

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка та дослідження чат-боту для сфери послуг, розробленого
на основі великої мовної моделі
(тема)

Виконав: здобувач групи СПРМ-22-1
Слінкін Олександр Вячеславович
(прізвище, ініціали)

Спеціальність 122 – Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне проєктування
(повна назва освітньої програми)

Керівник доц. каф. СТ Імангулова З. А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. Кафедри системотехніки _____
(підпис)

Гребеннік І. В.
(прізвище, ініціали)

2024 р.

Я як студент ХНУРЕ розумію і підтримую політику закладу із академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

10.06.2024



Слiнкiн

(дата, підпис, прізвище студента/-ки)

Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Попередній захист проведено 11 червня 2024 р.

Керівник кваліфікаційної роботи

доц. Імангулова З.А

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук
Кафедра _____ Системотехніки
Рівень вищої освіти _____ другий (магістерський)
Спеціальність _____ 122-Комп'ютерні науки
(код і повна назва)
Тип програми _____ освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма _____ Системне проектування
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« 12 » червня 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Слінкіну Олександрю Вячеслаавовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та дослідження чат-боту для сфери послуг, розробленого на основі великої мовної моделі

затверджена наказом по університету від 01 квітня 2024 р. № 259 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії 12.06.2024р.

3. Вихідні дані до роботи: Проаналізувати існуючі великі мовні моделі і методи реалізації чат-ботів для сфери послуг, проаналізувати фреймворки для побудови чат-ботів для сфери послуг на основі великих мовних моделей, та застосувати їх для оптимізації та покращення взаємодії з користувачами. Розробити чат-бот на основі отриманих даних. Протестувати роботу чат-боту з різними великими мовними моделями і обрати найбільш придатну для українського ринку сфери послуг. Перелік використаних програмних засобів: ОС Windows v.10, СУБД PostgreSQL. Середовище розробки: Microsoft Visual Studio 2022.

Технічне забезпечення: Intel(R) Core(TM) i5-6605 CPU @ 3.50GHz

4. Перелік питань, що потрібно опрацювати в роботі: 4.1 Вступ. 4.2 Опис предметної області. 4.2.1 Тенденції заміни служб підтримки чат-ботами. 4.2.2 Загальний огляд чат-ботів. 4.2.3 Основи розробки чат-ботів. 4.2.4 Постановка задачі. 4.3 Великі мовні моделі (LLM) 4.3.1 Основи технології трансформерів. 4.3.2 Огляд основних LLM 4.4 Деталізація ідеї чат-бота. 4.4.1 Функціональні вимоги до чат-бота. 4.4.2 Опис алгоритму чат-бота. 4.5 Технічна реалізація. 4.5.1 Вибір та налаштування фреймворку. 4.5.2 Реалізація збереження контексту та підказок. 4.6 Експериментальна частина. 4.6.1 Тестування чат-бота з різними LLM.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій: 5.1 Приклад високорівневої архітектури розмовного ШІ (CAA) (1 аркуш формату

РЕФЕРАТ

Кваліфікаційна робота: 102 стор., 3 рис., 0 табл., 3 додатки, 34 джерела.

ВЕЛИКІ МОВНІ МОДЕЛІ, МАШИННЕ НАВЧАННЯ, СФЕРА ПОСЛУГ, СИСТЕМА РЕКОМЕНДАЦІЙ, ТРАНСФОРМЕРИ, ЧАТ-БОТИ, .NET, SEMANTIC KERNEL.

Об'єкт дослідження – процес розробки чат-боту для сфери послуг, побудованого з використанням великої мовної моделі.

Предмет дослідження – методи та шляхи реалізації чат-боту, розробка, тестування взаємодії з різними LLM, а також інструменти та технології, необхідні для реалізації зазначених цілей.

Мета роботи – аналіз існуючих методів реалізації чат-ботів з використанням великих мовних моделей, оптимізація роботи з контекстом та покращення взаємодії боту з користувачами.

Методи розробки – у розробці чат-боту, що використовує велику мовну модель використовувався системний підхід, методи структурного аналізу і моделювання реляційних баз даних, методи аналізу розвитку чат-ботів та великих мовних моделей.

Результати роботи – розроблений чат-бот з використанням великої мовної моделі.

Область застосування – суб'єкти господарської діяльності у сфері послуг, що є зацікавленими у автоматизації взаємодії з користувачами.

ABSTRACT

Qualification work: 102 p., 3 pic., 34 source, 0 table, 34 source, 3 applications.

LARGE LANGUAGE MODEL, MACHINE LEARNING, SERVICE INDUSTRIES, RECOMMENDATION SYSTEM, TRANSFORMERS, CHAT BOTS, .NET, SEMANTIC KERNEL

Research object – The process of developing a chatbot for the service industry, built using a large language model..

Subject of research – Methods and ways of implementing a chatbot, development, testing interactions with various LLMs, as well as tools and technologies needed to achieve the stated goals..

Purpose of the work – Analysis of existing methods of implementing chatbots using large language models, optimization of context handling, and improvement of bot interactions with users..

Research methods – In the development of a chatbot using a large language model, a systematic approach was used, including methods of structural analysis and modeling of relational databases, as well as methods for analyzing the development of chatbots and large language models.

Work results – A chatbot developed using a large language model.

Scope – Business entities in the service sector that are interested in automating interactions with users.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень, термінів.....	9
Вступ.....	11
1 Опис предметної області.....	14
1.1 Тенденції заміни служб підтримки чат-ботами.....	14
1.1.1 Огляд ринку та вплив чат-ботів на сервісну індустрію.....	14
1.1.2 Приклади успішних імплементацій чат-ботів	16
1.2 Загальний огляд чат-ботів.....	20
1.2.1 Види чат-ботів і особливості їх застосування	20
1.2.2 Використання великих мовних моделей у чат-ботах	23
1.3 Основи розробки чат-ботів.....	26
1.3.1 Огляд сучасних фреймворків для розробки чат-ботів.....	26
1.3.2 Методи і техніки програмування чат-ботів.....	30
1.4 Постановка задачі дослідження.....	34
2 Великі мовні моделі (LLM)	37
2.1 Основи технології трансформерів.....	37
2.1.1 Як працюють трансформери.....	37
2.1.2 Особливості навчання і функціонування трансформерів.....	41
2.2 Огляд основних LLM	44
3 Деталізація ідеї чат-бота.....	50
3.1 Функціональні вимоги до чат-бота	50
3.1.1 Опис інформації, яка буде використовуватися ботом.....	50
3.1.2 Послуги, які надає чат-бот	54
3.2 Опис алгоритму роботи чат-бота.....	59
3.2.1 Сценарії взаємодії з користувачем.....	59
3.2.2 Обробка і реагування на запити користувачів	64
4 Технічна реалізація	71
4.1 Вибір та налаштування фреймворку	71
4.2 Реалізація збереження контексту та підказок	73
4.2.1 Методи зберігання діалогів у базах даних	73

4.2.2 Методи збереження контексту розмови.	78
5 Експериментальна частина.....	88
5.1 Тестування чат-бота з різними LLM	88
5.1.1 Тестування з GPT-3.5-turbo	89
5.1.2 Тестування з GPT-4	90
5.1.3 Тестування з Gemini	92
5.2 Методи виправлення помилок та оптимізації продуктивності.....	95
Висновки.....	105
Перелік джерел посилання	108

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

- AI - штучний інтелект;
- API - інтерфейс прикладного програмування;
- ASR - автоматичне розпізнавання мови;
- BERT - двонаправлені кодерні представлення з трансформерів;
- BLEU - двомовний стажер з оцінювання;
- CNN - згорткова нейронна мережа;
- CPU - центральний процесор;
- CUDA - обчислювальна уніфікована архітектура пристроїв;
- DL - глибоке навчання DNN - глибока нейронна мережа;
- GPU - графічний процесор;
- IT - інформаційні технології;
- LLM - велика мовна модель;
- ML - машинне навчання;
- NLP - обробка природної мови;
- NLU - розуміння природної мови;
- NMDP - нейронний машинний переклад;
- NLTK - інструментарій природної мови;
- RNN - рекурентна нейронна мережа;
- RNNLM - рекурентна нейромережева мовна модель;
- RoBERTa - надійно оптимізований підхід до попередньої підготовки BERT;
- SDK - комплект для розробки програмного забезпечення;
- TensorFlow - бібліотека програмного забезпечення з відкритим кодом для машинного навчання та штучного інтелекту;
- Токенізація - процес розбиття послідовності тексту на менші одиниці, які називаються токенами;
- TPU (Tensor Processing Unit Transformer) - модель глибокого навчання, яка використовує механізми уваги для обробки послідовних даних;

Трансферне навчання - техніка машинного навчання, яка передбачає використання попередньо навченої моделі як відправної точки для виконання нового завдання.

ВСТУП

Останніми роками розвиток і застосування технологій штучного інтелекту (ШІ) стрімко прискорилися в багатьох галузях. Однією зі сфер, де спостерігається значне зростання, є використання чат-ботів і віртуальних асистентів на основі ШІ для покращення обслуговування та підтримки клієнтів. Використовуючи досягнення в обробці природної мови (NLP), машинному навчанні (ML) і великих мовних моделях (LLM), організації тепер можуть розгорнути інтелектуальних діалогових агентів, які можуть розуміти і відповідати на запити клієнтів природною мовою.

Сфера послуг, зокрема, отримує значну вигоду від впровадження чат-ботів зі штучним інтелектом. В Україні на сферу послуг припадає значна частина економічної діяльності та зайнятості. За даними Державної служби статистики України, у 2021 році сектор послуг забезпечив понад 60% ВВП країни [1]. Сюди входять такі ключові галузі, як роздрібна торгівля, готельно-ресторанний бізнес, банківська справа, охорона здоров'я, телекомунікації тощо. Оскільки ці галузі стикаються зі зростаючою конкуренцією та очікуваннями клієнтів, потреба в ефективному, персоналізованому та завжди доступному обслуговуванні клієнтів стає першочерговою.

Традиційні моделі обслуговування клієнтів, які покладаються виключно на людський фактор, часто не встигають за обсягом і складністю взаємодії з клієнтами. Тривалий час очікування, непослідовні відповіді, обмежена доступність і високі витрати є загальними больовими точками. Саме тут чат-боти зі штучним інтелектом можуть суттєво вплинути на ситуацію. Автоматизуючи рутинні запити і завдання, чат-боти можуть забезпечити миттєву підтримку 24/7, скоротити час відповіді і звільнити людей, щоб зосередитися на більш складних питаннях. Дослідження показали, що добре розроблені чат-боти можуть обробляти до 80% поширених запитань і запитів клієнтів [2].

