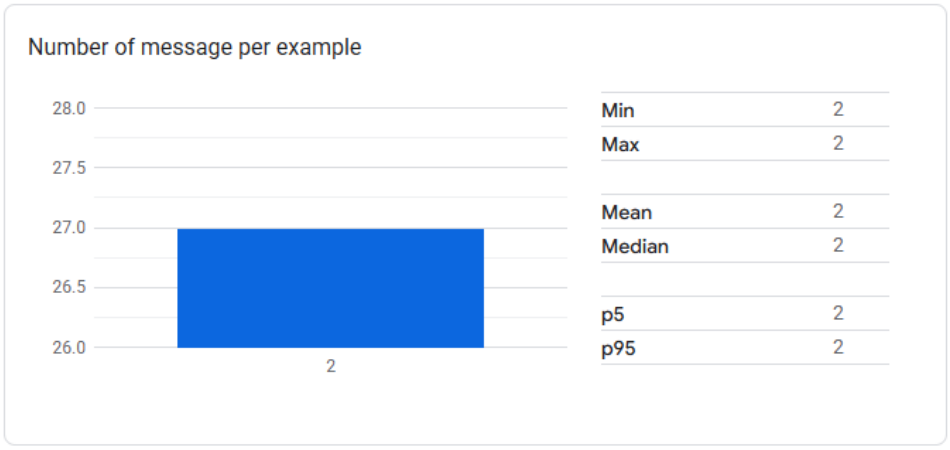
Набір складається з 27 прикладів (повідомлень), що включають загалом 19220 токенів. Приклади представляють собою діалогові пари «запит-відповідь», що відображають потенційні звернення користувачів до інтелектуального помічника. Тематика запитів охоплює загальну інформацію про платформу дистанційного навчання, навчальний процес, використання ШІ для допомоги учням, особливості дистанційного навчання тощо.



а)



б)



в)

Рис.2.6. Характеристика датасет для процесу донавчання

Аналіз розподілу даних показує наступні характеристики.

1. Кількість вхідних токенів на приклад (Number of input tokens per example – рис. 2.6,а). Мінімальне значення – 4, максимальне – 138, середнє – 36, медіана – 45. Гістограма свідчить про переважну концентрацію запитів в діапазонах 27-48 та 72-93 токенів.

2. Кількість вихідних токенів на приклад (Number of output tokens per example – рис. 2.6,б). Мінімальне значення – 70, максимальне – 2173, середнє – 676, медіана – 656. Розподіл демонструє значну варіативність довжини відповідей, що є характерним для надання детальної інформації, з піком у діапазоні 70-420 токенів.

3. Кількість повідомлень на приклад (Number of message per example – рис. 2.6,в). Стале значення 2 для всіх прикладів (мінімум, максимум, середнє, медіана), що підтверджує структуру даних у вигляді пар «запит-відповідь».

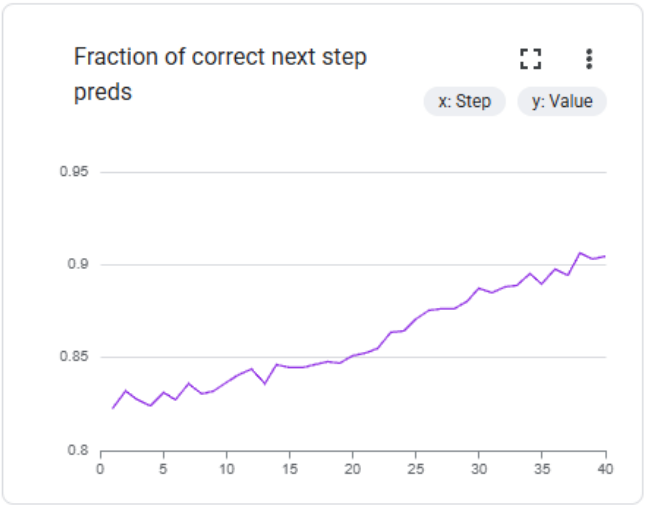
Результати доналаштування моделі

Процес доналаштування моделі Gemini Flash 2.0 відстежувався за допомогою ключових метрик, динаміка яких представлена на графіках рис.2.7.

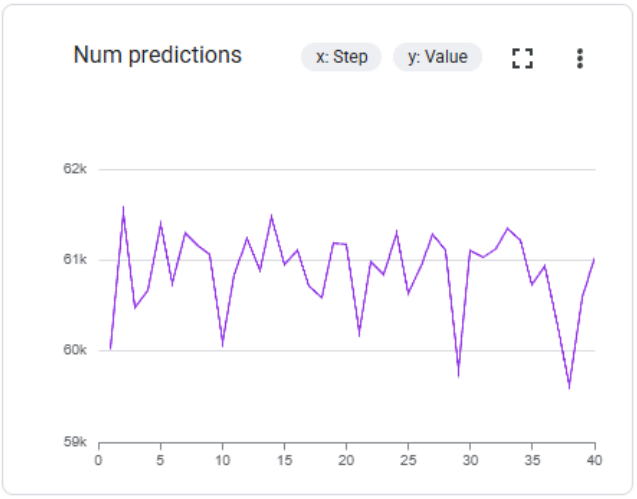
1. Загальні втрати (Total loss – рис. 2.7,в). Графік демонструє стійке зниження функції втрат протягом 40 кроків навчання. Початкове значення втрат становило близько 0.8, а наприкінці процесу воно зменшилося до рівня нижче 0.3. Така динаміка свідчить про успішне навчання моделі та її адаптацію до наданого набору даних, тобто модель ефективно засвоювала патерни для генерації цільових відповідей.

2. Частка коректних передбачень наступного кроку (Fraction of correct next step preds– рис. 2.7,а): Ця метрика показує точність моделі на рівні передбачення наступних елементів послідовності. Спостерігається чітка тенденція до зростання показника з приблизно 0.82 до понад 0.90. Це корелює зі зниженням втрат і підтверджує, що модель з кожним кроком навчання все краще генерує текст, який відповідає цільовим зразкам у навчальному датасеті.

3. Кількість передбачень (Num predictions– рис. 2.7,б): Графік показує кількість передбачень, що виконувались на кожному кроці навчання.



а)



б)

в)

Рис.2.7. Характеристика процесу донавчання

Значення коливаються в діапазоні приблизно від 59.5 тис. до 61.5 тис. Цей показник переважно відображає стабільність обчислювального процесу під час тренування (наприклад, розмір батчу та оброблюваних послідовностей) і не є прямим індикатором якості самої моделі порівняно з метриками втрат та точності.

Комбінаторний аналіз результатів досліджень, поданих на графіках рис.2.6, 2.7 дозволяє зробити наступні висновки.

Прийняті рішення, які обумовлюють створення та імплементацію в конкретику програмної архітектури та реалізацію методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання є працездатними та ефективними. Прийняті до використання в конкретній програмній реалізації сучасні технології великих мовних моделей (LLM) дозволяють реалізувати формування релевантних відповідей інтелектуального помічника.

Аналіз метрик процесу доналаштування моделі Gemini Flash 2.0 свідчить про успішну її адаптацію до специфіки завдання інтелектуального помічника до конкретної платформи дистанційного навчання. Зменшення функції втрат та зростання точності передбачень вказують на те, що модель навчилася генерувати відповіді, релевантні до контексту запитів та інформації про заклад освіти, закладеної у навчальному датасеті.

Ефективність формування релевантних відповідей досягнуто на рівні не гірше 0,9 при втратах не більше 0,3 одиниць. Зазначені показники відповідають прийнятним нормам розв’язку подібних завдань в інформаційній підтримці в ф форматі діалогу природною мовою.

ВИСНОВОК ДО РОЗДІЛУ 2

У ході досліджень розроблено метод побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей. Метод базується на послідовному визначенні контенту інформаційного повідомлення з використанням великих мовних моделей та відрізняється від аналогів формування інтерфейсного (короткого) запиту до різнорідної інформації, яка становить суть методичного забезпечення освітнього процесу платформи дистанційного навчання. Короткий запит формується із застосуванням LLM моделі. Результатом є вичерпана релевантна відповідь про інформаційні кластери методичного забезпечення, що підтримують суть відповіді на запит, сформований природною мовою.

З метою практичної імплементації розробленого методу запропоновано архітектуру програмно-апаратної платформи дистанційного навчання з інтелектуальною інформаційною підтримкою. Архітектура відображає класичні принципи розподілених комп'ютерних мереж з клієнт серверною взаємодією Апаратно-програмна компонента забезпечує фізичну підтримку розрахункових процесів LLM моделей та інтегрована до інформаційного середовища компанії, що надає освітні послуги у дистанційному форматі.

У ході досліджень розроблено програмну компоненту інтелектуального помічника для платформи дистанційного навчання. Розробка реалізовано мовою програмування високого рівня python із відповідними бібліотеками. Функціонал програмної компоненти – надання тим, хто навчається на платформі дистанційного навчання релевантної відповіді на питання, що охоплюють кластери: заняття; оцінювання; навчально-методичні матеріалів; організаційні питання; поточний стан тих, хто навчається. Ключові архітектурні рішення проекту: архітектура RAG (Retrieval-Augmented Generation) – дозволяє подолати обмеження стандартних LLM; інтеграція структурованої бази даних (MariaDB) та векторної бази знань (Milvus); використання спеціалізованої векторної бази даних (Milvus); залучення сучасної LLM (Gemini) з доналаштуванням;

контейнеризація за допомогою Docker та Docker Compose; структурований конвеєр підготовки даних. Ключові технологічнимі компоненти: векторна база даних Milvus; реляційна база даних MariaDB; LLM Gemini (Google); модуль пошуку та ранжування (Retriever & Reranker).

В рамках досліджень було проведено доналаштування (fine-tuning) великої мовної моделі Gemini Flash 2.0. Метою є адаптація базової моделі для специфічного завдання – створення інтелектуального помічника, здатного надавати релевантну та структуровану інформацію щодо діяльності української платформи дистанційного навчання. Для доналаштування було підготовлено спеціалізований набір даних (dataset) з 27 прикладів (повідомлень), що включають загалом 19220 токенів. Аналіз метрик процесу доналаштування моделі Gemini Flash 2.0 довів успішну її адаптацію до специфіки завдання інтелектуального помічника до конкретної платформи дистанційного навчання. Ефективність формування релевантних відповідей досягнуто на рівні не гірше 0,9 при втратах не більше 0,3 одиниць. Зазначені показники відповідають прийнятним нормам розв’язку подібних завдань в інформаційній

Однією з ключових особливостей запропонованої програмної компоненти інтелектуального помічника є його здатність до персоналізації та розуміння контексту. Зберігання історії розмови у сесіях користувачів дозволяє враховувати попередні запити та відповіді для підвищення достовірності. Інформації про користувача, наприклад, його вік, дозволяє адаптувати відповіді до вікової групи. Підсистема інтелектуальної підтримки розроблена з урахуванням вимог до надійності та масштабованості. Використання асинхронних операцій та багатопотоковості дозволяє обробляти велику кількість запитів одночасно, не знижуючи її продуктивності.

РОЗДІЛ 3
ПРИКЛАДНІ АСПЕКТИ ЗАСТОСУВАННЯ ПІДСИСТЕМИ ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ ДЛЯ ПРОГРАМНО-АПАРАТНОЇ ПЛАТФОРМ ДИСТАНЦІЙНОГО НАВЧАННЯ ЗА ТЕХНОЛОГІЯМИ NLP ТА ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

3.1. Тестування та оцінювання ефективності запропонованих рішень

Важливим етапом розробки будь-якої інформаційної системи, зокрема інтелектуального помічника, є оцінювання її ефективності та якості функціонування. Для об'єктивної оцінки розробленого в рамках даної роботи інтелектуального помічника було проведено етап тестування, що включав порівняльний аналіз його роботи з однією з провідних універсальних великих мовних моделей – ChatGPT від OpenAI.

Метою даного тестування є:

1. Верифікація здатності розробленого підсистеми інтелектуальної підтримки для програмно-апаратної платформи дистанційного навчання надавати точні, релевантні та повні відповіді на запити, що стосуються специфічної предметної області – діяльності платформи дистанційного навчання;

2. Порівняння ефективності спеціалізованого рішення, побудованого на архітектурі RAG (Retrieval-Augmented Generation) з використанням власної бази знань (Milvus), із можливостями потужної, але універсальної мовної моделі (ChatGPT), яка не має прямого доступу до специфічних даних платформи дистанційного навчання;

3. Виявлення сильних та слабких сторін розробленої підсистеми інтелектуальної підтримки порівняно з базовим рішенням у контексті розв’язку поставленого завдання.

Надалі детально описано методологію тестування, представлено приклади запитів та відповідей обох систем, а також проведено їх порівняльний аналіз за

ключовими критеріями якості: точність, повнота, релевантність та відсутність фактологічних помилок чи «галюцинацій».

Методика та критерії оцінювання.

Тестування проводилося шляхом постановки однакових запитів до обох систем: розробленого інтелектуального помічника та моделі ChatGPT (використовувалася версія GPT-4o). Запити були сформульовані таким чином, щоб перевірити знання системи щодо специфіки платформи дистанційного навчання, а також загальноосвітніх тем, що можуть виникати в контексті навчання.

Оцінювання відповідей здійснювалася за наступними показниками.

1. **Точність:** Відповідність наданої інформації фактам та реальним даним середовища дистанційного навчання або загальновідомим фактам.
2. **Повнота:** Обсяг наданої релевантної інформації у відповідь на запит.
3. **Релевантність:** Відповідність відповіді поставленому запитанню.
4. **Специфічність:** Здатність надавати інформацію, унікальну для платформи дистанційного навчання (правила, процедури, особливості навчальних матеріалів), а не лише загальні дані.
5. **Наявність інтеграції:** Здатність посилатися на конкретні навчальні матеріали чи ресурси платформи дистанційного навчання (очікується лише від розробленого інтелектуального помічника).
6. **Відсутність «галюцинацій»:** Генерація / відсутність генерації правдоподібної, але неправдивої або непідтвердженої інформації.

Результати порівняльного тестування.

Тестовий випадок 1: Авторство вірша «Конвалія»

Запит: Хто написав вірш «Конвалія»?

Результати роботи інтелектуального помічника подано в оригінальному форматі на рис.3.1. Відповідь сформовано за матеріалами, розміщеними на платформі дистанційного навчання.

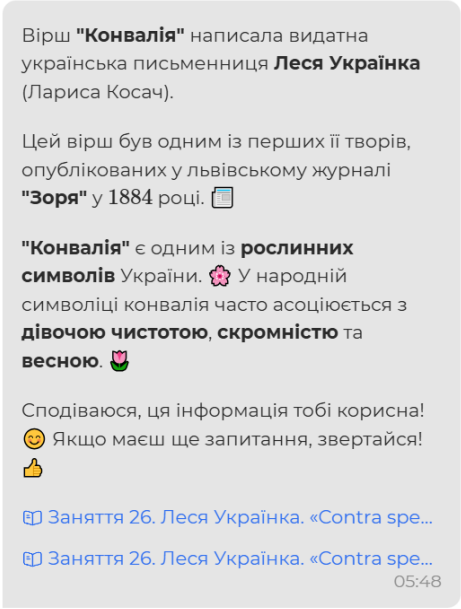


Рис.3.1. Відповідь розробленого інтелектуального помічника

Система коректно ідентифікувала автора як Лесю Українку (Ларису Косач). Додатково було надано контекстну інформацію: згадка про першу публікацію твору у львівському журналі «Зоря» у 1884 році, інформація про символічне значення конвалії в українській культурі. Важливою особливістю є наявність у відповіді прямих посилань на релевантні навчальні матеріали платформи дистанційного навчання – «Заняття 26. Леся Українка».

Нижче (див. рис.3.2) наведено відповідь на аналогічний запит від платформи генеративного штучного інтелекту ChatGPT.

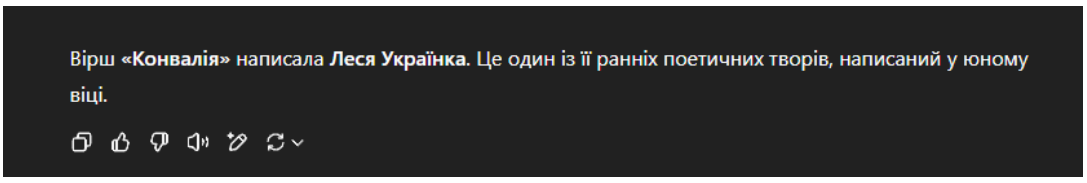


Рис.3.2. Відповідь ChatGPT

Система ChatGPT також правильно вказала автора вірша – Лесю Українку. Відповідь містить коротке доповнення, що це один із її ранніх поетичних творів, написаний у юному віці.

Аналіз результатів тестування.

Обидві системи надали правильну відповідь на пряме запитання про авторство. Однак відповідь розробленого інтелектуального помічника демонструє значні переваги в контексті освітнього застосування. Вона є значно повнішою, надаючи додаткові фактологічні (публікація) та культурологічні (символізм) відомості, що збагачує знання користувача. Ключовою перевагою є інтеграція з платформою дистанційного навчання через прямі посилання на заняття, що є неможливим для універсальної моделі ChatGPT. Відповідь ChatGPT є точною, але мінімалістичною та позбавленою специфічного для платформи дистанційного навчання контексту, а також додаткової корисної інформації, яку надав розроблений інтелектуальний помічник.

Стилістично, відповідь запропонованої компоненти інтелектуальної підтримки містить елементи дружнього спілкування (емодзі, заключні фрази), що може бути доречним для цільової аудиторії платформи дистанційного навчання.

Тестовий випадок 2: Інформація про батьків Ліни Костенко.

Запит: Хто були батьки Ліни Костенко?

Результати роботи розробленої підсистеми інтелектуальної підтримки наведені в оригіналі на рис.

Розроблена система повідомила, що батьки Ліни Костенко були вчителями, та надала розгорнуту інформацію про батька, Василя Костенка.

Було зазначено його високу освіченість, знання 12 мов (поліглот-самоук), педагогічний хист та факт сталінських репресій, через які він провів 10 років далеко від сім'ї.

Також згадано рік народження поетеси (1930) у вчительській сім'ї, яка прищеплювала високі моральні та естетичні цінності. Відповідь містить велику кількість релевантних посилань на навчальні матеріали платформи дистанційного навчання (Уроки 39, 46, 47 та Лекція 41).

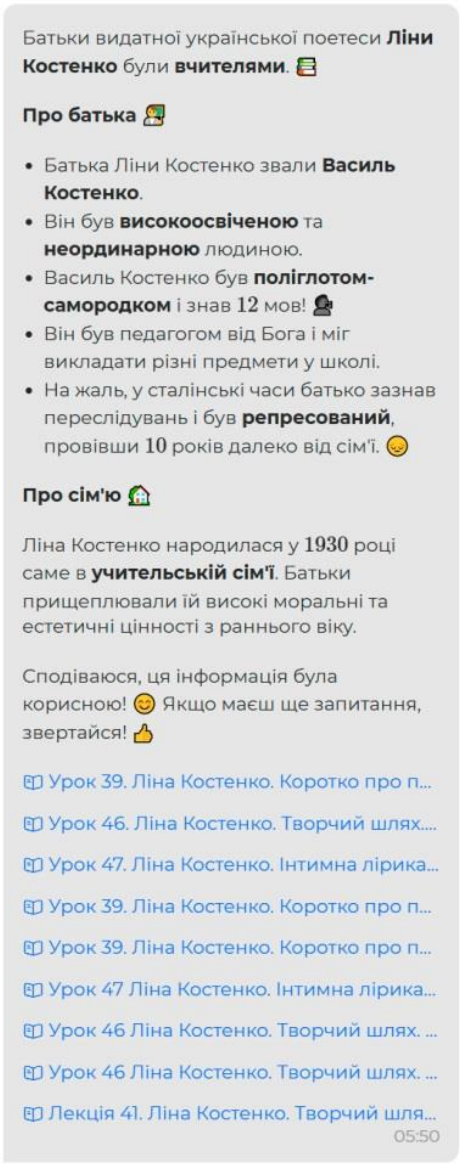


Рис.3.3. Відповідь розробленого інтелектуального помічника

На рис.3.4 надано відповідь ChatGPT.

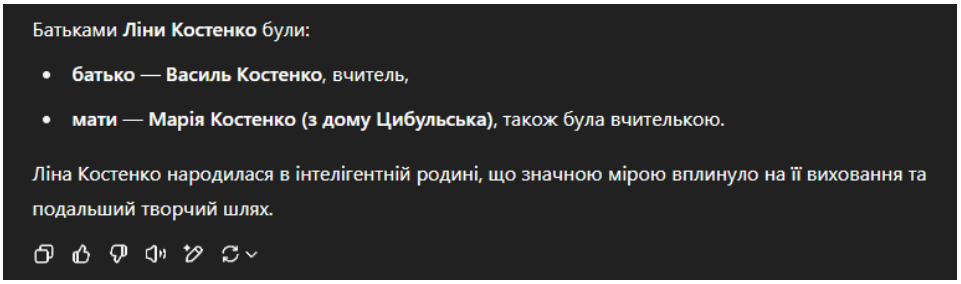


Рис.3.4. Відповідь ChatGPT

Генеративний інтелект ChatGPT надав більш структуровану відповідь щодо імен батьків: батько – Василь Костенко (вчитель), мати – Марія Костенко (з дому Цибульська). Додатково міститься узагальнююче речення про народження поетеси в інтелігентній родині та її вплив на виховання і творчий шлях.

Примітка: Надана ChatGPT інформація щодо імені та дівочого прізвища матері Ліни Костенко є невірною. (Матір поетеси звали Зінаїда Юхимівна Костенко).

Аналіз результатів тестування.

У другому тестовому випадку порівняння виявляє критичну перевагу розробленого інтелектуального помічника у точності наданої інформації. Хоча обидві системи правильно ідентифікували батька та його професію. ChatGPT припустився суттєвої фактологічної помилки, надавши невірне ім'я та дівоче прізвище матері поетеси. Це є прикладом так званої «галюцинації» або генерації неправдивої інформації, що є ризиком при використанні великих універсальних моделей для специфічних фактологічних запитів.

Розроблений інтелектуальний помічник хоч і не надав імені матері (що робить його відповідь менш повною у цьому конкретному аспекті), але надав достовірну та детальну інформацію про батька, яка є коректною. Це демонструє перевагу RAG-підходу, який спирається на базу знань: система надає ту інформацію, яка є в її базі (деталі про батька), і не намагається «вигадати» відсутні дані (ім'я матері), уникаючи таким чином помилки.