Однак розробка ефективних чат-ботів зі штучним інтелектом не є тривіальним завданням. Воно вимагає досвіду в NLP, ML, програмній інженерії та знанні предметної області. Одна з ключових проблем полягає в тому, щоб дати можливість чат-ботам брати участь у відкритих, контекстних розмовах, які імітують людське розуміння та генерування мови. Саме тут у гру вступають великі мовні моделі. LLM - це моделі глибокого навчання, які навчаються на величезних обсягах текстових даних, щоб вивчати тонкощі мови. Використовуючи знання, накопичені в цих моделях, чат-боти можуть генерувати більш зв'язні, релевантні та природні відповіді.

Серед найвідоміших LLM - GPT-3 від OpenAI, BERT від Google та RoBERTa від Facebook [4]. Ці моделі досягли найсучаснішої продуктивності в широкому спектрі завдань NLP , включаючи відповіді на запитання, узагальнення тексту, аналіз настроїв і генерацію діалогів. Однак розгортання ШІМ для реальних додатків, таких як чат-боти, все ще створює значні проблеми. До них належать вимоги до обчислювальних ресурсів, затримка, конфіденційність даних, упередженість та етичні міркування [4].

В Україні впровадження чат-ботів на основі LLM все ще перебуває на початковій стадії. Хоча в таких галузях, як банківська справа та електронна комерція, вже відбулося кілька помітних впроваджень, у секторі послуг існує значний невикористаний потенціал [5]. Однак організації, які прагнуть впровадити таких чат-ботів, повинні орієнтуватися в складному ландшафті технологій, постачальників і передових практик. Вони також повинні враховувати унікальний мовний, культурний та регуляторний контекст українського ринку.

Ця кваліфікаційна робота спрямована на вирішення проблем шляхом розробки фреймворку для створення чат-ботів на базі LLM, адаптованих для українського сектору послуг. Основними завданнями роботи є:

- ознайомитися з сучасним станом справ у галузі інформаційних систем та їх застосуванням у чат-ботах, з акцентом на сферу послуг;
- визначити ключові вимоги, міркування та найкращі практики для розробки чат-ботів на базі LLM для українських організацій;

- запропонувати архітектуру та методологію побудови, навчання та розгортання таких чат-ботів з використанням інструментів та платформ з відкритим вихідним кодом;
- продемонструвати здійсненність та ефективність запропонованого підходу через перевірку концепції у вибраній сфері послуг;
- надати рекомендації щодо майбутніх досліджень і розробок у цій сфері з урахуванням українського контексту.

Досягнувши цих цілей, ми сподіваємося зробити свій внесок у зростаючу кількість знань про чат-ботів зі штучним інтелектом та їхній потенціал для трансформації сектору послуг в Україні. Отримані результати та артефакти можуть слугувати цінним ресурсом для дослідників, практиків та організацій, які прагнуть використати можливості ШІ для покращення клієнтського досвіду та операційної ефективності.

Результати даної роботи доповідалися та обговорювалися на XXVIII Міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті» (Харків, 16-18 квітня 2024 р.) [6].

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Тенденції заміни служб підтримки чат-ботами

1.1.1 Огляд ринку та вплив чат-ботів на сервісну індустрію

Останніми роками сфера обслуговування клієнтів в Україні зазнає значних змін, зумовлених швидким впровадженням цифрових технологій та зміною споживчих очікувань. Однією з найпомітніших тенденцій стало дедалі ширше використання чат-ботів для заміни або доповнення традиційних послуг підтримки, що надаються людьми. У цьому розділі подано огляд українського ринку чат-ботів та досліджено вплив цієї технології на сферу послуг.

Згідно з нещодавнім звітом Української Ні-Tech Ініціативи, очікується, що ринок чат-ботів в Україні зростатиме зі середньорічним темпом зростання (CAGR) 30% у період з 2021 по 2026 рік, досягнувши до кінця прогнозованого періоду 120 мільйонів доларів США. Цьому зростанню сприяють кілька факторів, серед яких збільшення проникнення смартфонів та доступу до Інтернету, зростання вартості людської праці та бажання бізнесу надавати підтримку клієнтам у режимі 24/7.

Впровадження чат-ботів особливо помітне в таких галузях, як банківська справа, страхування, роздрібна торгівля та телекомунікації, де взаємодія з клієнтами є частою і часто повторюваною за своєю природою. Наприклад, ПриватБанк, один з найбільших банків України, впровадив чат-бота під назвою "Чат Приват24", який може обробляти широкий спектр запитів клієнтів, пов'язаних із залишками на рахунках, операціями, платежами тощо [25]. Бот, який доступний на популярних платформах обміну повідомленнями, таких як Telegram та Viber, обробив понад 10 мільйонів запитів користувачів з моменту свого запуску у 2018 році.

Аналогічно, провідна українська платформа електронної комерції Rozetka розгорнула чат-бота для допомоги клієнтам у пошуку товарів, відстеженні замовлень та повернень. Компанія повідомляє, що бот допоміг скоротити витрати

на підтримку клієнтів на 30%, одночасно покращивши час реагування та показники задоволеності клієнтів. Серед інших відомих українських компаній, які впровадили чат-ботів, - мобільний оператор "Київстар", страховий провайдер ARX та онлайн-тревел-агентство Tickets.ua [3].

Вплив чат-ботів на українську сферу послуг був багатограним. З одного боку, чат-боти дозволили компаніям надавати швидшу, більш послідовну та економічно ефективну підтримку клієнтів. Автоматизуючи рутинні запити та завдання, боти можуть обробляти велику кількість взаємодій з клієнтами одночасно, скорочуючи час очікування та звільняючи людей, які можуть зосередитися на більш складних питаннях. Це призвело до підвищення операційної ефективності та економії коштів для бізнесу.

З іншого боку, впровадження чат-ботів також викликало занепокоєння щодо витіснення робочих місць і потенційної втрати людського контакту у взаємодії з клієнтами. Опитування, проведене Українською асоціацією маркетингу, показало, що хоча 60% споживачів готові використовувати чат-ботів для простих запитів, 80% все ще віддають перевагу людській взаємодії для більш складних або делікатних питань [4]. Це підкреслює необхідність для бізнесу знаходити баланс між автоматизацією та людською підтримкою, а також розробляти чат-ботів, які можуть легко переходити до людських агентів, коли це необхідно.

Ще однією проблемою, з якою стикається український ринок чат-ботів, є брак лінгвістичних ресурсів та інструментів обробки природної мови (NLP), спеціально пристосованих для української мови. Хоча були певні зусилля з розробки україномовних платформ і фреймворків для чат-ботів, таких як BotMother і ChatBot.ua, екосистема все ще перебуває на стадії становлення порівняно з англійською та іншими основними мовами [5]. Це може обмежити можливості українського бізнесу створювати складних контекстно-орієнтованих чат-ботів, які можуть розуміти та реагувати на нюанси української мови.

Незважаючи на ці виклики, перспективи використання чат-ботів в українській сфері послуг залишаються позитивними. Оскільки технології NLP продовжують розвиватися і все більше підприємств визнають переваги

автоматизації, очікується, що впровадження чат-ботів прискориться. Однак успіх цих впроваджень залежатиме від здатності компаній створювати ботів, які будуть не лише функціональними, але й відповідатимуть потребам та вподобанням українських споживачів.

Отже, тенденція до заміни служб підтримки чат-ботами в Україні набуває все більших обертів, що зумовлено обіцянками підвищити ефективність, заощадити кошти та забезпечити доступність у режимі 24/7. Хоча ринок все ще розвивається, перші користувачі в таких галузях, як банківська справа, роздрібна торгівля та телекомунікації, вже відчули значні переваги. З розвитком технології та появою більшої кількості лінгвістичних ресурсів чат-боти стануть невід'ємною частиною українського сервісного ландшафту. Однак, щоб повністю реалізувати потенціал цієї трансформаційної технології, компаніям необхідно врахувати виклики, пов'язані з витісненням робочих місць, мовною складністю та споживчими уподобаннями.

1.1.2 Приклади успішних імплементацій чат-ботів

Останніми роками використання чат-ботів в українському секторі послуг неухильно зростає, що зумовлено потенційними перевагами покращення клієнтського досвіду, економії витрат та операційної ефективності. Хоча впровадження чат-ботів в Україні все ще перебуває на початковій стадії порівняно з деякими іншими країнами, вже є кілька помітних історій успіху в різних галузях. У цьому розділі ми розглянемо деякі приклади успішного впровадження чат-ботів українськими компаніями, проаналізувавши випадки використання, технології та досягнуті результати.

Банківські та фінансові послуги. Одним із перших і найвідоміших користувачів технології чат-ботів в Україні став банківський та фінансовий сектор. ПриватБанк, найбільший комерційний банк країни, у 2018 році запустив свого чат-бота Privat24 Chat, доступного на популярних платформах обміну повідомленнями, таких як Telegram, Viber та Facebook Messenger [24]. Чат-бот, що працює на основі

алгоритмів обробки природної мови та машинного навчання, може обробляти широкий спектр запитів і транзакцій клієнтів, включаючи перевірку балансу рахунку, грошові перекази, обробку платежів і заявки на кредити.

За даними ПриватБанку, з моменту запуску чат-бот «Приват24 Чат» обробив понад 50 мільйонів запитів користувачів, а середній час відповіді становив лише 2 секунди [25]. Банк повідомляє, що чат-бот допоміг зменшити навантаження на їхні колл-центри на 40%, що призвело до значної економії коштів та підвищення рівня задоволеності клієнтів. ПриватБанк продовжує розширювати можливості чат-бота, плануючи інтегрувати розпізнавання голосу та більш просунуті функції розмовного ШІ.

Ще одним яскравим прикладом у фінансовому секторі є чат-бот, розроблений українським фінтех-стартапом Raucell. Чат-бот Raucell, доступний у Telegram та Facebook Messenger, надає користувачам персоналізовані фінансові поради та рекомендації на основі їхніх доходів, витрат та фінансових цілей [2]. Чат-бот використовує обробку природної мови та машинне навчання, щоб розуміти запити користувачів і надавати індивідуальні пропозиції щодо бюджетування, заощаджень та інвестиційних стратегій.

За словами генерального директора Raucell, чат-бот зіграв важливу роль у залученні та утриманні клієнтів, особливо серед молодого покоління, яке віддає перевагу цифровим та розмовним інтерфейсам. Стартап повідомив про 30% збільшення залученості користувачів і 20% скорочення витрат на залучення клієнтів після впровадження чат-бота.

Електронна комерція та роздрібна торгівля. Галузь електронної комерції та роздрібною торгівлі в Україні також прийняла технологію чат-ботів як засіб для покращення підтримки клієнтів, пошуку продуктів та продажів. Одним із найяскравіших прикладів є чат-бот, розроблений Rozetka, найбільшим онлайн-рітейлером країни.

Чат-бот Rozetka, доступний у Telegram, Facebook Messenger та на веб-сайті компанії, може допомогти клієнтам у пошуку товарів, відстеженні замовлень та обробці повернень. Чат-бот використовує обробку природної мови та інформацію

про товари з каталогу Rozetka, щоб надавати релевантні та персоналізовані рекомендації на основі вподобань користувача та його попередніх покупок.

За даними Rozetka, чат-бот обробив понад 2 мільйони взаємодій з клієнтами з моменту його запуску в 2019 році, і, як повідомляється, рівень задоволеності клієнтів збільшився на 25% [3]. Компанія також вважає, що чат-бот знизив витрати на підтримку клієнтів завдяки автоматизації рутинних запитів і звільнив людей для вирішення більш складних питань.

Ще одним успішним впровадженням у сфері електронної комерції є чат-бот, розроблений українським ритейлером модного одягу LeBoutique. Чат-бот LeBoutique, доступний у Telegram та Facebook Messenger, виступає в ролі віртуального стиліста, допомагаючи клієнтам підібрати ідеальне вбрання відповідно до типу фігури, стильових уподобань та нагоди. Чат-бот використовує алгоритми машинного навчання для аналізу даних користувача та надання персоналізованих рекомендацій з каталогу товарів LeBoutique.

За даними компанії, чат-бот сприяв збільшенню середньої вартості замовлення на 15% та зменшенню кількості покинутих кошиків на 20% [8]. Генеральний директор LeBoutique пояснює успіх чат-бота його здатністю забезпечити більш цікавий та персоналізований досвід покупок, особливо для молодих клієнтів, які цінують зручність та цифрову взаємодію.

Охорона здоров'я та телемедицина. Пандемія COVID-19 прискорила впровадження чат-ботів в українському секторі охорони здоров'я, оскільки провайдери прагнуть пропонувати віддалені та безконтактні послуги, одночасно задовольняючи зростаючий попит на медичну інформацію та підтримку. Одним із яскравих прикладів є чат-бот, розроблений українською телемедичною платформою Helsi.

Чат-бот Helsi, доступний у Telegram та Viber, надає користувачам доступ до віртуальних консультацій з ліцензованими медичними працівниками, планування зустрічей та медичних інформаційних ресурсів [9]. Чат-бот використовує обробку природної мови для розуміння запитів користувачів і сортує запити за ступенем

терміновості та складності, перенаправляючи більш серйозні випадки до лікарів-людей для подальшої оцінки.

За даними Helsei, з моменту запуску на початку 2020 року чат-бот обробив понад 100 000 взаємодій з користувачами, при цьому середній час відповіді становив менше 5 хвилин [10]. Компанія повідомляє, що чат-бот допоміг полегшити навантаження на медичних працівників під час пандемії, надаючи своєчасну та достовірну медичну інформацію, а також полегшуючи дистанційні консультації та подальше лікування.

Іншим прикладом у сфері охорони здоров'я є чат-бот, розроблений українською фармацевтичною компанією «Дарниця». Чат-бот «Дарниця», доступний у месенджерах Telegram та Facebook, надає інформацію про продукцію компанії, інструкції з дозування та потенційні побічні ефекти [8]. Чат-бот також пропонує персоналізовані нагадування про прийом ліків та відстеження прихильності до лікування, щоб допомогти пацієнтам більш ефективно керувати своїми схемами лікування.

За даними Дарниця, чат-бот сприяв підвищенню рівня прихильності до ліків серед користувачів на 10%, а також зменшенню кількості запитів до служби підтримки, пов'язаних з інформацією про продукт та його використання [12]. У майбутньому компанія планує розширити можливості чат-бота, включивши в нього перевірку симптомів і віртуальні консультації з фармацевтами.

Хоча ці приклади демонструють значні успіхи, важливо визнати, що не всі впровадження чат-ботів в Україні були повністю успішними. Були випадки, коли чат-боти не відповідали очікуванням користувачів, надавали неточну або нерелевантну інформацію або стикалися з проблемами при обробці складних або нюансованих запитів [13]. Однак цей досвід допоміг сформувати найкращі практики та винести уроки для подальшого розвитку чат-ботів у країні.

Оскільки український сектор послуг продовжує впроваджувати технологію чат-ботів, організаціям вкрай важливо ретельно враховувати конкретні потреби та вподобання своєї цільової аудиторії, інвестувати у високоякісні дані та мовні ресурси, а також надавати пріоритет користувацькому досвіду та етичним

міркуванням. Навчаючись як на успіхах, так і на невдачах, українські компанії можуть продовжувати вдосконалювати та покращувати свої впровадження чат-ботів, розкриваючи весь потенціал цієї трансформаційної технології.

1.2 Загальний огляд чат-ботів

1.2.1 Види чат-ботів і особливості їх застосування

Типи чат-ботів та особливості їх застосування Чат-боти, також відомі як розмовні агенти або віртуальні помічники, - це комп'ютерні програми, призначені для імітації людської розмови за допомогою текстової або голосової взаємодії. Вони використовують методи обробки природної мови (NLP) і машинного навчання (ML), щоб розуміти введення користувача, визначати наміри і генерувати відповідні відповіді. Чат-боти можна класифікувати на різні типи залежно від технології, що лежить в їхній основі, призначення та рівня складності.

Згідно з однією з поширених класифікацій, чат-боти поділяються на системи, що базуються на правилах, і системи на основі штучного інтелекту [9]. Чат-боти на основі правил, також відомі як боти з деревом рішень, відповідають на запити користувачів за заздалегідь визначеним набором правил і сценаріїв. Зазвичай вони обробляють прості, структуровані діалоги і покладаються на зіставлення шаблонів і розпізнавання ключових слів для визначення відповідної відповіді. Хоча ботів на основі правил відносно легко впровадити і вони можуть бути ефективними для конкретних випадків використання, їм бракує гнучкості та контекстного розуміння, притаманних більш просунутим системам.

З іншого боку, чат-боти зі штучним інтелектом використовують алгоритми машинного навчання та обробку природної мови, щоб забезпечити більш природні, відкриті розмови. Ці боти можуть вчитися на попередніх взаємодіях, адаптуватися до вподобань користувачів і генерувати відповіді динамічно, виходячи з контексту розмови. Існує кілька типів чат-ботів зі штучним інтелектом, зокрема моделі на основі пошуку та генеративні моделі.

Чат-боти, засновані на пошуку, використовують сховище заздалегідь визначених відповідей і вибирають найбільш підходящу з них на основі введення користувачем. Вони часто використовують такі методи, як зіставлення ключових слів, семантична схожість та пошук інформації, щоб знайти найкращий варіант відповіді. Хоча пошукові боти можуть обробляти широкий спектр запитів, їхні відповіді обмежені заздалегідь визначеним набором і не завжди можуть бути контекстуально релевантними.

Генеративні чат-боти, що працюють на основі моделей глибокого навчання, таких як рекурентні нейронні мережі (RNN) та трансформери, можуть генерувати нові відповіді слово за словом на основі контексту розмови. Вони навчаються на великих масивах даних людських розмов і можуть навчитися імітувати мовні патерни та стилі, подібні до людських. Генеративні боти пропонують більшу гнучкість і можуть обробляти відкриті розмови, але вони вимагають значних обчислювальних ресурсів і іноді можуть генерувати нерелевантні або непослідовні відповіді.

Інший вимір класифікації чат-ботів базується на їхньому призначенні або сфері застосування. Деякі з найпоширеніших типів включають:

- чат-боти для обслуговування клієнтів: ці боти призначені для обробки запитів клієнтів, надання інформації та вирішення проблем у певній сфері бізнесу, наприклад, банківській, роздрібній торгівлі або телекомунікаціях [3];
- віртуальні персональні асистенти: це чат-боти загального призначення, які можуть виконувати широкий спектр завдань, наприклад, встановлювати нагадування, давати рекомендації та відповідати на запитання на різні теми [26];
- медичні чат-боти: ці боти допомагають пацієнтам у перевірці симптомів, наданні медичної інформації, плануванні зустрічей та дистанційному моніторингу;
- освітні чат-боти: ці боти надають підтримку в навчанні, відповідають на запитання студентів і полегшують викладання курсу в освітніх установах [5];

– чат-боти для HR та рекрутингу: ці боти автоматизують різні HR-процеси, такі як перевірка кандидатів, планування співбесід та адаптація співробітників.

Вибір типу та технології чат-бота залежить від конкретного випадку використання, бажаного рівня взаємодії та наявних ресурсів. Боти на основі правил підходять для простих, лінійних розмов з вузькою тематикою, тоді як боти на основі штучного інтелекту краще підходять для більш складних, відкритих взаємодій. Моделі на основі пошуку пропонують більше контролю над відповідями, але їм може бракувати контекстуальної релевантності, тоді як генеративні моделі забезпечують більш природні розмови, але вимагають ретельного навчання та моніторингу.

В українському контексті використання різних типів чат-ботів варіюється залежно від галузі. У банківському секторі для таких завдань, як запити щодо рахунків, транзакції та підтримка клієнтів, зазвичай використовують моделі, засновані на правилах та пошуку [9]. В електронній комерції досліджуються чат-боти зі штучним інтелектом і генеративними можливостями для персоналізованих рекомендацій продуктів і залучення клієнтів [10]. Медичні працівники експериментують з чат-ботами на основі правил та ШІ для сортування, телемедицини та підтримки пацієнтів [11].

Однак розробка та впровадження чат-ботів в Україні стикається з низкою проблем. Відсутність великих масивів даних української мови та попередньо навчених моделей ускладнює створення складних чат-ботів зі штучним інтелектом, здатних розуміти та генерувати природні українські розмови [12]. Складність української мови з її багатою морфологією та синтаксисом вимагає значних зусиль для збору даних, анотування та адаптації моделей.