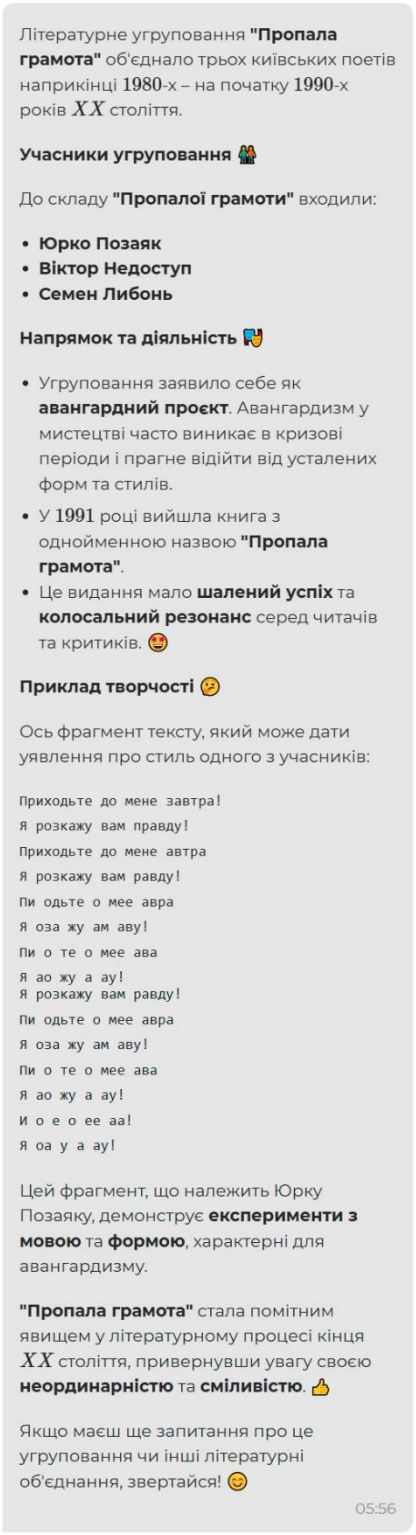
Окрім вищої точності, відповідь розробленого інтелектуального помічника залишається значно ціннішою в освітньому контексті завдяки глибоким біографічним деталям про батька та інтеграції з навчальними матеріалами через посилання.

Таким чином, для даного запиту розроблений інтелектуальний помічник виявився не тільки більш інформативним у певних аспектах та інтегрованим з платформою, але й суттєво більш надійним джерелом інформації через відсутність фактологічних помилок, які продемонстрував ChatGPT.

Тестовий випадок 3: Літературне угруповання «Пропала грамота».

Запит: Розкажи про літературне угрупування «Пропала грамота».

Відповідь розробленого інтелектуального помічника та ChatGPT наведені на рис.3.5,3.6 відповідно.



Літературне угруповання **"Пропала грамота"** об'єднало трьох київських поетів наприкінці 1980-х – на початку 1990-х років XX століття.

Учасники угруповання

До складу **"Пропалої грамоти"** входили:

- **Юрко Позаяк**
- **Віктор Недоступ**
- **Семен Либонь**

Напрямок та діяльність

- Угруповання заявило себе як **авангардний проєкт**. Авангардизм у мистецтві часто виникає в кризові періоди і прагне відійти від усталених форм та стилів.
- У 1991 році вийшла книга з однойменною назвою **"Пропала грамота"**.
- Це видання мало **шалений успіх** та **колосальний резонанс** серед читачів та критиків.

Приклад творчості

Ось фрагмент тексту, який може дати уявлення про стиль одного з учасників:

```
Приходьте до мене завтра!
Я розкажу вам правду!
Приходьте до мене автра
Я розкажу вам равду!
Пи одьте о мее авра
Я оза жу ам аву!
Пи о те о мее ава
Я ао жу а ау!
Я розкажу вам равду!
Пи одьте о мее авра
Я оза жу ам аву!
Пи о те о мее ава
Я ао жу а ау!
И о е о ее аа!
Я оа у а ау!
```

Цей фрагмент, що належить Юрку Позаяку, демонструє **експерименти з мовою** та **формою**, характерні для авангардизму.

"Пропала грамота" стала помітним явищем у літературному процесі кінця XX століття, привернувши увагу своєю **неординарністю** та **сміливістю**.

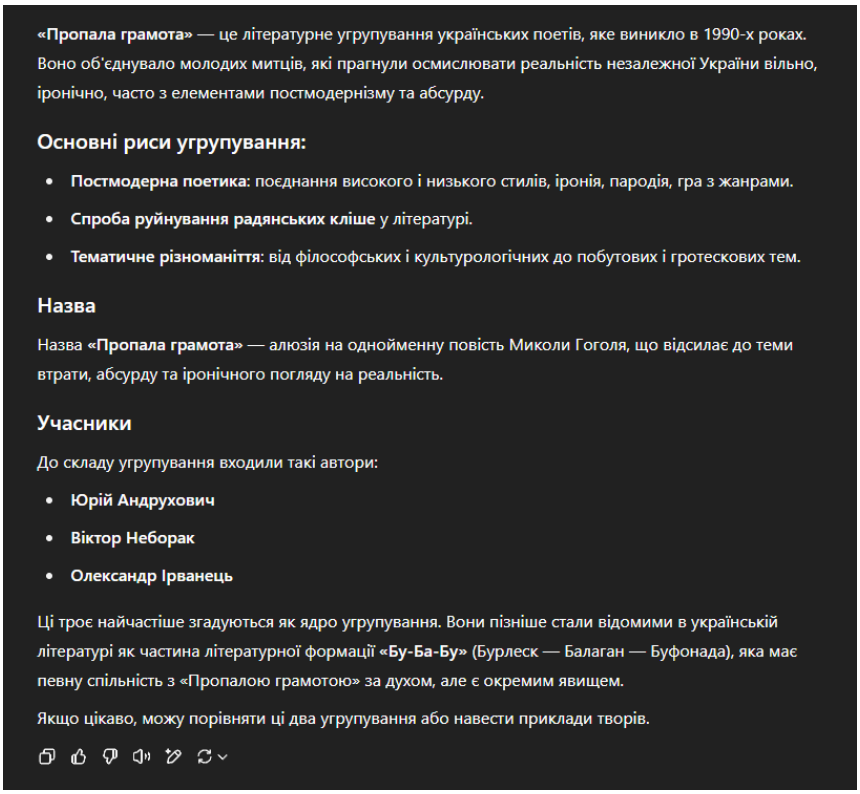
Якщо маєш ще запитання про це угруповання чи інші літературні об'єднання, звертайся!

05:56

Рис.3.5. Відповідь розробленого інтелектуального помічника

Розроблений інтелектуальний помічник надав коректну інформацію про період діяльності угруповання (кінець 1980-х – початок 1990-х рр.) та точно перелічив його основних учасників: Юрко Позаяк, Віктор Недоступ, Семен Либонь. Угруповання охарактеризовано як авангардний проєкт, що виник у кризовий період. Згадано вихід книги «Пропала грамота» у 1991 році та її значний резонанс. Цінним доповненням є приклад творчості – фрагмент тексту Юрка Позаяка, що ілюструє авангардні експерименти з мовою та формою. Відповідь також містить оцінку значення угруповання в літературному процесі того часу.

Результати тестування ChatGPT.



«Пропала грамота» — це літературне угруповання українських поетів, яке виникло в 1990-х роках. Воно об'єднувало молодих митців, які прагнули осмислювати реальність незалежної України вільно, іронічно, часто з елементами постмодернізму та абсурду.

Основні риси угруповання:

- **Постмодерна поетика:** поєднання високого і низького стилів, іронія, пародія, гра з жанрами.
- **Спроба руйнування радянських кліше** у літературі.
- **Тематичне різноманіття:** від філософських і культурологічних до побутових і гротескових тем.

Назва

Назва **«Пропала грамота»** — алюзія на однойменну повість Миколи Гоголя, що відсилає до теми втрати, абсурду та іронічного погляду на реальність.

Учасники

До складу угруповання входили такі автори:

- **Юрій Андрухович**
- **Віктор Неборак**
- **Олександр Ірванець**

Ці троє найчастіше згадуються як ядро угруповання. Вони пізніше стали відомими в українській літературі як частина літературної формації **«Бу-Ба-Бу»** (Бурлеск — Балаган — Буфонада), яка має певну спільність з «Пропалою грамотою» за духом, але є окремим явищем.

Якщо цікаво, можу порівняти ці два угруповання або навести приклади творів.

Рис.3.6. Відповідь ChatGPT

Система ChatGPT визначила часові рамки (1990-ті рр.) та спробувала описати стилістику через постмодернізм, іронію та руйнацію радянських кліше, а також пояснила назву як алюзію на твір М. Гоголя. Однак, система надала абсолютно невірну інформацію щодо складу угруповання, назвавши учасниками

Юрія Андруховича, Віктора Неборака та Олександра Ірванця. Подальша інформація у відповіді ChatGPT також є хибною, оскільки вона намагається пов'язати цих авторів (з «Бу-Ба-Бу») з «Пропалою грамотою» та самою «Бу-Ба-Бу», створюючи плутанину.

Аналіз результатів тестування.

Цей тестовий випадок є найбільш показовим щодо переваг спеціалізованої системи розробленого інтелектуального помічника над універсальною моделлю для запитів, що вимагають точних знань у вузькій предметній області (українська література кінця XX століття). Розроблена програмні компонента надала фактологічно бездоганну відповідь, коректно ідентифікувавши учасників, період, стилістику та надавши релевантний приклад творчості. Це свідчить про якість бази знань, на яку спирається система.

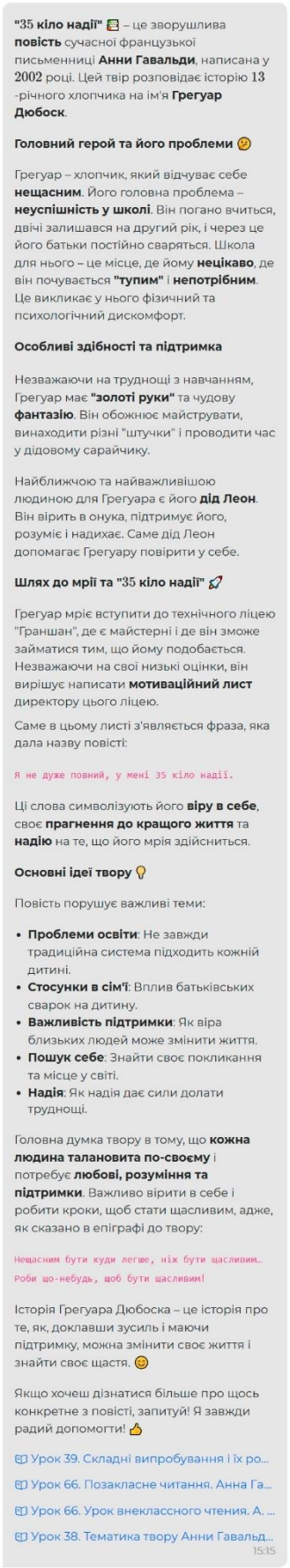
На противагу цьому, ChatGPT продемонстрував повну некомпетентність щодо даного запиту, припустившись грубих фактологічних помилок. Система не тільки назвала неправильних учасників, але й сплутала два різні літературні угруповання («Пропала грамота» та «Бу-Ба-Бу»). Така відповідь не просто марна, а шкідлива, оскільки вона дезінформує користувача. Це яскравий приклад «галюцинації» коли модель генерує хибну інформацію, яка може виглядати правдоподібно через впевнений тон викладу.

Висновок очевидний. Для отримання точної інформації зі специфічних галузей знань, особливо тих, що можуть бути недостатньо або суперечливо представлені в загальних тренувальних даних універсальних LLM, використання спеціалізованої системи з керованою базою знань (як розроблений інтелектуальний помічник з архітектурою RAG) є значно надійнішим та ефективнішим підходом.