Крім того, культурні та соціальні норми українських користувачів можуть відрізнятися від норм інших країн, що вимагає розробки чат-ботів з урахуванням місцевих уподобань та стилів спілкування [13]. Довіра користувачів та їхнє сприйняття чат-ботів також є важливими факторами, особливо в таких чутливих сферах, як охорона здоров'я та фінанси.

Незважаючи на ці виклики, потенційні переваги чат-ботів у покращенні клієнтського досвіду, зниженні витрат і підвищенні операційної ефективності сприяють їхньому впровадженню в українському секторі послуг. Оскільки технології NLP продовжують розвиватися і з'являється все більше локалізованих ресурсів, можна очікувати, що ми побачимо зростаюче розмаїття типів чат-ботів і додатків, пристосованих до потреб українського бізнесу та споживачів.

1.2.2 Використання великих мовних моделей у чат-ботах

Останніми роками використання великих мовних моделей (LLM) стало перспективним підходом для розробки більш досконалих і ефективних чат-ботів. LLM - це моделі глибокого навчання, навчені на величезних обсягах текстових даних, що дозволяє їм вловлювати тонкощі та нюанси людської мови набагато краще, ніж традиційні методи, засновані на правилах або пошуку. Використовуючи знання та лінгвістичні закономірності, отримані з цих навчальних даних, LLM дозволяють чат-ботам брати участь у більш природних, контекстних і невимушених розмовах.

Однією з ключових переваг використання LLM у чат-ботах є їхня здатність розуміти та генерувати мову, схожу на людську, у широкому діапазоні доменів і тем. Традиційні чат-боти часто мають проблеми з обробкою запитів, які виходять за рамки їхньої компетенції або правил. На відміну від них, LLM можуть спиратися на свою широку базу знань, щоб надавати релевантні та узгоджені відповіді навіть на нові або неочікувані запити користувачів. Така гнучкість дозволяє чат-ботам на базі LLM задовольняти набагато ширший спектр потреб і вподобань клієнтів, покращуючи загальний досвід користувачів.

Ще однією перевагою LLM є їхня здатність до контекстного розуміння та багатостороннього діалогу. Вловлюючи залежності та зв'язки між словами і реченнями, LLM можуть підтримувати узгодженість і послідовність протягом тривалих розмов [26]. Вони можуть враховувати контекст попередніх повідомлень користувача і генерувати відповіді, які природно спираються на цей контекст. Це

дозволяє чат-ботам мати більш цікаву та персоналізовану взаємодію, на відміну від надання загальних або автономних відповідей.

Більше того, LLM можна точно налаштувати або адаптувати до конкретних сфер або завдань за допомогою відносно невеликої кількості додаткових навчальних даних. Ця здатність до трансферного навчання дозволяє розробникам чат-ботів використовувати попередньо навчених LLM і спеціалізувати їх для певних галузей або випадків використання, таких як підтримка клієнтів, продажі або створення контенту [9]. Точне налаштування також допомагає чат-ботам відповідати голосу, тону та термінології бренду організації, створюючи більш плавний та інтегрований користувацький досвід.

Однак використання LLM у чат-ботах також пов'язане з певними проблемами та міркуваннями, особливо в українському контексті. Однією з важливих проблем є обчислювальні ресурси, необхідні для навчання та розгортання великомасштабних мовних моделей. LLM часто мають мільярди параметрів і потребують значних обсягів даних та обчислювальних потужностей для досягнення оптимальної продуктивності. Це може стати бар'єром для багатьох українських організацій, які можуть не мати доступу до необхідної інфраструктури чи експертизи.

Ще одним викликом є потреба у високоякісних даних про українську мову для ефективного навчання та доопрацювання LLM. Незважаючи на зусилля, спрямовані на створення корпусів [17] і наборів даних українською мовою, доступність і різноманітність таких ресурсів все ще обмежені порівняно з основними мовами, такими як англійська чи китайська. Цей дефіцит може вплинути на продуктивність і лінгвістичну компетентність українських чат-ботів на базі LLM, особливо в обробці розмовних виразів, діалектів або специфічної термінології.

Крім того, використання LLM викликає занепокоєння щодо конфіденційності даних, безпеки та етичних міркувань. Оскільки LLM навчаються на великих обсягах даних, отриманих з Інтернету, вони можуть ненавмисно захоплювати та відтворювати конфіденційну або упереджену інформацію [5].

Українські організації повинні переконатися, що їхні чат-боти відповідають відповідним правилам захисту даних та етичним нормам, таким як Загальний регламент про захист даних (GDPR) та Закон України "Про захист персональних даних".

Незважаючи на ці виклики, потенціал LLM у розширенні можливостей чат-ботів сприяє їхньому все більшому впровадженню та дослідженню в Україні. Кілька українських компаній і стартапів вивчають можливості використання LLM для розробки більш досконалих розмовних агентів для різних галузей. Наприклад, український фінтех-стартап Raucell впровадив чат-бота на базі LLM для надання персоналізованих фінансових порад і рекомендацій клієнтам [7]. Аналогічно, українська платформа електронної комерції Rozetka експериментувала з використанням LLM для створення описів товарів і відповідей служби підтримки клієнтів [8].

Українські дослідники також роблять активний внесок у розробку та адаптацію програм підготовки магістрів з української мови. Український католицький університет у співпраці з Національним університетом "Києво-Могилянська академія" розпочав проект зі створення масштабної моделі української мови під назвою UkrBERT. Ця модель, заснована на архітектурі BERT, має на меті забезпечити основу для різних завдань NLP українською мовою, включаючи розробку чат-ботів.

Оскільки українська NLP -спільнота продовжує зростати, а локалізовані ресурси стають все більш доступними, очікується, що використання NLP у чат-ботах зростатиме. Однак успіх цих впроваджень залежатиме від вирішення проблем, пов'язаних з обчислювальними ресурсами, якістю даних та етичними міркуваннями. Українські організації також повинні переконатися, що їхні чат-боти на основі LLM розроблені таким чином, щоб задовольняти специфічні потреби та вподобання українських користувачів, враховуючи культурні, мовні та соціальні фактори.

Отже, використання великих мовних моделей є значним кроком вперед у розробці чат-ботів, що забезпечує більш природне, контекстне та універсальне

спілкування. Хоча впровадження LLM в українських чат-ботах все ще перебуває на ранніх стадіях, потенційні переваги зумовлюють підвищений інтерес та інвестиції в цю сферу. Оскільки українські організації орієнтуються у викликах і можливостях чат-ботів на основі LLM, вони повинні надавати пріоритет довірі користувачів, конфіденційності даних та етичним міркуванням, щоб забезпечити відповідальне та ефективне розгортання цієї трансформаційної технології.

1.3 Основи розробки чат-ботів

1.3.1 Огляд сучасних фреймворків для розробки чат-ботів

Розвиток чат-ботів значно полегшився завдяки появі різних фреймворків і платформ для розробки чат-ботів в останні роки. Ці фреймворки надають необхідні інструменти, бібліотеки та інфраструктуру для створення, навчання та розгортання чат-ботів на різних каналах і в різних сферах використання. Вони абстрагуються від значної частини основної складності і дозволяють розробникам зосередитися на проектуванні розмовного потоку та бізнес-логіки чат-бота. У цьому розділі представлено огляд найпопулярніших і найфункціональніших фреймворків для розробки чат-ботів, що використовуються в Україні, а також порівняння їхніх можливостей, простоти використання та придатності для різних типів чат-бот-проектів.

Однією з провідних платформ для розробки чат-ботів в Україні є Dialogflow - фреймворк від Google, який дозволяє створювати діалогові інтерфейси для веб-сайтів, мобільних додатків та платформ обміну повідомленнями [12]. Dialogflow використовує розуміння природної мови (NLU) для аналізу вводу користувача, вилучення відповідних сутностей та намірів і зіставлення їх із заздалегідь визначеними діями для виконання. Він підтримує широкий спектр інтеграцій, включаючи Google Assistant, Facebook Messenger, Telegram і Viber, що робить його універсальним вибором для українських компаній, які хочуть розгорнути чат-ботів на декількох каналах.

Основні функції Dialogflow включають візуальний інтерфейс перетягування для створення потоків розмов, вбудоване навчання NLU з підтримкою української мови та безперешкодну інтеграцію з хмарними сервісами Google для розширеного функціоналу, такого як аналіз настроїв та розпізнавання мови [12]. Однак, хоча Dialogflow пропонує щедрий безкоштовний рівень, його цінова політика для вищих рівнів використання та корпоративних функцій може стати бар'єром для деяких українських компаній.

Ще одним популярним фреймворком є BotPress - платформа для розробки чат-ботів з відкритим вихідним кодом, яка наголошує на гнучкості та розширюваності [5]. BotPress має модульну архітектуру, яка дозволяє розробникам створювати власні компоненти чат-ботів, інтегруватися з різними механізмами NLU та каналами обміну повідомленнями, а також розширювати функціональність платформи за допомогою плагінів. Він також пропонує зручний візуальний інтерфейс для проектування потоків розмов і вбудовану систему управління контентом для обробки відповідей чат-ботів.

BotPress набув популярності в українській спільноті розробників завдяки своєму відкритому вихідному коду, великій документації та активному форуму користувачів. Кілька українських стартапів та агенцій використовували BotPress для створення чат-ботів для клієнтів у таких галузях, як електронна комерція, охорона здоров'я та освіта [8]. Однак BotPress може вимагати більше технічних знань порівняно з деякими іншими фреймворками, а його продуктивність може залежати від вибору NLU-рушія та хостингової інфраструктури.

Rasa - ще один фреймворк з відкритим вихідним кодом, який набуває популярності в Україні для створення контекстних асистентів і чат-ботів зі штучним інтелектом. Ключова відмінність Rasa полягає в тому, що вона зосереджена на забезпеченні плавного переходу між діалогами зі штучним інтелектом і людськими агентами, що робить її добре придатною для сценаріїв підтримки клієнтів і довідкових служб [4]. Вона надає гнучкий конвеєр для обробки повідомлень користувачів, включаючи попередньо створені компоненти для

класифікації намірів, вилучення сутностей, вибору відповідей та управління діалогом.

Однією з сильних сторін Rasa є підтримка моделей української мови та попередньо навчені вставки слів, що може прискорити розробку україномовних чат-ботів [5]. Rasa також пропонує інтеграцію з популярними платформами обміну повідомленнями, такими як Telegram та Facebook Messenger, а також українськомовними каналами, такими як Viber та WeChat. Однак крива навчання Rasa може бути крутішою, ніж у деяких інших фреймворків, і може вимагати більше зусиль для підготовки даних та навчання моделей.

Окрім цих фреймворків, українські розробники також мають доступ до платформ для розробки чат-ботів, що надаються місцевими технологічними компаніями та стартапами. Наприклад, український стартап у сфері розмовного ШІ Reply.ai пропонує платформу для розробки чат-ботів, яка використовує машинне навчання та обробку природної мови, щоб уможливити розмови, подібні до людських. Платформа Reply.ai підтримує розуміння та генерування української мови, а також інтеграцію з українськими програмами обміну повідомленнями та голосовими помічниками.

Ще однією платформою чат-ботів українського виробництва є BotMaker, яка надає візуальний конструктор для створення чат-ботів без кодування. BotMaker пропонує шаблони та готові компоненти для поширених випадків використання чат-ботів, таких як підтримка клієнтів, лідогенерація та бронювання зустрічей. Він також підтримує інтеграцію з українськими платформами електронної комерції та платіжними системами, що робить його зручним вибором для малого бізнесу та стартапів.

Обираючи фреймворк для розробки чат-ботів, українські компанії повинні враховувати такі фактори, як складність сценарію використання, канали, які вони хочуть підтримувати, рівень кастомізації та контролю, який їм потрібен, а також свій бюджет і ресурси. Такі фреймворки, як Dialogflow і Reply.ai, можуть бути більш придатними для компаній з обмеженим технічним досвідом, тоді як BotPress і Rasa пропонують більшу гнучкість і розширюваність для досвідчених

розробників. Локальні платформи, такі як BotMaker, можуть добре підійти для малого бізнесу та стартапів, які шукають швидкий і доступний спосіб запуснути чат-бота.

Незалежно від обраного фреймворку, українські розробники чат-ботів повинні також враховувати поточний соціально-політичний та технологічний контекст країни. У зв'язку з агресією зі сторони російської федерації та пандемією COVID-19 зріс попит на чат-ботів, які можуть надавати точну та своєчасну інформацію, допомагати у наданні віддалених послуг та підтримувати зусилля з реагування на кризові ситуації [9]. Розробники чат-ботів повинні гарантувати, що їхні рішення є надійними, безпечними та відповідають українському законодавству про захист даних і приватності.

Крім того, оскільки українська мова продовжує розвиватися та адаптуватися до нових сфер і технологій, розробники чат-ботів повинні підтримувати свої мовні моделі та бази знань в актуальному стані. Це може передбачати співпрацю з українськими лінгвістами, постачальниками даних та дослідницькими установами для створення та підтримки високоякісних україномовних ресурсів для розробки чат-ботів [12].

Отже, ландшафт фреймворків для розробки чат-ботів в Україні пропонує широкий вибір варіантів для компаній і розробників, які прагнуть створити рішення для розмовного ШІ. Від глобальних платформ, таких як Dialogflow і Rasa, до вітчизняних рішень, таких як Reply.ai і BotMaker, українські компанії мають доступ до різноманітних інструментів і технологій для створення чат-ботів, які відповідають їхнім конкретним потребам і обмеженням. Однак вибір фреймворку - це лише один з аспектів успішної розробки чат-ботів. Не менш важливими є такі міркування, як якість даних, мовна підтримка, безпека та відповідність вимогам, а також здатність адаптуватися до мінливого технологічного та соціально-політичного контексту України.

1.3.2 Методи і техніки програмування чат-ботів

Розробка ефективних і цікавих чат-ботів вимагає поєднання різних методів і прийомів зі сфер обробки природної мови (NLP), машинного навчання (ML) та програмної інженерії. Ці підходи дозволяють чат-ботам розуміти вхідні дані користувача, генерувати відповідні реакції та інтегруватися із зовнішніми системами та сервісами. Цей розділ містить огляд основних методів і прийомів, що використовуються при програмуванні чат-ботів, з акцентом на їхньому застосуванні в українському контексті.

Одним із фундаментальних завдань у розробці чат-ботів є розпізнавання намірів, яке передбачає визначення наміру або мети користувача на основі його введення. Розпізнавання намірів дозволяє чат-ботам зрозуміти мету повідомлення користувача та спрямувати його до відповідної логіки обробки або модуля генерації відповідей. Поширені підходи до розпізнавання намірів включають системи на основі правил, класифікатори машинного навчання та моделі глибокого навчання.

Розпізнавання намірів на основі правил спирається на заздалегідь визначені шаблони та ключові слова, щоб зіставити вхідні дані користувача з конкретними намірами. Наприклад, система на основі правил може класифікувати вхідні дані, що містять такі слова, як "замовити" (order) або "купити" (buy), як намір купити. Хоча підходи, засновані на правилах, прості в реалізації, вони можуть бути нестійкими і не завжди можуть впоратися з варіаціями або неоднозначністю мови користувача.

Класифікатори машинного навчання, такі як наївний Байєс або машини опорних векторів (SVM), можуть навчитися розпізнавати наміри за маркованими навчальними даними. Ці класифікатори представляють вхідні дані користувача у вигляді векторів ознак і вивчають межі прийняття рішень для розділення різних класів намірів. Класифікатори машинного навчання були застосовані для розпізнавання намірів у різних українських проектах чат-ботів, таких як чат-боти підтримки клієнтів для банків та веб-сайтів електронної комерції [8].

Моделі глибокого навчання, зокрема рекурентні нейронні мережі (RNN) та трансформери, досягли найсучасніших показників у задачах розпізнавання намірів. Ці моделі можуть фіксувати семантичну та контекстну інформацію у вхідних даних користувача та узагальнювати їх до небачених прикладів. Українські дослідники досліджували використання моделей глибокого навчання для розпізнавання намірів в україномовних чат-ботах, використовуючи попередньо навчені вставки слів та методи навчання з перенесенням.

Ще одним важливим аспектом програмування чат-ботів є відповіді на запити, які передбачають отримання відповідної інформації або генерування відповідних відповідей на основі даних, введених користувачем, і його намірів. Відповіді на запити можуть бути реалізовані за допомогою різних методів, таких як системи, засновані на правилах, пошук інформації та генеративні моделі.

Відповіді на запити на основі правил використовують заздалегідь визначені відповідності між намірами та відповідями, подібно до розпізнавання намірів. Для кожного розпізнаного наміру чат-бот вибирає відповідний шаблон відповіді або дію для виконання. Системи на основі правил підходять для простих, структурованих розмов, але їм може бракувати гнучкості та природності.

Інформаційно-пошукові підходи спрямовані на пошук найбільш релевантних відповідей із заздалегідь визначеної бази знань або корпусу текстів. Цього можна досягти за допомогою таких методів, як зіставлення ключових слів, семантична схожість або моделі векторного простору. Українські розробники чат-ботів використовували методи інформаційного пошуку для створення чат-ботів, які можуть відповідати на поширені запитання (FAQ) або надавати інформацію про товари та послуги [3].

Генеративні моделі, такі як моделі "від послідовності до послідовності" або моделі на основі трансформерів, такі як GPT-2, можуть генерувати нові відповіді слово за словом на основі вхідного контексту. Ці моделі навчаються на великих масивах розмовних даних і можуть створювати більш різноманітні та контекстуально відповідні відповіді. Однак генеративні моделі потребують

ретельного навчання та оцінювання, щоб забезпечити якість і безпеку згенерованого контенту.

Інтеграція із зовнішніми системами та сервісами - ще один важливий аспект програмування чат-ботів. Чат-боти часто потребують доступу до даних з різних джерел, таких як бази даних, API або сторонні сервіси, і маніпулюють ними, щоб виконати запити користувачів або персоналізувати розмову. Методи інтеграції включають веб-хуки, виклики API та безсерверні функції.

Веб-хуки дозволяють зовнішнім системам надсилати чат-боту сповіщення або дані в режимі реального часу, що дає йому змогу реагувати на події або оновлення. Наприклад, чат-бот, інтегрований із системою відстеження доставки, може отримувати вебхуки про статус замовлення користувача та своєчасно надавати оновлення [4].

Виклики API дозволяють чат-ботам отримувати або надсилати дані із зовнішніх сервісів, таких як погодні API, платіжні шлюзи або системи управління взаємовідносинами з клієнтами (CRM). Українські розробники чат-ботів використовували API для інтеграції чат-ботів з популярними українськими сервісами, такими як "Нова Пошта" для відстеження доставки, "Приват24" для банківських послуг та "Розетка" для електронної комерції [3].

Безсерверні функції, такі як AWS Lambda або Google Cloud Functions, дозволяють розробникам чат-ботів запускати власну логіку або завдання обробки даних у відповідь на події або дії користувача. Безсерверні функції можна використовувати для реалізації більш складних функцій чат-ботів, таких як персоналізовані рекомендації, аналіз настроїв або мовний переклад [28].

Для створення ефективних чат-ботів українські розробники повинні враховувати лінгвістичні та культурні особливості української мови та суспільства. Це включає в себе роботу зі складною морфологією та синтаксисом української мови, а також знання про українські суб'єкти, місцевості та соціальні норми.

Українські інструменти та ресурси NLP, такі як проект lang-uk та українська Вікіпедія, можуть бути використані для навчання мовних моделей і підвищення точності розпізнавання намірів та відповідей на запити [8]. Крім того, співпраця з

українськими лінгвістами та експертами в галузі може допомогти забезпечити використання чат-ботами відповідної лексики, тону та стилю спілкування для українських користувачів.

Оскільки попит на чат-ботів в Україні зростає, розробники також повинні приділяти пріоритетну увагу безпеці, конфіденційності та етичним аспектам програмування чат-ботів. Це включає в себе впровадження належної автентифікації та контролю доступу, шифрування конфіденційних даних та дотримання правил захисту даних, таких як GDPR.

Крім того, розробники чат-ботів повинні усвідомлювати потенційні ризики та упередження, пов'язані з системами на основі штучного інтелекту, такі як поширення дезінформації, посилення стереотипів або виключення певних груп користувачів. Впровадження таких методів, як змагальне навчання, виявлення упереджень та людський нагляд, може допомогти зменшити ці ризики та гарантувати, що чат-боти будуть безпечними, справедливими та корисними для українських користувачів.