Тестовий випадок 4: Опис твору «35 кіло надії».

Запит: 35 кіло надії.

Результати тестування наведені на рис.3.7, 3.8.



"35 кіло надії" – це зворушлива **повість** сучасної французької письменниці **Анни Гавальди**, написана у 2002 році. Цей твір розповідає історію 13 -річного хлопчика на ім'я **Грегуар Дюбоск**.

Головний герой та його проблеми

Грегуар – хлопчик, який відчуває себе **нещасним**. Його головна проблема – **неуспішність у школі**. Він погано вчиться, двічі залишався на другий рік, і через це його батьки постійно сваряться. Школа для нього – це місце, де йому **нецікаво**, де він почувається **"тупим"** і **непотрібним**. Це викликає у нього фізичний та психологічний дискомфорт.

Особливі здібності та підтримка

Незважаючи на труднощі з навчанням, Грегуар має **"золоті руки"** та чудову **фантазію**. Він обожнює майструвати, винаходити різні "штучки" і проводити час у дідовому сарайчику.

Найближчою та найважливішою людиною для Грегуара є його **дід Леон**. Він вірить в онука, підтримує його, розуміє і надихає. Саме дід Леон допомагає Грегуару повірити у себе.

Шлях до мрії та "35 кіло надії"

Грегуар мріє вступити до технічного ліцею "Граншан", де є майстерні і де він зможе займатися тим, що йому подобається. Незважаючи на свої низькі оцінки, він вирішує написати **мотиваційний лист** директору цього ліцею.

Саме в цьому листі з'являється фраза, яка дала назву повісті:

```
Я не дуже повний, у мені 35 кіло надії.
```

Ці слова символізують його **віру в себе**, своє **прагнення до кращого життя** та **надію** на те, що його мрія здійсниться.

Основні ідеї твору

Повість порушує важливі теми:

- **Проблеми освіти**: Не завжди традиційна система підходить кожній дитині.
- **Стосунки в сім'ї**: Вплив батьківських сварок на дитину.
- **Важливість підтримки**: Як віра близьких людей може змінити життя.
- **Пошук себе**: Знайти своє покликання та місце у світі.
- **Надія**: Як надія дає сили долати труднощі.

Головна думка твору в тому, що **кожна людина талановита по-своєму** і потребує **любові, розуміння та підтримки**. Важливо вірити в себе і робити кроки, щоб стати щасливим, адже, як сказано в епіграфі до твору:

```
Нещасним бути куди легше, ніж бути щасливим...
Роби що-небудь, щоб бути щасливим!
```

Історія Грегуара Дюбоска – це історія про те, як, доклавши зусиль і маючи підтримку, можна змінити своє життя і знайти своє щастя.

Якщо хочеш дізнатися більше про щось конкретне з повісті, запитуй! Я завжди радий допомогти!

Урок 39. Складні випробування і їх ро...

Урок 66. Позакласне читання. Анна Га...

Урок 66. Урок внеклассного чтения. А. ...

Урок 38. Тематика твору Анни Гавальд...

15:15

Рис.3.7. Відповідь розробленого інтелектуального помічника

Розроблена система ідентифікувала запит як назву твору, правильно вказала авторку (Анна Гавальда) та головного героя (Грегуар Дюбоск, 13 років). Відповідь містить детальний опис сюжету: труднощі Грегуара у школі, його взаємини з люблячим дідусем Леоном, талант до майстрування («золоті ручки»). Пояснюється символізм назви «35 кіло надії», пов'язаний із вагою Грегуара та його щирим листом-проханням до директора технічного ліцею, що символізує віру в себе та прагнення змінити життя. Наведено розгорнутий список основних тем твору (проблеми освіти, стосунки в сім'ї, важливість підтримки, пошук себе, надія), сформульовано головну думку та процитовано епіграф. Традиційно, відповідь містить посилання на релевантні навчальні матеріали платформи дистанційного навчання (№ 38, 39, 66).

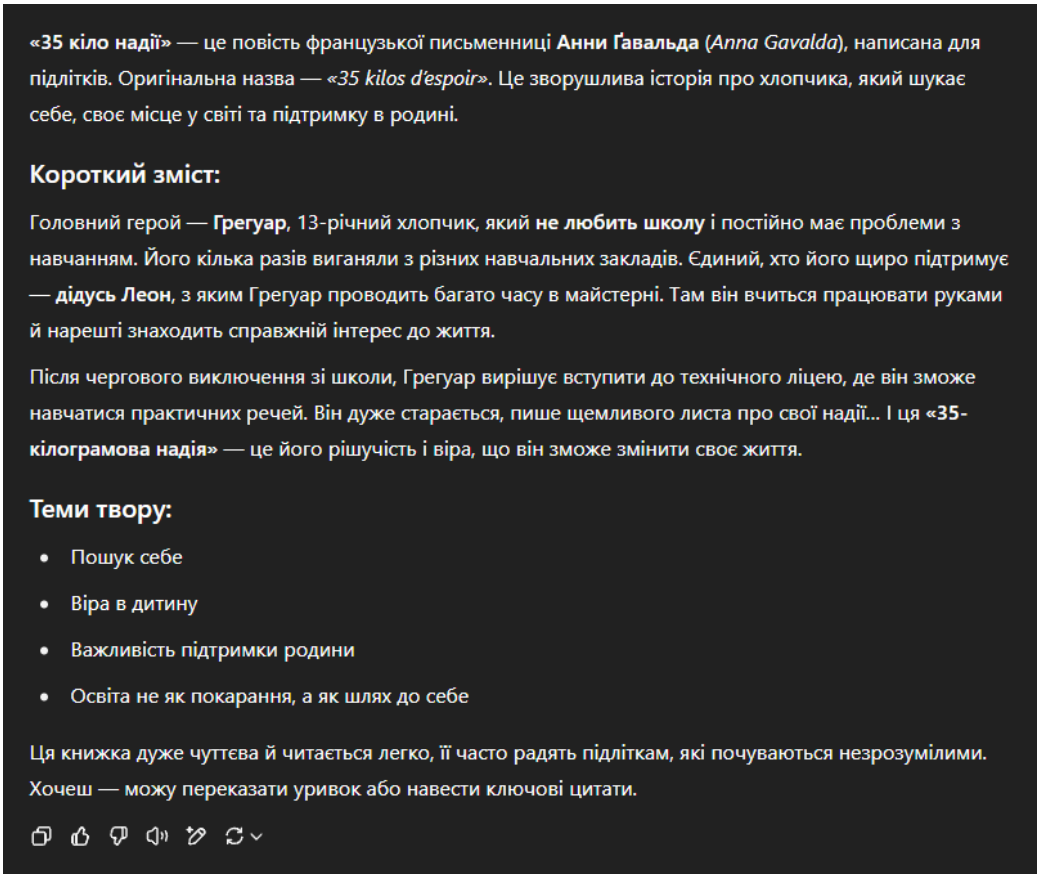


Рис.3.8. Відповідь ChatGPT

Система ChatGPT також коректно розпізнала запит як назву твору Анни Гавальди, вказавши оригінальну назву («35 kilos d'espoir») та цільову аудиторію

(підлітки). Надано короткий, але інформативний синопсис: головний герой Грегуар, його проблеми з навчанням та відрахування зі шкіл, підтримка дідуся Леона, захоплення роботою в майстерні, рішення вступити до технічного ліцею та написання мотиваційного листа. Значення назви пояснено як символ рішучості та віри героя. Перелічено ключові теми (пошук себе, віра в дитину, підтримка родини, освіта як шлях до себе). Відповідь завершується позитивною характеристикою книги та пропозицією надати уривки.

Аналіз результатів тестування.

На відміну від попередніх випадків зі специфічними запитами з української літератури, щодо цього відомого твору світової літератури для підлітків обидві системи надали високоякісні та фактологічно точні відповіді. Обидві системи інтелектуальної підтримки успішно впоралися із завданням описати сюжет, персонажів та теми повісті. Запропонований інтелектуальний помічник вирізняється трохи глибшим аналізом тем, формулюванням головної думки та наведенням епіграфа, що є цінним для навчального контексту. Інтеграція з навчальними матеріалами платформи дистанційного навчання залишається його унікальною перевагою.

ChatGPT надав дуже чіткий, добре структурований синопсис та вказав оригінальну назву твору. Його відповідь є лаконічною та повною водночас. Важливо відзначити, що для цього запиту ChatGPT не продемонстрував фактологічних помилок чи «галюцинацій», які спостерігалися раніше. Це може свідчити про те, що загальна база знань універсальних моделей є більш надійною для широко відомих тем світової культури, ніж для вузькоспеціалізованих національних тем.

Загалом, обидві відповіді є корисними, але розроблена компонента інтелектуальної підтримки із впровадженням запропонованих рішень пропонує додаткову освітню цінність через глибший аналіз та інтеграцію з навчальними матеріалами платформи дистанційного навчання.

Відповідно до описаної схеми досліджень було проведено статистичний аналіз головних показників, залежно від кількості слів у запиті. Узагальнені результати тестування наведені в табл.3.1 та на графіку рис.3.9.

Таблиця 3.1

Показники ефективності

функціонування розробленого інтелектуального показника

	Кількість слів в запиті					
Показники, %	3	4	5	10	15	20
Точність	0,4	0,7	1	1,5	2	4
Повнота	0,9	2	3	6	9	10
Релевантність	0,5	0,8	1,5	2,2	3	6
"Галюцинації"	0,2	0,5	0,8	1	1,2	3

Рис.3.9. Показники ефективності
функціонування розробленого інтелектуального показника

Наведені результати показують, що запропоновані рішення, орієнтовані на конкретну предметну область із підтримкою генерації відповідей релевантними джерелами платформи дистанційного навчання дозволяють підвищити ефективність функціонування інтелектуального помічника 10% порівняно з універсальними рішеннями генеративного штучного інтелекту типу ChatGPT.

Таким чином, наведені результати тестування та приклади застосування підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей довели дієвість, працездатність та ефективність запропонованих рішень.

ВИСНОВОК ДО РОЗДІЛУ 3

З метою демонстрації можливостей, придатності для практики застосування та для оцінювання ефективності запропонованих рішень було проведено тестування розробленої підсистеми інтелектуальної підтримки шляхом порівняльного аналізу її роботи з однією з провідних універсальних великих мовних моделей – ChatGPT від OpenAI.

Мета тестування: верифікація здатності розробленого підсистеми інтелектуальної підтримки надавати точні, релевантні та повні відповіді на запити; порівняння ефективності спеціалізованого рішення, побудованого на архітектурі RAG (Retrieval-Augmented Generation) з використанням власної бази знань (Milvus); виявлення сильних та слабких сторін розробленої підсистеми інтелектуальної підтримки.

Тестування проводилося шляхом постановки однакових запитів до обох систем: розробленого інтелектуального помічника та моделі ChatGPT.

Оцінювання відповідей здійснювалася за показниками: точність; повнота; релевантність; специфічність» наявність інтеграції до платформи дистанційного навчання; відсутність «галюцинацій».

Отримані результати показують, що запропоновані рішення, орієнтовані на конкретну предметну область із підтримкою генерації відповідей релевантними джерелами платформи дистанційного навчання дозволяють підвищити ефективність функціонування інтелектуального помічника до 10% за основними показниками, порівняно з універсальними рішеннями генеративного штучного інтелекту типу ChatGPT.

Таким чином, наведені результати досліджень та приклади застосування підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей довели дієвість, працездатність та ефективність запропонованих рішень.

РОЗДІЛ 4
СТАРТАП: «ПІДСИСТЕМА ІНТЕЛЕКТУАЛЬНОЇ ПІДТРИМКИ ДЛЯ ПРОГРАМНО-АПАРАТНОЇ ПЛАТФОРМ ДИСТАНЦІЙНОГО НАВЧАННЯ ЗА ТЕХНОЛОГІЯМИ NLP ТА ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ»

4.1. Загальна характеристика стартап-проєкту

Стартап-проєкт «Підсистема інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей» передбачає розробку та впровадження інтелектуального помічника для конкретного закладу освіти. Цей помічник використовує систему на основі ретривера з векторною базою даних Milvus для надання швидких, точних та релевантних відповідей на запити користувачів стосовно навчальних матеріалів та організаційних питань.

У табл. 4.1 наведено ключові аспекти ідеї проекту – її суть, потенційні сфери застосування в рамках навчального закладу та конкретні переваги для користувачів. Також було проведено порівняння з існуючими альтернативними методами отримання інформації (таблиця 4.2).

Таблиця 4.1

Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Інтелектуальний помічник на базі ретривер-системи (Milvus, MariaDB) та LLM (Gemini) освітнього закладу	Надання інформаційної підтримки студентам щодо навчальних матеріалів, розкладу, завдань.	Швидкий доступ до потрібної інформації 24/7, економія часу на пошук, отримання структурованих відповідей.
	Допомога викладачам у відповідях на типові запитання, надання швидких посилань на релевантні матеріали.	Зменшення рутинного навантаження на викладачів, стандартизація відповідей на поширені запитання.

Таблиця 4.1 (продовження)

Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	Інформування батьків щодо організаційних аспектів навчання, правил закладу освіти, важливих оголошень.	Зручний канал отримання офіційної інформації від закладу освіти, оперативне інформування.
	Автоматизація відповідей на запити до адміністрації закладу освіти щодо загальних питань (прийом, процедури, контакти).	Розвантаження адміністративного персоналу, швидке надання відповідей на стандартні запити.
	Потенційне розширення: інтеграція з іншими шкільними системами, аналітика запитів для покращення навчального процесу.	Персоналізовані рекомендації (в перспективі), покращення комунікації та доступності інформації в межах закладу освіти.

Таблиця 4.2

Визначення сильних, слабких та нейтральних характеристик ідеї проєкту

№	Техніко-економічні характеристики ідеї	(Потенційні) товари/концепції конкурентів (альтернативи)	W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
1	Вартість використання для кінцевого користувача (студент, вчитель)	Ручний пошук на сайт і/ платформі закладу освіти; Звернення до вчителя/адміністрації; Статичні FAQ-сторінки	-	Безкоштовне використання для цільової аудиторії закладу освіти	Миттєвий доступ 24/7 без очікування відповіді людини
2	Точність та релевантність інформації	Інформація на сайті (може бути застарілою / важко знайти); Відповідь людини (залежить від	Потенційна неточність LLM ("галюцинації"), залежність від якості	Актуальність залежить від оновлення даних в MariaDB/Milvus	Висока релевантність завдяки пошуку по матеріалах закладу

Таблиця 4.2 (продовження)

Визначення сильних, слабких та нейтральних характеристик ідеї проєкту

№	Техніко-економічні характеристики ідеї	(Потенційні) товари/концепції конкурентів (альтернативи)	W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		обізнаності); Загальні помічники (неспецифічні)	даних в БД		освіти (Milvus + MariaDB); Використання LLM для генерації
3	Швидкість отримання відповіді	Пошук на сайті (залежить від структури); Очікування відповіді людини (години/дні);FAQ (швидко, але обмежено)	-	Залежить від навантаження на систему та складності запиту	Майже миттєва відповідь на більшість запитів
4	Технологічна складність реалізації	Створення та підтримка сайту/FAQ; Людські ресурси для підтримки	Потреба у кваліфікованих розробниках та інфраструктурі (GPU для LLM / Milvus)	Використання існуючих технологій (Milvus, LLM API)	Сучасний технологічний стек, можливість масштабування
5	Специфічність для закладу освіти	Загальні освітні платформи; Загальні пошукові системи; Інші сайти закладів освіти(нерелевантні)	Вузька спрямованість на один заклад освіти (обмежує ринок поза закладом освіти)	Можливість адаптації для інших шкіл (потенційно)	Ідеальна відповідність потребам та контенту саме конкретного закладу освіти
6	Вартість розробки та підтримки	Вартість розробки сайту; З/п персоналу підтримки	Витрати на розробку; Витрати на хмарні ресурси (	Використання open-source	Потенційне зниження витрат на підтримку

4.2. Технологічний аудит ідеї проєкту

Для оцінки можливості реалізації проєкту «Підсистема інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей» проведено аналіз необхідних технологій. Результати наведено в табл. 4.3.

Таблиця 4.3

Технологічна здійсненність ідеї проєкту

№	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність
1	Бекенд-сервер чат-бота та API	Python 3.10+ (FastAPI/Flask)	Наявна	У відкритому доступі (Open Source)
2	Зберігання та пошук векторних представлень навчальних матеріалів	Milvus v2.5.7+	Наявна	У відкритому доступі (Open Source), вимагає Docker
3	Зберігання вихідних навчальних матеріалів та структурованих даних закладу освіти	MariaDB	Наявна	У відкритому доступі (Open Source)
4	Генерація відповідей на основі знайденого контексту	Google Gemini API (або інша LLM)	Наявна	Доступна через API (комерційна, є безкоштовні ліміти/рівні)
5	Векторизація тексту (перетворення тексту у вектори для Milvus)	Моделі з Hugging Face / Milvus Model	Наявні	У відкритому доступі (Open Source)
6	Система розгортання та управління контейнерами	Docker / Docker Compose	Наявна	У відкритому доступі (Open Source)
7	Інфраструктура для роботи	NVIDIA GPU / CUDA / Хмарні провайдери	Наявна	Комерційна (апаратне забезпечення або хмарні сервіси)
8	Інструмент для візуалізації даних у Milvus	Attu	Наявна	У відкритому доступі (Open

	(опціонально, для розробки/адміністрування)			Source), вимагає Docker

Аналіз інформації табл. 4.3 свідчить про високу технологічну здійсненність проєкту. Усі основні програмні компоненти (Python, Milvus, MariaDB, Docker, моделі векторизації) є доступними у відкритому доступі. Ключовий компонент генерації відповідей – LLM (Gemini) – доступний через API, що спрощує інтеграцію, хоча й передбачає операційні витрати. Потреба в специфічній інфраструктурі (GPU) залежить від обраного способу розгортання LLM та обсягів даних у Milvus, але існують хмарні рішення, що робить цю вимогу досяжною. Обраний технологічний стек є сучасним та активно підтримується спільнотами розробників.

4.3. Аналіз ринкових можливостей запуску стартап-проєкту

Оскільки проєкт «Підсистема інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей» спочатку орієнтований на внутрішнє використання в інтересах окремого закладу освіти, поняття "ринку" є специфічним. Проте, для оцінювання потенціалу (включаючи можливе масштабування на інші заклади) проаналізуємо ринкові аспекти.

Таблиця 4.4

Попередня характеристика потенційного ринку стартап-проєкту (ринок EdTech чат-ботів / систем підтримки)

№	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3-5 великих міжнародних гравців у сфері освітніх платформ з елементами AI; ~10+ менших спеціалізованих розробників чат-ботів для освіти.
2	Загальний обсяг продаж, грн/ум.од (оціночно,	Зростаючий, мільярди USD глобально (точна цифра для ніші чат-ботів важкодоступна, але

	глобальний ринок EdTech AI)	є частиною великого ринку). Локальний ринок значно менший, але з потенціалом.
3	Динаміка ринку (якісна оцінка)	Швидко зростає (особливо після пандемії та зростання інтересу до AI).

Таблиця 4.4 (продовження)

Попередня характеристика потенційного ринку стартап-проєкту

(ринок EdTech чат-ботів / систем підтримки)

№	Показники стану ринку (найменування)	Характеристика
4	Наявність обмежень для входу (вказати характер обмежень)	Потреба в AI/ML експертизі; Доступ до даних навчальних закладів; Вимоги до безпеки даних учнів (GDPR, локальні норми); Вартість розробки та інфраструктури.
5	Специфічні вимоги до стандартизації та сертифікації	Вимоги щодо захисту персональних даних (особливо дитячих); Можливі вимоги щодо інтеграції з існуючими освітніми стандартами чи платформами (LMS).
6	Середня норма рентабельності в галузі (або по ринку), % (SaaS EdTech)	20-35% (для успішних SaaS-продуктів у сфері EdTech).

Таблиця 4.5

Характеристика потенційних клієнтів стартап-проєкту

№	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Потреба в швидкому доступі до навчальної та організаційної інформації 24/7	Студенти закладу освіти	Молодші учні потребують простішого інтерфейсу; старші – більш складних запитів. Очікують миттєвих відповідей, переважно через мобільні пристрої.	До продукту: Простота використання, швидкість, точність відповідей, доступність на різних пристроях, інтерактивність. До компанії (закладу освіти): Надійність, актуальність інформації.

2	Зменшення рутинного навантаження, швидкий пошук матеріалів для	Викладачі закладу освіти	Потребують інструменту, що економить час. Можуть використовувати для швидкої	До продукту: Точність, повнота бази знань, легкість пошуку специфічної інформації, можливо, можливість додавати/коригувати

Таблиця 4.5 (продовження)

Характеристика потенційних клієнтів стартап-проєкту

№	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	студентів		перевірки інформації або надання посилань студентам. Цінують точність і надійність.	інформацію. До компанії (закладу освіти): Інтеграція в робочі процеси, підтримка.
3	Отримання актуальної інформації про навчання та правила закладу освіти	Батьки студентів	Зацікавлені в організаційних питаннях, безпеці, правилах, оголошеннях. Віддають перевагу чітким, офіційним відповідям.	До продукту: Достовірність інформації, простота доступу, зрозумілість відповідей, конфіденційність. До компанії: Прозорість, оперативність інформування.

Таблиця 4.6

Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Технологічні ризики	"Галюцинації" LLM, що призводять до невірних відповідей; Залежність від API сторонніх сервісів (Gemini); Складність підтримки Milvus/векторів.	Ретельне тестування; Впровадження механізмів перевірки фактів (RAG); Моніторинг якості відповідей; Розгляд альтернативних LLM; Навчання персоналу або аутсорс підтримки.

2	Безпека та конфіденційність	Ризик витоку персональних даних учнів або конфіденційної шкільної інформації; Недотримання норм GDPR або локального законодавства.	Шифрування даних; Контроль доступу; Анонімізація запитів; Аудит безпеки; Консультації з юристами; Вибір безпечних хмарних провайдерів.