Отже, програмування чат-ботів включає в себе безліч методів і прийомів з NLP, ML та програмної інженерії. Українські розробники мають доступ до зростаючого спектру інструментів, фреймворків та ресурсів для створення чат-ботів, які можуть розуміти наміри користувача, надавати релевантні відповіді та інтегруватися із зовнішніми системами. Однак успіх українських чат-ботів також залежить від здатності адаптуватися до мовних і культурних особливостей українського контексту, а також від дотримання стандартів безпеки, конфіденційності та етики. У міру становлення української екосистеми чат-ботів співпраця між розробниками, дослідниками та експертами галузі матиме вирішальне значення для створення чат-ботів, які справді відповідатимуть потребам та очікуванням українських користувачів.

1.4 Постановка задачі дослідження

Стрімкий розвиток і впровадження технологій чат-ботів в українському секторі послуг створює як значні можливості, так і виклики. Як зазначалося в попередніх розділах, чат-боти на основі великих мовних моделей (LLM) мають потенціал для революції в обслуговуванні, підтримці та залученні клієнтів у таких галузях, як банківська справа, електронна комерція, охорона здоров'я тощо. Автоматизуючи рутинні завдання, надаючи миттєву допомогу 24/7 та пропонуючи персоналізовані рекомендації, чат-боти на основі LLM можуть підвищити операційну ефективність, зменшити витрати та підвищити рівень задоволеності клієнтів для українського бізнесу.

Однак впровадження чат-ботів на основі LLM в Україні не позбавлене труднощів. Однією з головних проблем є брак якісних, різноманітних та актуальних наборів даних про українську мову та ресурсів для навчання та налаштування LLM [23]. Українська мова з її складною морфологією, синтаксисом і лексикою вимагає значних зусиль зі збору, анотування та попередньої обробки даних для створення чат-ботів, здатних розуміти та генерувати природні, контекстно-залежні розмови. Крім того, швидкий розвиток української мови, особливо у сфері технологій та бізнесу, вимагає постійного оновлення та адаптації мовних моделей, щоб йти в ногу з новою термінологією та моделями використання.

Ще однією проблемою є обмежена доступність моделей української мови з відкритим вихідним кодом та попередньо навчених слів, які можна легко інтегрувати у фреймворки для розробки чат-ботів. Хоча існують певні ініціативи зі створення україномовних ресурсів, як-от проект UkrBERT [24], ці зусилля все ще перебувають на ранніх стадіях і можуть бути недостатніми для задоволення різноманітних потреб українського ринку чат-ботів. Така нестача ресурсів може перешкоджати українському бізнесу, особливо малому та середньому, розробляти та впроваджувати сучасних чат-ботів на основі LLM без значних інвестицій у дані та інфраструктуру.

Окрім технічних викликів, існують також соціокультурні та етичні міркування, які необхідно враховувати при розробці та впровадженні чат-ботів на основі LLM в Україні. Оскільки чат-боти стають все більш поширеними і обробляють все більш конфіденційну інформацію, питання конфіденційності, безпеки та прозорості даних виходять на перший план [27]. Українські споживачі можуть вагатися щодо взаємодії з чат-ботами, яким бракує чіткої інформації про методи обробки даних та заходи безпеки. Крім того, чат-боти, які не розроблені з урахуванням українських культурних норм, стилів спілкування та вподобань користувачів, можуть не встановити довіру та взаєморозуміння з користувачами [5].

Сучасний соціально-політичний контекст України також створює унікальні виклики та можливості для розвитку чат-ботів. У зв'язку з агресією зі сторони російської федерації, та пандемією COVID-19 спостерігається сплеск попиту на чат-ботів, які можуть надавати достовірну інформацію, підтримувати віддалені послуги та допомагати в кризових ситуаціях [7]. Однак розробка чат-ботів, здатних впоратися зі складністю та чутливістю цих ситуацій, вимагає ретельного врахування етичних принципів, таких як уникнення упередженості, дезінформації та потенційної шкоди для вразливих груп населення [27].

З огляду на ці виклики, існує нагальна потреба в дослідженні та розробці чат-ботів на базі LLM, адаптованих до конкретних потреб та обмежень українського сектору послуг. Ця робота спрямована на вирішення цієї проблеми, пропонуючи структуру та методологію для створення та розгортання чат-ботів на основі LLM, які є лінгвістично компетентними, культурно релевантними, етично обґрунтованими та технічно надійними.

Метою данної кваліфікаційної роботи є дослідження та вдосконалення розробки україномовних чат-ботів, а також демонстрація їхньої ефективності в секторі послуг України. Для досягнення цієї мети необхідно вирішити наступні завдання:

- дослідити поточний стан україномовних ресурсів та моделей для розробки чат-ботів, виявити прогалини та можливості для вдосконалення;

- розробити методологію збору, анотування та попередньої обробки україномовних даних для навчання та доопрацювання LLM для чат-ботів;
- запропонувати архітектуру та конвеєр для інтеграції LLM у фреймворки для розробки чат-ботів, беручи до уваги специфічні вимоги та обмеження українського сектору послуг, в особливості для ресторанної галузі;
- розробити та впровадити прототип чат-бота на основі LLM для обраної галузі послуг в Україні, продемонструвавши доцільність та ефективність запропонованого підходу;
- оцінити продуктивність, зручність використання та етичні наслідки прототипу чат-бота за допомогою досліджень користувачів та експертних оцінок, а також надати рекомендації щодо майбутніх досліджень і розробок.

Актуальність і значущість цього дослідження полягає в тому, що воно може сприяти розвитку сучасних україномовних технологій та розробці чат-ботів, одночасно вирішуючи практичні потреби та виклики українського сектору послуг. Надаючи основу та методологію для створення чат-ботів на базі LLM, оптимізованих для українського контексту, ця робота може сприяти зростанню та конкурентоспроможності українського бізнесу в глобальній цифровій економіці. Крім того, висвітлюючи етичні та соціально-культурні аспекти розробки чат-ботів, це дослідження може сприяти відповідальним та орієнтованим на людину інноваціям у галузі розмовного ШІ.

2 ВЕЛИКІ МОВНІ МОДЕЛІ (LLM)

2.1 Основи технології трансформерів

2.1.1 Як працюють трансформери

Трансформери з'явилися як революційна архітектура для обробки та розуміння послідовних даних, особливо в галузі обробки природної мови (NLP). Розроблена дослідниками Google у 2017 році [14], модель трансформера представила новий підхід до обробки послідовних даних, покладаючись виключно на механізми уваги, а не на рекурентні або згорткові структури, що використовуються в традиційних нейронних мережах.

Основна інновація трансформерів полягає в їхній здатності фіксувати довгострокові залежності та контекстну інформацію всередині послідовностей, таких як речення або абзаци. Це досягається завдяки використанню механізмів самоуваги, які дозволяють моделі зважувати важливість різних частин вхідної послідовності при обробці конкретного елемента. Враховуючи взаємозв'язки між усіма елементами послідовності, трансформери можуть ефективно моделювати складні лінгвістичні патерни та нюанси.

Архітектура трансформера складається з двох основних компонентів: кодера та декодера (рис.2.1).

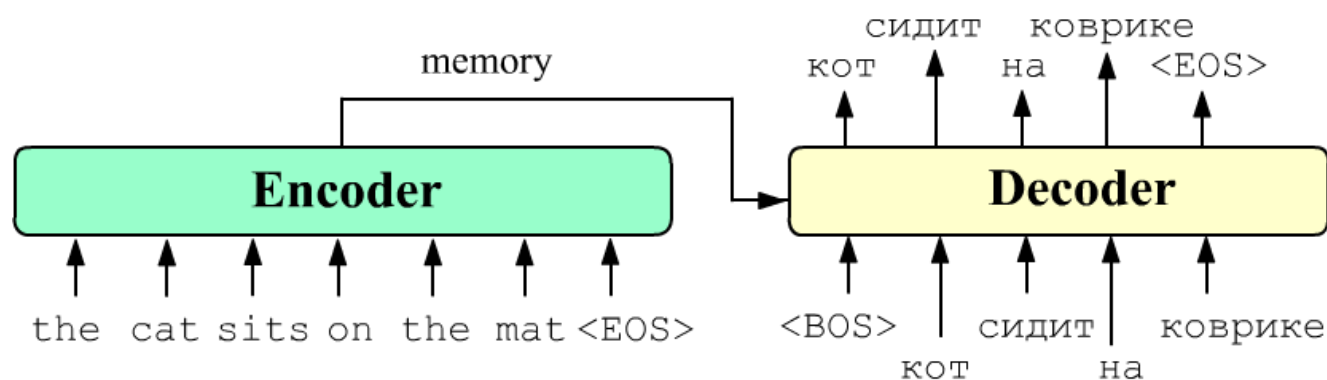


Рисунок 2.1 – Архітектура трансформера

Кодер обробляє вхідну послідовність і генерує послідовність векторних зображень, відомих як «вектори контексту». Ці контекстні вектори фіксують контекстну інформацію та взаємозв'язки у вхідній послідовності. Декодер, з іншого боку, приймає вектори контексту і генерує вихідну послідовність, по одному елементу за раз, враховуючи при цьому раніше згенеровані елементи за допомогою механізму, який називається «прихована самоувага» [13].

В основі архітектури трансформера лежить механізм самоуваги, який дозволяє моделі фокусуватися на різних частинах вхідної послідовності під час обчислення представлення конкретного елемента. Це досягається за допомогою серії обчислень, в яких беруть участь три вектори: вектор запиту, вектор ключа та вектор значення. Вектор запиту представляє поточний елемент, який обробляється, тоді як вектори ключа і значення представляють інші елементи в послідовності. Механізм самоуваги обчислює релевантність кожного елемента в послідовності до поточного елемента, беручи точковий добуток вектора запиту на вектори ключів, а потім застосовуючи функцію `softmax` для отримання набору ваг уваги. Ці ваги уваги потім використовуються для обчислення зваженої суми векторів значень, що призводить до остаточного представлення поточного елемента [13].

Однією з ключових переваг механізму самоуваги є його здатність вловлювати довгострокові залежності у вхідній послідовності. На відміну від рекурентних нейронних мереж (RNN), які обробляють послідовності послідовно і можуть страждати від проблеми зникаючого градієнта при роботі з довгими послідовностями, трансформери можуть звертати увагу на будь-яку частину вхідної послідовності, незалежно від її положення. Це дозволяє трансформерам ефективно моделювати складні лінгвістичні явища, такі як роздільна здатність співвіднесення та семантичні залежності великого діапазону.