Таблиця 4.6 (продовження)

Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
3	Сприйняття користувачами	Небажання студентів/вчителів використовувати новий інструмент; Недовіра до точності відповідей; Скарги на незручний інтерфейс.	Проведення навчальних сесій; Активна комунікація переваг; Збір зворотного зв'язку та ітеративне покращення інтерфейсу та функціоналу; Демонстрація успішних кейсів використання.
4	Залежність від даних	Якість відповідей прямо залежить від повноти та актуальності даних у MariaDB та якості їх векторизації у Milvus.	Розробка чітких процесів оновлення бази знань; Автоматизація процесу векторизації нових даних; Моніторинг цілісності даних.
5	Вартість експлуатації	Витрати на хмарні ресурси (особливо GPU для LLM/Milvus, якщо потрібно); Вартість API LLM (Gemini); Витрати на підтримку та оновлення.	Оптимізація використання ресурсів; Вибір оптимальних тарифних планів API; Розгляд можливості використання менш ресурсоємних моделей; Планування бюджету на підтримку.
6	Потенційна конкуренція	Поява нових спеціалізованих EdTech чат-ботів; Розвиток AI-функцій у великих освітніх платформах (LMS).	Вдосконалення продукту; Фокусування на унікальних перевагах (глибока інтеграція, точність); Гнучка цінова політика;

			Побудова сильного бренду

Таблиця 4.7

Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання попиту на AI в освіті	Підвищення інтересу закладів освіти до автоматизації процесів та персоналізації навчання за допомогою AI.	Позиціонування інтелектуального помічника як інноваційного рішення; Демонстрація переваг для закладу освіти; Розгляд можливості масштабування на інші заклади освіти.
2	Покращення технологій LLM/NLP	Поява нових, більш точних та ефективних мовних моделей; Розвиток технік RAG для підвищення якості відповідей.	Інтеграція нових моделей (наприклад, оновлення Gemini або альтернативи); Вдосконалення алгоритмів ретривера та реранкера; Покращення точності та природності відповідей.
3	Потреба у персоналізації навчання	Можливість аналізувати запити студентів для виявлення прогалин у знаннях або інтересів, надання персоналізованих рекомендацій.	Розробка модуля аналітики запитів (з дотриманням конфіденційності); Інтеграція з системами управління навчанням (LMS) для надання персоналізованих підказок або матеріалів (в майбутньому).
4	Розширення функціоналу	Додавання нових можливостей: перевірка завдань, інтерактивні квізи, планування розкладу, інтеграція з календарями тощо.	Поступове розширення функціоналу на основі зворотного зв'язку від користувачів та потреб заклада освіти; Розробка модульної архітектури для легкого додавання нових функцій.

5	Інтеграція з іншими системами	Можливість інтеграції з LMS, електронним щоденником, сайтом закладу освіти для створення єдиної інформаційної екосистеми.	Розробка API для взаємодії з іншими системами; Дослідження можливостей інтеграції з платформами, що використовуються в конкретному закладі освіти.

Таблиця 4.8

Ступеневий аналіз конкуренції на ринку (для потенційного зовнішнього ринку EdTech помічників)

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції	Олігополія з елементами монополістичної конкуренції: декілька великих гравців (LMS з AI) та багато дрібних нішевих розробників.	Фокусування на ніші (наприклад, закладу подібних за структурою до поточного); Диференціація за рахунок глибокої інтеграції з даними дистанційної платформи та якості RAG; Гнучкість у налаштуванні та ціноутворенні.
2	За рівнем конкурентної боротьби	Переважно національний/регіональний, але з впливом глобальних гравців та трендів AI.	Адаптація до специфіки місцевого освітнього ринку; Надання підтримки локальною мовою; Врахування місцевих норм щодо даних.
3	За галузевою ознакою	Внутрішньогалузева (EdTech) та міжгалузева (конкуренція з загальними AI-асистентами, які можуть намагатися відповідати на освітні запити).	Підкреслення освітньої специфіки та безпеки даних; Демонстрація переваг спеціалізованого рішення над загальними AI.

4	Конкуренція за видами товарів	Товарно-родова: конкуренція між різними підходами (чат-боти, інтегровані AI-помічники в LMS, бази знань, людська підтримка).	Постійне вдосконалення функціоналу чат-бота; Позиціонування як найбільш ефективного інструменту для швидкого та точного доступу до специфічної інформації закладу освіти.
5	За характером конкурентних переваг	Змішана: цінова (для стандартизованих рішень) та нецінова (якість AI, інтеграція, безпека,	Акцент на нецінових перевагах: точність RAG-системи, легкість інтеграції з даними закладу освіти,

Таблиця 4.8 (продовження)

Ступеневий аналіз конкуренції на ринку (для потенційного зовнішнього ринку EdTech помічників)

№	Особливості конкурентного середовища	В чому проявляється характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
		функціонал, підтримка).	безпека, зручність інтерфейсу, якість підтримки.
6	За інтенсивністю	Помірна до високої у певних сегментах	Постійний моніторинг конкурентів; Інновації та швидка реакція на зміни ринку; Побудова лояльності клієнтів через якісний продукт та сервіс.

Таблиця 4.9

Аналіз конкуренції в галузі за М. Портером (для зовнішнього ринку)

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкурен	Постачальники	Клієнти (заклади освіти)	Товари-замінники
Висновки:	Інші розробники EdTech чат-ботів; Великі LMS-платформи з AI-функціями.	Нові стартапи в сфері AI EdTech; Великі технологічні компанії (Google, Microsoft), що	Провайдери LLM API (Google, OpenAI); Провайдери хмарних сервісів (AWS, Azure, GCP); Постачальник	Заклади освіти (приватні, державні, онлайн). Вплив клієнтів: Середній до високого. Вимагають	Традиційні FAQ-сторінки; Бази знань; Форуми для учнів / батьків; Підтримка (викладачі

	Інтенсив-ність конкуренції: Середня, але зростаюча.	пропону-ють загальні AI-інстру-менти, адаптовані для освіти. Бар'єри входу: Середні (техноло-гії, дані, безпека).	и GPU (Nvidia). Вплив постачаль-ників: Високий (особливо LLM та хмари), залежність від їхніх цін та умов.	демонстрац ії ROI, безпеки, легкості інтеграції, відповід-ності освітнім потребам. Чутливі до ціни, але цінують якість та	, адміністра -ція) Загальні пошукові системи; Ручний пошук по сайту. Загроза замінників : Висока, але помічник.

Таблиця 4.9 (продовження)

Аналіз конкуренції в галузі за М. Портером (для зовнішнього ринку)

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкурен	Постачаль-ники	Клієнти (заклади освіти)	Товари-замінники
		Час виходу: 6-18 місяців.		підтримку. Важлива репутація постачаль-ника.	пропонує значно вищу швидкість, доступн-ість 24/7 та потенційно вищу точність

Таблиця 4.10

Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор значущим для порівняння)
1	Точність та релевантність відповідей (RAG система)	Використання Milvus для пошуку по конкретних векторизованих матеріалах закладу освіти перед генерацією відповіді LLM (Gemini) мінімізує «галюцинації» та забезпечує високу релевантність, що критично для освітнього середовища.
2	Глибока інтеграція з даними закладу освіти	Прямий доступ до актуальних даних з MariaDB (уроки, розклади, правила)

		дозволяє надавати найсвіжішу інформацію, на відміну від статичних FAQ або загальних ботів.
3	Специфічність для освітніх потреб конкретного закладу освіти	Інтелектуальний помічник розроблений з урахуванням конкретних запитів студентів, вчителів та батьків саме цієї платформи, що робить його значно кориснішим за універсальні рішення.
4	Використання сучасної LLM (Gemini)	Забезпечує генерацію природних, зрозумілих та граматично правильних відповідей, покращуючи досвід користувача порівняно з простішими скриптовими ботами.

Таблиця 4.10 (продовження)

Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор значущим для порівняння)
5	Швидкість та доступність 24/7	Миттєві відповіді в будь-який час значно перевершують очікування відповіді від людини або час, витрачений на ручний пошук інформації.
6	Потенціал для масштабування та розширення функціоналу	Архітектура (Python, Docker, Milvus) дозволяє відносно легко додавати нові джерела даних, інтегрувати нові функції або адаптувати рішення для інших шкіл (у перспективі).
7	Контроль над даними та безпека (при правильному розгортанні)	Можливість розгортання компонентів (окрім LLM API) у контрольованому середовищі (включаючи потенційно локальне) дає закладу освіти більше впевненості у безпеці даних порівняно з повністю хмарними сторонніми сервісами.

Таблиця 4.11

Порівняльний аналіз сильних та слабких сторін запропонованої системи інтелектуальної підтримки (відносно альтернатив/потенційних конкурентів)

№	Фактор конкурентоспроможності	Бали (1-5, де 5 -	Порівняння з альтернативами (Ручний пошук /	Порівняння з потенційними конкурентами

		дуже сильно)	FAQ / Людська підтримка)	(Інші EdTech боти / AI в LMS)
1	Точність та релевантність відповідей (RAG)	5	Значно вище	Потенційно вище завдяки специфічності даних
2	Глибока інтеграція з даними конкретного закладу освіти	5	Значно вище	Ключова перевага над загальними рішеннями

Таблиця 4.11

Порівняльний аналіз сильних та слабких сторін запропонованої системи інтелектуальної підтримки (відносно альтернатив/потенційних конкурентів)

№	Фактор конкурентоспроможності	Бали (1-5, де 5 - дуже сильно)	Порівняння з альтернативами (Ручний пошук / FAQ / Людська підтримка)	Порівняння з потенційними конкурентами (Інші EdTech боти / AI в LMS)
3	Специфічність для освітніх потреб конкретного закладу освіти	5	Значно вище	Ключова перевага над загальними рішеннями
4	Якість генерації відповідей (LLM Gemini)	4	Значно вище	На рівні або краще за багатьох, залежить від конкурента
5	Швидкість та доступність 24/7	5	Значно швидше і завжди доступно	Порівнянно з іншими ботами
6	Потенціал масштабування/розширення функціоналу	4	Значно вищий	Залежить від архітектури конкурента
7	Контроль над даними та безпека	4	Потенційно вищий контроль	Перевага над повністю хмарними сервісами

8	Вартість розробки/впровадження (початкова)	2	Значно вище	Порівнянно або вище за деякі готові рішення
9	Вартість експлуатації (LLM API, інфраструктура)	3	Вище за статичний сайт, але може бути нижче за людську підтримку	Порівнянно або вище, залежить від моделі
10	Простота використання (для кінцевого користувача)	4	Простіше за складний пошук	Очікується інтуїтивний інтерфейс

Таблиця 4.12

SWOT-аналіз стартап-проєкту

SWOT-аналіз стартап-проєкту. Сильні сторони:					
Висока точність та релевантність відповідей завдяки RAG (Milvus + MariaDB + Gemini).	Глибока інтеграція з конкретними даними закладу освіти.	Використан-ня сучасної LLM (Gemini) для якісної генерації тексту.	Доступ-ність 24/7 та висока швидкість відпові-дей.	Потенціал для аналізу запитів та покращен-ня освітньог о процесу.	Модульна архітекту-ра на базі сучасних технологі й (Python, Docker).

SWOT-аналіз стартап-проєкту. Слабкі сторони:						
Висок і початк ові витра ти на розро бку.	Постійні операційні витрати (LLM API, хмарна інфрастру ктура).	Залежніст ь від якості та актуально сті даних у MariaDB/ Milvus.	Потенційн і ризики неточ-ностей («галюцин ацій») LLM.	Необхідн ість у кваліфіко ваній технічній підтримц і та оновленн і системи.	Обмеже ність початко вого ринку (тільки конкрет ний заклад освіти).	Залежн-ість від зовнішніх постачаль ників (Google Cloud для Gemini).

SWOT-аналіз стартап-проєкту. Можливості:						
Покращен ня якості обслуговув	Зни-ження навантаж	Розши-рення функціонал	Поява більш доскона	Зроста ння ринку	OМожливі сть масштабув	Інтегра ція з іншими

ання та стейкхолдерів закладу освіти	ення на адміністратив-ний та викладацький склад.	у (персоналізація, інтерактивні завдання, аналітика)	лих та доступних LLM та векторних БД.	EdTech та попиту на AI-рішення в освіті.	ання рішення на інші навчальні заклади (SaaS-модель).	шкільними системами (LMS тощо).