Ще однією важливою особливістю трансформерів є їхня здатність до розпаралелювання обчислень, що дозволяє значно підвищити швидкість навчання та виведення. На відміну від ШНМ, які вимагають послідовної обробки вхідної послідовності, трансформери можуть обробляти всі елементи послідовності

одночасно, використовуючи переваги сучасних методів апаратного прискорення, таких як GPU і TPU.

Успіх трансформерів призвів до їх широкого застосування в різних завданнях NLP , включаючи машинний переклад [16], моделювання мови, генерацію тексту та відповіді на запитання. Одним із найпомітніших застосувань трансформерів є розробка великих мовних моделей (LLM), таких як GPT (Generative Pre-trained Transformer) та BERT (Bidirectional Encoder Representations from Transformers) [13].

В українському контексті трансформери та LLM вже показали багатообіцяючі результати в різних завданнях NLP , пов'язаних з українською мовою. Наприклад, дослідники Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» розробили модель української мови під назвою UkrBERT, засновану на архітектурі BERT, яка продемонструвала сучасну продуктивність у таких завданнях, як класифікація тексту, розпізнавання іменованих сутностей та відповіді на запитання [13].

Крім того, українські технологічні компанії та стартапи почали досліджувати використання трансформерів і LLM для різних застосувань, зокрема чат-ботів і віртуальних асистентів. Наприклад, український стартап у сфері розмовного ШІ Reply.ai розробив платформу для чат-ботів, яка використовує мовні моделі на основі трансформерів , щоб забезпечити більш природні та контекстні розмови українською мовою [18].

Однак застосування трансформерів і LLM в українському контексті також висвітлює кілька викликів і міркувань. Однією з головних проблем є брак великих, високоякісних наборів даних українською мовою для ефективного навчання цих моделей. Незважаючи на зусилля, спрямовані на створення українських корпусів і мовних ресурсів, таких як Український національний лінгвістичний корпус [19], доступність і різноманітність цих ресурсів все ще обмежені порівняно з іншими основними мовами.

Іншою проблемою є обчислювальна складність і ресурсні вимоги, пов'язані з підготовкою та розгортанням великих трансформаційних моделей. Ці моделі

можуть мати мільярди параметрів і вимагають значних обчислювальних потужностей та ресурсів пам'яті, які можуть бути недоступними для багатьох українських організацій, особливо малих і середніх підприємств (МСП).

Крім того, використання трансформерів і LLMs викликає етичні занепокоєння щодо конфіденційності даних, безпеки та потенційних упереджень. Оскільки ці моделі навчаються на величезних обсягах даних з Інтернету та інших джерел, існує ризик фіксації та увічнення шкідливих упереджень або конфіденційної інформації. Українські організації повинні забезпечити, щоб їхнє впровадження технологій на основі трансформації відповідало відповідним правилам захисту даних та етичним нормам, таким як Загальний регламент про захист даних (GDPR) та Закон України «Про захист персональних даних».

Незважаючи на ці виклики, потенційні переваги трансформерів і LLM для українського сектору послуг є значними. Забезпечуючи більш природну, контекстну та персоналізовану взаємодію, чат-боти та віртуальні асистенти на основі трансформерів можуть покращити клієнтський досвід, підвищити операційну ефективність та надати цінну інформацію для бізнесу. Оскільки українська екосистема NLP продовжує зростати, а ресурсів стає все більше, можна очікувати, що ми побачимо більш широке впровадження та інновації в застосуванні трансформерів і LLM в різних сферах, включаючи обслуговування клієнтів, електронну комерцію, охорону здоров'я та освіту.

Отже, трансформери зробили революцію в галузі обробки природної мови, запровадивши новий підхід до обробки послідовних даних. Їхні механізми самонавчання та здатність фіксувати довгострокові залежності дозволили розробити потужні мовні моделі, які здатні розуміти і генерувати людський текст. Хоча впровадження трансформерів та LLM в українському контексті стикається з проблемами, пов'язаними з доступністю даних, обчислювальними ресурсами та етичними міркуваннями, їхній потенціал для трансформації сфери послуг та стимулювання інновацій є беззаперечним. Вирішуючи ці проблеми спільними зусиллями дослідників, розробників і політиків, Україна може використати

потенціал трансформерів і магістрів для створення більш інтелектуальних, ефективних і зручних для користувача рішень у сфері розмовного ШІ.

2.1.2 Особливості навчання і функціонування трансформерів

Навчання моделей-трансформерів є обчислювально інтенсивним процесом, який вимагає значних обчислювальних ресурсів і великих наборів даних. Враховуючи складність української мови та обмежену доступність високоякісних даних, навчання трансформерів для українських завдань NLP представляє унікальні виклики, які необхідно вирішувати шляхом ретельного планування та стратегічних підходів.

Процес підготовки перекладачів зазвичай включає два основні етапи: попередню підготовку та доопрацювання. Попереднє навчання - це процес тренування моделі перетворювача на великій кількості немаркованих текстових даних, що дозволяє їй засвоїти загальні мовні шаблони та уявлення [15]. Цей крок має вирішальне значення для побудови надійних та ефективних моделей трансформації, оскільки він дозволяє моделі вловлювати контекстну та семантичну інформацію, що міститься в навчальних даних.

Для української мови на етапі попереднього навчання постає проблема отримання достатньо великого та різноманітного корпусу [17] українських текстових даних. Незважаючи на зусилля, спрямовані на створення корпусів української мови, таких як Український національний лінгвістичний корпус та Український веб-корпус, розмір та якість цих ресурсів не можна порівняти з тими, що доступні для таких мов, як англійська або китайська.

Щоб вирішити цю проблему, українські дослідники та розробники дослідили різні стратегії для збільшення та покращення даних з української мови для попередньої підготовки. Один із підходів полягає у використанні методів навчання з перенесенням, коли попередньо навчені моделі зі споріднених слов'янських мов, таких як російська або польська, допрацьовуються на українських даних [19]. Це

може допомогти пом'якшити проблему дефіциту даних і полегшити перенесення лінгвістичних знань з багатих на ресурси мов на українську.

Інша стратегія полягає у використанні методів доповнення даних, таких як зворотний переклад і генерація тексту, для штучного збільшення обсягу та різноманітності українських навчальних даних [19]. Ці методи передбачають переклад тексту з інших мов на українську [16], а потім використання отриманих перекладів як додаткових навчальних даних.

Після завершення етапу попереднього навчання модель трансформера можна налаштувати для конкретних завдань NLP , таких як класифікація тексту, розпізнавання іменованих сутностей або генерація мови. Точне налаштування передбачає подальше навчання попередньо навченої моделі на меншому наборі даних для конкретної задачі, що дозволяє їй адаптуватися до предметної області та вивчити релевантні для задачі закономірності [5].

В українському контексті етап тонкого налаштування також створює проблеми, пов'язані з доступністю та якістю даних. Для багатьох завдань та доменів NLP може бракувати високоякісних, анотованих українських наборів даних, що ускладнює ефективне налаштування попередньо навчених моделей. Щоб вирішити цю проблему, українські дослідники та розробники досліджували такі стратегії, як краудсорсинг анотацій даних, трансферне навчання із суміжних задач або доменів та використання синтетичних методів генерації даних [29].

На додаток до проблем з даними, навчання трансформерів для українських завдань NLP також вимагає значних обчислювальних ресурсів, включаючи потужні GPU або TPU та великі обсяги пам'яті. Це може стати перешкодою для багатьох українських організацій, особливо малих і середніх підприємств (МСП), які можуть не мати доступу до необхідної апаратної інфраструктури або ресурсів хмарних обчислень.

Щоб пом'якшити цю проблему, українські дослідники та розробники досліджували різні підходи, такі як методи стиснення моделей, розподілене навчання та використання фреймворків та інструментів з відкритим вихідним кодом, оптимізованих для ефективного навчання трансформерів [22]. Крім того,

співпраця між академічними установами, дослідницькими центрами та промисловими партнерами може допомогти об'єднати ресурси та спільно використовувати обчислювальну інфраструктуру для навчання трансформерів .

Іншим важливим аспектом навчання трансформерів є вибір алгоритмів оптимізації та гіперпараметрів. Трансформери зазвичай навчаються з використанням варіантів алгоритму стохастичного градієнтного спуску, таких як Adam або AdamW. Однак оптимальні налаштування гіперпараметрів, такі як графіки швидкості навчання, розміри партій та методи регуляризації, можуть змінюватися залежно від конкретної задачі, набору даних та архітектури моделі. Українські дослідники та розробники повинні ретельно налаштовувати ці гіперпараметри, щоб досягти оптимальної продуктивності та збіжності під час навчання.

Крім того, функціонування трансформерів під час виведення або розгортання також створює виклики в українському контексті. Однією з ключових проблем є потреба в ефективних стратегіях виведення, які можуть забезпечити відповіді в реальному часі, мінімізуючи при цьому обчислювальні витрати і затримки. Такі методи, як квантування моделі, відсікання та динамічна пакетна обробка, були досліджені для оптимізації трансформантного виведення для розгортання в середовищах з обмеженими ресурсами або на периферійних пристроях.

Іншим аспектом є інтеграція трансформантних моделей в існуючі програмні системи та додатки, такі як чат-боти, віртуальні асистенти або конвеєри генерації контенту. Це вимагає ретельної уваги до практики розробки програмного забезпечення, включаючи модульний дизайн, масштабованість і ремонтпридатність, щоб забезпечити безперебійну інтеграцію і довгострокову стійкість рішень на основі трансформерів .

Крім того, розгортання трансформаторних моделей у виробничих середовищах викликає занепокоєння щодо безпеки та конфіденційності, особливо при роботі з конфіденційними або персональними даними. Українські організації повинні впроваджувати надійні заходи безпеки, такі як шифрування даних,

контроль доступу та безпечні конвеєри розгортання моделей, щоб захистити конфіденційність користувачів та відповідати відповідним нормам захисту даних.