Таблиця 4.12 (продовження)

SWOT-аналіз стартап-проєкту

SWOT-аналіз стартап-проєкту. Загрози:					
Ризики безпеки даних та конфіденційності (витоки, невідповід-ність нормам).	Негативне сприйняття або низький рівень використання інтелектуального помічника цільовою аудиторією.	Швидкий розвиток технологій AI, що може зробити поточне рішення застарілим.	Зростан-ня вартості LLM API або хмарних ресурсів.	Поява сильних конкурен-тів (якщо вихід на зовнішній ринок).	Зміни в освітньому законо-давстві або політиках закладу освіти.

Таблиця 4.13

Альтернативи ринкового впровадження стартап-проєкту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів (конкретного закладу освіти)	Строки реалізації (MVP)
1	Внутрішнє впровадження (пілотний проєкт): Розробка MVP (Minimum Viable Product) для конкретного закладу освіти. Фокус на основних функціях (відповіді на запити по навчальним матеріалам). Збір фідбеку, ітеративне покращення.	Висока (якщо є внутрішня підтримка та бюджет)	4-6 місяців

2	Розробка SaaS-платформи: Створення гнучкого рішення, адаптованого для різних шкіл. Пошук інвестицій (ангели, венчурні фонди). Активний маркетинг та продажі іншим закладам освіти.	Низька/Середня (потребує значних інвестицій та доведеної цінності)	12-18 місяців
3	Партнерство з LMS-провайдером: Інтеграція чат-бота як модуля до існуючої платформи управління навчанням. Спільне просування.	Низька (залежить від зацікавленості LMS-провайдера та складності інтеграції)	9-15 місяців
4	Open Source модель з комерційною підтримкою: Випуск базового функціоналу у відкритий доступ.	Середня (може залучити спільноту, але потребує ресурсів на підтримку та розвиток)	6-9 місяців

Обрана альтернатива: Після аналізу, найбільш доцільною та реалістичною для початкового етапу є Альтернатива 1: Внутрішнє впровадження (пілотний проєкт) конкретиний заклад освіти. Цей підхід дозволяє:

1. Перевірити гіпотези та технології в реальних умовах з мінімальними ризиками.
2. Отримати цінний зворотний зв'язок від кінцевих користувачів.
3. Створити робочий прототип та кейс успішного впровадження.
4. Зібрати дані для обґрунтування подальшого розвитку або масштабування (наприклад, переходу до Альтернативи 2 або 4).

4.4. Розроблення ринкової стратегії проєкту (в межах конкретного закладу освіти на етапі пілотування)

Таблиця 4.14

Вибір цільових груп потенційних споживачів (внутрішніх)

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті (з альтернативними	Простота входу у сегмент (донесення інформації, навчання)

				способами отримання інформації)	
1	Студенти закладу освіти – старші курси	Висока (звиклі до цифрових інструментів, цінують швидкість)	Високий (часті запити щодо матеріалів, розкладу)	Середня (активно користу-ються пошуком, але стикаються з трудно-щами)	Висока (через корпоративн ий канали комунікації)

Таблиця 4.14(продовження)

Вибір цільових груп потенційних споживачів (внутрішніх)

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті (з альтернативними способами отримання інформації)	Простота входу у сегмент (донесення інформації, навчання)
2	Викладачі закладу освіти	Середня/Висока (якщо продукт реально економить час та надійний)	Середній (для відповідей на типові питання, пошуку матеріалів)	Низька (прямих аналогів мало, але є звичка до існуючих методів)	Середня (потрібне навчання та демонстра-ція переваг)
3	Батьки студентів закладу освіти	Середня (залежить від технічної грамотності та потреби в інформації)	Середній (запити щодо організації, правил)	Низька	Середня (потрібна чітка комунікація про можливості)
4	Студенти закладу освіти – молодші курси	Низька / Середня (потребують дуже простого інтерфейсу,	Низький / Середній (простіші запити)	Низька	Низька (складніше впровадж-ення та навчання)

		можуть потребувати допомоги батьків / викладачів)			

Які цільові групи обрано для пілотного етапу:

1. Студенти старших курсів: Як основна група, найбільш готова до використання та з високим потенційним попитом.

2. Викладачі: Як важлива група для оцінки корисності інструменту та надання фідбеку, а також для промоції серед студентів. Стратегія охоплення: Концентрований маркетинг (внутрішній) – фокусування зусиль на обраних ключових групах в межах конкретного закладу освіти.

Таблиця 4.15

Визначення базової стратегії розвитку (для пілотного етапу)

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку (внутрішнього)	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Внутрішнє впровадження (пілотний проєкт) у конкретному закладі освіти	Концентрований маркетинг	Точність відповідей (RAG); Специфічність для конкретного закладу освіти; Швидкість 24/7; Інтеграція з актуальними даними.	Стратегія фокусування (на потребах конкретного закладу освіти) / Диференціації (за рахунок унікальних технологій RAG та інтеграції)

Обґрунтування вибору: Стратегія Фокусування / Диференціації є оптимальною для пілотного етапу. Проєкт концентрується на специфічних потребах вузького сегменту (конкретного закладу освіти) та пропонує унікальну цінність (точність, інтеграція, швидкість) завдяки застосуванню передових технологій (Milvus, RAG, Gemini), що відрізняє його від існуючих альтернатив (ручний пошук, FAQ). Це дозволить продемонструвати переваги та зібрати дані для подальших рішень.

Таблиця 4.16

Визначення базової стратегії конкурентної поведінки (відносно альтернатив всередені конкретного закладу освіти)

№	Чи є проект «першопрохідцем» на ринку (внутрішньому ринку конкретного закладу освіти як інтегрований AI-помічник)?	Чи буде компанія (проєкт) шукати нових споживачів, або забирати існуючих у конкурентів (альтернатив)?	Чи буде компанія (проєкт) копіювати основні характеристики товару конкурента (альтернатив), і які?	Стратегія конкурентної поведінки
1	Так, як інтегрований AI-асистент на базі RAG для специфічних даних платформи.	Буде «забирати 2 користувачів у менш ефективних альтернатив (ручний пошук, очікування відповіді людини).	Базова функція – надання інформації – спільна, але реалізація унікальна (AI, RAG, швидкість, 24/7).	Стратегія лідера ніші / Інноватора (в межах платформи)

Обґрунтування: У контексті конкретного закладу освіти проєкт виступає інноватором, пропонуючи новий, значно ефективніший спосіб доступу до інформації. Стратегія полягає у демонстрації переваг над існуючими (менш зручними) альтернативами та зайнятті позиції основного інструменту для інформаційної підтримки в обраних цільових групах.

Таблиця 4.17

Визначення стратегії позиціонування

(для внутрішніх користувачів конкретного закладу освіти)

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Швидкість отримання відповіді; Точність та релевантність інформації; Доступність 24/7; Простота використання.	Стратегія фокусування / диференціації	Висока точність (RAG); Миттєві відповіді; Доступ 24/7; Специфічність для конкретного закладу освіти; Використання передових AI технологій (Gemini, Milvus).	Швидкий Помічник закладу освіти; Точні Відповіді Миттєво; Твій Навігатор по платформі
2	Актуальність інформації; Надійність системи;	Стратегія фокусування / диференціації	Інтеграція з актуальною базою даних (MariaDB); Природна мова відповідей (Gemini);	Завжди Актуальна Інформація; Надійне Джерело

	Зрозумілість відповідей.		Надійна технологічна платформа (Docker).	Знань; Зрозуміло та Просто

4.5. Розроблення маркетингової програми стартап-проєкту (для внутрішнього пілотного запуску в конкретному закладі освіти)

Таблиця 4.18

Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед альтернативами (ручний пошук, FAQ, люди)
1	Отримати відповідь на питання щодо уроку/теми/розкладу швидко.	Миттєва відповідь у будь-який час доби.	Значно швидше за очікування відповіді вчителя/адміністратора; Швидше та зручніше за пошук по сайту чи документах.
2	Знайти точну та актуальну інформацію по конкретному матеріалу закладу освіти.	Надання релевантної інформації безпосередньо з бази знань конкретного закладу освіти.	Вища точність та релевантність, ніж у загальних пошуковиків або застарілих FAQ; Мінімізація ризику «галюцинацій» завдяки RAG.
3	Отримати інформацію у зручному, зрозумілому форматі.	Відповідь генерується LLM у вигляді природного діалогу.	Більш інтерактивно та зрозуміло, ніж статичні сторінки FAQ або сухі

			документи; Можливість уточнюючих запитань.
4	Зменшити час на пошук інформації та рутинні запити (для вчителів).	Автоматизація відповідей на типові запитання студентів.	Економія часу викладачів; Стандартизація відповідей; Швидке надання посилань на потрібні матеріали.
5	Мати єдину точку доступу до різної шкільної інформації.	Централізований помічник, що інтегрує знання з різних джерел (в перспективі).	Зручніше, ніж перемикатися між сайтом, платформою, чатами; Послідовний досвід отримання інформації.

Таблиця 4.19

Опис трьох рівнів моделі товару Інтелектуального помічника)

Рівні товару	Сутність та складові	М/Нм	Вр/Тх/Тл/Е/Ор
I. Товар за задумом	Вирішення проблеми неефективного доступу до навчальної та організаційної інформації в кокретному закладі освіти, надання швидкої, точної та цілодобової підтримки.		
II. Товар у реальному виконанні			
	Інтерфейс чат-бота (веб, можливо інтеграція в месенджери/платформу)	Нм	Е
	Функціонал: обробка запитів природною мовою, пошук релевантної інформації (RAG на базі Milvus/MariaDB), генерація відповідей (LLM Gemini).	М	Тх
	База знань: векторизовані навчальні матеріали та організаційна інформація конкретного закладу освіти.	М	Тх
	Технічні показники: швидкість відповіді (цільова <5 сек); точність пошуку (цільова >90% релевантності); доступність системи (цільова >99.5%).	М	Тх

	Показники надійності: стабільна робота під очікуваним навантаженням; механізми моніторингу та відновлення.	М	Тх
	Ергономічні показники: інтуїтивно зрозумілий інтерфейс; простота формулювання запитів; зрозумілість відповідей.	Нм	Е
	Показники безпеки: захист даних відповідно до політик закладу освіти та законодавства; контроль доступу до даних; анонімізація (де можливо).	М	Тх, Ор

Таблиця 4.19 (продовження)

Опис трьох рівнів моделі товару Інтелектуального помічника)

Рівні товару	Сутність та складові	М/Нм	Вр/Тх/Тл/Е/Ор
	Якість: відповідність стандартам коду (напр., PEP 8); наявність технічної документації; покриття тестами (цільове >90%).	М	Тх
	Пакування: Docker-контейнери для розгортання (Milvus, Attu, Chatbot-backend).	М	Тх
	Марка: «Інтелектуальний помічник» (або інша внутрішня назва, напр., «Інтелектуальний помічник студента»).	Нм	Ор
III. Товар із підкріпленням	**До запуску (внутрішнього):** Інформаційна кампанія в закладі освіти; навчальні матеріали/інструкції для користувачів; презентації для вчителів.		

За рахунок чого потенційний товар буде захищено від копіювання: На етапі внутрішнього пілотування основний захист – це специфічність інтеграції з унікальними даними та процесами конкретного закладу освіти. При потенційному виході на зовнішній ринок:

1. Ноу-хау: Унікальні налаштування RAG-системи, оптимізовані моделі векторизації для освітнього контенту.
2. Ліцензування ПЗ: Якщо буде комерційна версія.
3. Торгова марка: Реєстрація назви та логотипу.
4. Постійне вдосконалення: Швидший розвиток та додавання нових функцій, ніж у потенційних копій.
5. Угоди про нерозголошення (NDA): З клієнтами та партнерами.

Таблиця 4.20

Визначення меж встановлення ціни (для потенційної SaaS-моделі)

№	Рівень цін на товари-замінники (внутрішні витрати закладу освіти)	Рівень цін на товари-аналоги (інші EdTech чат-боти/AI-рішення для закладу освіти)	Рівень доходів/бюджетів цільової групи споживачів (інших платформ)	Нижня межа встановлення ціни на товар/послугу (SaaS, $/міс/в закладі)	Верхня межа встановлення ціни на товар/послугу (SaaS, $/міс/в закладі)
1	Вартість часу вчителів/адміністраторів на відповіді на запити (умовно 500-1500$/міс); Вартість підтримки застарілих FAQ/сайту (100-300$/міс).	200 - 2000 $/міс залежно від функціоналу, кількості користувачів, рівня підтримки. Базові боти дешевші, інтегрова	Бюджети на IT та інновації у приватних/онлайн платформах можуть варіюватися від кількох тисяч до десятків тисяч $ на рік.	Нижня межа: ~150-250$/міс (щоб бути конкурентним з базовими рішеннями та показати економію порівняно з	Верхня межа: ~800-1500$/міс (обґрунтовується унікальністю RAG, якістю інтеграції, економією часу, залежить від розміру платформи).

		ні AI-рішення дорожчі.		витратами на персонал).	

Таблиця 4.21

Формування системи збуту (внутрішньої комунікації та впровадження)

№	Специфіка поведінки цільових клієнтів (внутрішніх)	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення (інформаційного)	Концепція рекламного (інформаційного) звернення
1	Студенти: шукають швидкі відповіді, користуються месенджерами/шкільними платформами.	Онлайн-платформа закладу; Оголошення в класах / на зборах; Розсилки; Навчальні групи в соцмережах / месенджерах.	Швидкий Помічник закладу освіти; Точні Відповіді Миттєво; Твій Навігатор по закладу освіти.	Поінформувати про появу нового інструменту; Показати простоту використання; Пояснити переваги (швидкість, доступність)	«Запитай у інтелектуального помічника – отримай відповідь за секунди!», «Твій розумний помічник для навчання в вже тут!»
2	Викладач: цінують економію часу, потребують надійних інструментів, використовують	Педагогічні ради; Внутрішні розсилки для персоналу; Навчальні	Надійне Джерело Знань; Завжди Актуальна Інформа-	Пояснити, як чат-бот може допомогти в роботі; Продемонструвати	"Менше рутини, більше часу на головне! Заклад Chatbot

	робочі чати / email.	семінари/воркшопи; Робочі чати.	ція; Економія часу викладача.	точність та надійність; Запропонувати спробувати та надати фідбек.	відповідає на типові питання студентів.», "Точна інформація під рукою: використовуйте чат-бота для швидких консультацій."

Таблиця 4.21 (продовження)

Формування системи збуту (внутрішньої комунікації та впровадження)

№	Специфіка поведінки цільових клієнтів (внутрішніх)	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення (інформаційного)	Концепція реклам-ного (інформаційного) звернення
3	Батьки: шукають офіційну інформацію, користуються сайтом закладу, email, батьківськими чатами.	Сайт закладу (розділ новин/оголошень); Email-розсилки для батьків; Батьківські збори; Батьківські чати.	Надійне Джерело Знань; Завжди Актуальна Інформа-ція; Зрозуміло та Просто.	Повідомити про новий зручний канал отримання інформації; Пояснити, які питання можна ставити; Підкреслити офіційність та достовір-ність.	«Актуальна інформація про навчання– тепер у зручному поміч-нику!», «Правила закладу, розклад канікул, важливі оголошення – запитуйте у Інтелекту-ального поміч-ника!»

Таблиця 4.22

Концепція маркетингових комунікацій (для внутрішнього запуску)

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного (інформаційного) повідомлення
1	Студенти (старші): Технологічно обізнані, цінують швидкість.	Шкільна платформа, месенджери, оголошення онлайн.	Швидкий Помічник; Точні Відповіді; 24/7.	Показати як швидке та легке рішення проблеми пошуку інформації. Зацікавити спробувати.

Таблиця 4.22 (продовження)

Концепція маркетингових комунікацій (для внутрішнього запуску)

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного (інформаційного) повідомлення
2	Викладачі: Завантажені, потребують ефективних інструментів, цінують точність.	Внутрішні семінари, email, робочі чати.	Економія часу; Надійне джерело; Актуальна інформація.	Продемонструвати реальну користь для роботи, надійність, простоту використання. Зібрати фідбек.
3	Батьки: Потребують достовірної та легкодоступної інформації про заклад.	Сайт закладу, email-розсилки, батьківські збори.	Офіційна інформація; Легкий доступ; Надійність.	Повідомити про новий зручний канал зв'язку з закладом. Підкреслити достовірність інформації.

ВИСНОВОК ДО РОЗДІЛУ 4

Проведений аналіз розробки стартап-проєкту «Підсистема інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей» свідчить про його значний потенціал, насамперед як внутрішнього інструменту для конкретного закладу освіти. Існує чітко визначена потреба цільових аудиторій (студентів, викладачів, батьків) у швидкому та точному доступі до навчальної та організаційної інформації, яку існуючі методи (ручний пошук, FAQ, звернення до персоналу) задовольняють не повною мірою.

Технологічна здійсненність проєкту висока, оскільки обраний стек технологій є доступним, сучасним та дозволяє реалізувати ключовий функціонал – систему відповідей на основі ретривера (RAG), що забезпечує високу релевантність та точність.

Ключовими конкурентними перевагами проєкту є:

1. Висока точність відповідей завдяки RAG-архітектурі та інтеграції з конкретними даними конкретного закладу освіти.
2. Швидкість та доступність 24/7.
3. Використання передової LLM для генерації природних відповідей.
4. Потенціал для подальшого розвитку та інтеграції.

Обрана стратегія впровадження – пілотного проєкту конкретному закладі освіти – дозволяє мінімізувати ризики, перевірити технологію в реальних умовах та зібрати дані для подальших рішень щодо розвитку та можливого масштабування. Маркетингова програма для внутрішнього запуску фокусується на чіткій комунікації переваг для кожної цільової групи та використанні внутрішніх каналів закладу освіти.

Незважаючи на певні виклики (вартість розробки та експлуатації, залежність від якості даних, необхідність навчання користувачів), переваги від впровадження інтелектуального помічника (підвищення ефективності доступу до інформації, економія часу персоналу, покращення досвіду користувачів)

роблять подальшу імплементацію проєкту доцільною та перспективною в межах конкретного закладу освіти, з можливістю подальшої комерціалізації у разі успішного пілотування.

ВИСНОВКИ ДО РОБОТИ

В магістерській дисертації розв’язане актуальне науково прикладне завдання з розробки методу побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформи дистанційного навчання.

Аналіз показав, що існує доволі поширена практика інтелектуальних помічників в багатьох сферах діяльності соціуму, але їх імплементація в структури курсів дистанційного навчання ще не знайшла широкого застосування.

З огляду на необхідність реалізації задач побудови інтелектуального помічника для платформи дистанційого навчання слід орієнтуватись в розробці інтелектуального помічника на методи глибинного навчання та використанням великих мовних моделей. Це обумовлено тим, що запит на обробку має бути не формалізованим за структурою, але таким, що сприймається за контентом. Тому класичні методи машинного навчання та інші підходи не забезпечать таку гнучкість. В тренді задач і рішень слід передбачити програмну імплементації продуктивних моделей з потужною апаратною підтримкою їх функціонування.

У ході досліджень розроблено метод побудови підсистеми інтелектуальної підтримки для програмно-апаратної платформ дистанційного навчання за технологіями NLP та великих мовних моделей. Метод базується на послідовному визначенні контенту інформаційного повідомлення з використанням великих мовних моделей та відрізняється від аналогів формування інтерфейсного (короткого) запиту до різнорідної інформації, яка становить суть методичного забезпечення освітнього процесу платформи дистанційного навчання. Короткий запит формується із застосуванням LLM моделі. Результатом є вичерпана релевантна відповідь про інформаційні кластери методичного забезпечення, що підтримують суть відповіді на запит, сформований природною мовою.

З метою практичної імплементації розробленого методу запропоновано архітектуру програмно-апаратної платформи дистанційного навчання з інтелектуальною інформаційною підтримкою. Архітектура відображає класичні принципи розподілених комп’ютерних мереж з клієнт серверною взаємодією

Апаратно-програмна компонента забезпечує фізичну підтримку розрахункових процесів LLM моделей та інтегрована до інформаційного середовища компанії, що надає освітні послуги у дистанційному форматі.

Розроблено програмну компоненту інтелектуального помічника для платформи дистанційного навчання. Розробка реалізовано мовою програмування високого рівня python із відповідними бібліотеками. Ключові новації в архітектурних рішення проекту: архітектура RAG (Retrieval-Augmented Generation) – дозволяє подолати обмеження стандартних LLM; інтеграція структурованої бази даних (MariaDB) та векторної бази знань (Milvus); використання спеціалізованої векторної бази даних (Milvus); залучення сучасної LLM (Gemini) з доналаштуванням; контейнеризація за допомогою Docker та Docker Compose; структурований конвеєр підготовки даних. Ключові технологічнимі компоненти: векторна база даних Milvus; реляційна база даних MariaDB; LLM Gemini (Google); модуль пошуку та ранжування (Retriever & Reranker). У ході досліджень було проведено доналаштування (fine-tuning) великої мовної моделі Gemini Flash 2.0.

Однією з ключових особливостей запропонованої програмної компоненти інтелектуального помічника є його здатність до персоналізації та розуміння контексту. Зберігання історії розмови у сесіях користувачів дозволяє враховувати попередні запити та відповіді для підвищення достовірності. Інформації про користувача, наприклад, його вік, дозволяє адаптувати відповіді до вікової групи. Підсистема інтелектуальної підтримки розроблена з урахуванням вимог до надійності та масштабованості. Використання асинхронних операцій та багатопотоковості дозволяє обробляти велику кількість запитів одночасно, не знижуючи її продуктивності.

З метою демонстрації можливостей, придатності для практики застосування та для оцінювання ефективності запропонованих рішень було проведено тестування розробленої підсистеми інтелектуальної підтримки шляхом порівняльного аналізу її роботи з однією з провідних універсальних великих мовних моделей – ChatGPT від OpenAI. Тестування проводилося шляхом

постановки однакових запитів до обох систем: розробленого інтелектуального помічника та моделі ChatGPT. Оцінювання відповідей здійснювалася за показниками: точність; повнота; релевантність; специфічність» наявність інтеграції до платформи дистанційного навчання; відсутність «галюцинацій».

Отримані результати показують, що запропоновані рішення, орієнтовані на конкретну предметну область із підтримкою генерації відповідей релевантними джерелами платформи дистанційного навчання дозволяють підвищити ефективність функціонування інтелектуального помічника до 10% за основними показниками, порівняно з універсальними рішеннями генеративного штучного інтелекту типу ChatGPT.

Таким чином, мета досліджень досягнута, усі завдання виконані у повному обсязі.