Підсумовуючи, можна сказати, що підготовка та функціонування трансформерів для українських завдань NLP представляють унікальні виклики, пов'язані з доступністю даних, обчислювальними ресурсами, оптимізацією та міркуваннями розгортання. Для подолання цих викликів українські дослідники та розробники повинні використовувати стратегічні підходи, такі як трансферне навчання, доповнення даних, стиснення моделей та спільне використання ресурсів. Вирішуючи ці питання та використовуючи можливості трансформаторних моделей, Україна може розкрити потенціал передових технологій NLP та стимулювати інновації в різних сферах, включаючи сферу послуг, охорону здоров'я, освіти та інші.

2.2 Огляд основних LLM

Великі мовні моделі (LLM) з'явилися як трансформаційна технологія, що розширює межі можливого в обробці природної мови та штучному інтелекті. Серед найвідоміших LLM - моделі GPT (Generative Pre-trained Transformer), розроблені OpenAI, і модель Gemini — мультимодальна велика мовна модель, розроблена Google DeepMind, яка є наступником PaLM. Ці моделі привернули значну увагу завдяки своїм вражаючим можливостям і потенційним застосуванням у різних галузях, зокрема в українському секторі послуг.

Моделі GPT. Першопрохідці в LLM Моделі GPT, розроблені OpenAI [22], були в авангарді революції LLM. Перша ітерація, GPT, була представлена у 2018 році і продемонструвала потенціал мовних моделей на основі трансформерів для створення тексту, схожого на людський. Спираючись на цей успіх, OpenAI випустив GPT-2 у 2019 році, більш потужну версію з покращеною продуктивністю на різних мовних завданнях [22].

Однак саме випуск GPT-3 у 2020 році по-справжньому привернув увагу всього світу. Маючи приголомшливі 175 мільярдів параметрів, GPT-3 встановила

нову планку для масштабу та можливостей LLM [22]. Ця модель продемонструвала чудову продуктивність у широкому діапазоні завдань природної мови, включаючи генерацію тексту, відповіді на запитання, генерацію коду і навіть творче письмо.

Однією з ключових переваг GPT-3 є його здатність адаптуватися до різних завдань з мінімальним налаштуванням під конкретне завдання, властивість, відома як «навчання з кількох пострілів». Це робить GPT-3 дуже універсальним і придатним для широкого спектру застосувань - від чат-ботів і віртуальних асистентів до створення контенту та мовного перекладу.

В українському контексті GPT-3 вже показав багатообіцяючі результати в різних завданнях, пов'язаних з мовою. Наприклад, дослідники з КПІ дослідили використання GPT-3 для генерації та узагальнення українського тексту, продемонструвавши його потенціал для покращення створення контенту та обробки інформації українською мовою [4].

Спираючись на успіх GPT-3, OpenAI нещодавно представив GPT-4, більш досконалу та потужну версію моделі [23]. Хоча деталі про архітектуру та процес навчання GPT-4 все ще обмежені, попередні звіти свідчать, що вона перевершує GPT-3 в різних бенчмарках і демонструє покращені міркування та мультимодальні можливості.

Потенційні можливості застосування GPT-4 в українському секторі послуг дуже широкі. Наприклад, він може бути використаний для розробки більш досконалих чат-ботів і віртуальних помічників, які можуть вести більш природні та контекстні розмови, розуміти складні запити та надавати індивідуальні рекомендації. Крім того, GPT-4 можна використовувати для автоматизованого створення контенту, наприклад, для створення маркетингових матеріалів, описів продуктів або навіть для написання художніх текстів українською мовою.

Крім того, здатність Gemini інтегрувати знання з конкретних галузей робить його придатним для спеціалізованих застосувань у таких секторах, як охорона здоров'я, фінанси та електронна комерція. Наприклад, чат-бот на основі Gemini може бути навчений надавати надійну і точну інформацію про медичні умови,

фінансові продукти або запити щодо онлайн-покупок, використовуючи своє розуміння української мови та знання специфічних галузей.

Хоча GPT і Gemini є одними з найвідоміших LLM, кілька інших моделей також отримали визнання завдяки своїм можливостям і потенційним застосуванням. Однією з таких моделей є BERT (Bidirectional Encoder Representations from Transformers), розроблена дослідниками з Google [10].

BERT - це попередньо навчена мовна модель, яка досягла найсучасніших результатів у різних завданнях NLP, включаючи класифікацію текстів, відповіді на запитання та розпізнавання іменованих сутностей. Хоча BERT не призначена для генерації текстів, як GPT, вона успішно застосовується в різних українських завданнях NLP, таких як аналіз настроїв, узагальнення текстів та пошук інформації [11].

Іншою відомою ШНМ є XLNet, розроблена дослідниками з Google Brain та Університету Карнегі-Меллона [12]. XLNet включає в себе кілька архітектурних удосконалень порівняно з BERT, включаючи використання перестановочної мови моделювання мети і новий механізм уваги. Ці вдосконалення дозволили XLNet досягти чудової продуктивності на різних тестах NLP, включаючи ті, що стосуються української мови [13].

В українському контексті XLNet досліджували для таких завдань, як класифікація тексту, розпізнавання іменованих об'єктів та відповіді на запитання. Здатність XLNet фіксувати довгострокові залежності та контекстну інформацію робить його особливо придатним для застосування у сфері послуг, де розуміння складних запитів та надання точних відповідей має вирішальне значення.

Іншим перспективним LLM є T5 (Text-to-Text Transfer Transformer), розроблений дослідниками з Google Brain [14]. T5 - це уніфікована модель, яка може бути застосована до різних завдань природної мови, включаючи переклад, узагальнення та відповіді на запитання, формулюючи їх як задачі перетворення тексту в текст.

T5 продемонстрував вражаючу продуктивність на кількох українських завданнях NLP, включаючи резюме тексту, машинний переклад і генерацію тексту

[15]. Здатність вирішувати різноманітні завдання за допомогою єдиної архітектури моделі робить її привабливим варіантом для розробки багатоцільових мовних додатків в українському секторі послуг.

Оскільки ці моделі навчаються на величезних обсягах даних з Інтернету та інших джерел, вони можуть ненавмисно засвоювати і посилювати суспільні упередження, стереотипи або неточну інформацію, наявну в навчальних даних. Це може призвести до створення упередженого або оманливого контенту, що може бути особливо шкідливим у таких чутливих сферах, як охорона здоров'я, фінанси або юридичні послуги.

Щоб зменшити цей ризик, українські дослідники та розробники повинні визначити пріоритетом розробку надійних методів виявлення та пом'якшення упередженості, а також кураторство високоякісних, неупереджених навчальних даних. Крім того, прозорість і чітке інформування про обмеження та потенційні упередження LLM мають вирішальне значення для побудови довіри та відповідального використання.

Ще однією етичною проблемою є потенційний вплив LLM на конфіденційність і безпеку даних. Оскільки ці моделі обробляють і генерують великі обсяги текстових даних, існує ризик розкриття конфіденційної особистої інформації або даних, що є власністю компанії. Українські організації повинні впроваджувати надійні заходи захисту даних, такі як шифрування, контроль доступу та безпечні конвеєри розгортання моделей, щоб захистити конфіденційність користувачів та відповідати відповідним нормам захисту даних [18].

Крім того, розробка і розгортання LLM піднімає питання щодо прав інтелектуальної власності, власності та справедливої компенсації. Оскільки ці моделі генерують текст на основі своїх навчальних даних, існує ризик ненавмисного відтворення контенту, захищеного авторським правом або власністю. Українські політики та зацікавлені сторони повинні встановити чіткі керівні принципи та рамки для вирішення цих питань і забезпечення справедливих та етичних практик у розробці та використанні LLMs.

Крім того, обчислювальні ресурси, необхідні для навчання та розгортання LLM, можуть мати значний вплив на навколишнє середовище, сприяючи викидам вуглецю та споживанню енергії. Українські організації та дослідники повинні надавати пріоритет сталим практикам, таким як використання відновлюваних джерел енергії, оптимізація архітектури моделей для підвищення ефективності та вивчення методів стиснення та квантування моделей [19].

Нарешті, стрімкий розвиток інформаційних програм та їхній потенційний вплив на різні галузі, включаючи сферу послуг, викликає занепокоєння щодо витіснення робочих місць і необхідності перекваліфікації та адаптації робочої сили. Хоча LLM можуть автоматизувати певні завдання та підвищити операційну ефективність, дуже важливо враховувати потенційний вплив на суспільство та розробляти стратегії підвищення кваліфікації та перекваліфікації працівників, які постраждали від цього, до нових ролей та обов'язків [20].

Великі мовні моделі, такі як GPT, Gemini та інші, відкрили нову еру обробки природної мови та штучного інтелекту. Їхні вражаючі можливості в розумінні та створенні людського тексту відкрили світ можливостей для різних застосувань, включаючи чат-боти, віртуальних асистентів, створення контенту та спеціалізовані завдання для конкретних доменів.

В українському контексті LLM відкривають значні можливості для розвитку сфери послуг, покращення клієнтського досвіду та стимулювання інновацій. Такі моделі, як GPT-3 та GPT-4, пропонують потужні можливості генерування та розуміння мови, які можуть бути використані для розробки передових чат-ботів та віртуальних асистентів. Тим часом, локально розроблені моделі, такі як Gemini, пропонують індивідуальне рішення для завдань з української мови, що включає культурні та контекстуальні знання, характерні для українського контексту.

Однак розробка та впровадження LLM також пов'язані з етичними міркуваннями та проблемами, які необхідно вирішувати. Питання упередженості, конфіденційності, прав інтелектуальної власності, впливу на навколишнє середовище та переміщення робочої сили є вирішальними факторами, які

потребують ретельного розгляду та проактивних заходів з боку українських дослідників, розробників, політиків та зацікавлених сторін.

3 ДЕТАЛІЗАЦІЯ ІДЕЇ ЧАТ-БОТА

3.1 Функціональні вимоги до чат-бота

3.1.1 Опис інформації, яка буде використовуватися ботом

Чат-бот буде використовувати різні джерела інформації для надання персоналізованих і точних послуг клієнтам в українському секторі послуг. Ці джерела включають:

Меню ресторану та інформацію про страви: чат-бот матиме доступ до всеосяжної бази даних, що містить детальну інформацію про пропозиції меню ресторану, зокрема:

- назви та описи страв;
- інгредієнти та інформація про алергени;
- дані про поживність (наприклад, калорійність, макроелементи);
- ціни та доступність;
- категорії страв (наприклад, супи, салати, основні страви);
- дієтичні маркування (наприклад, веганські, безглютенові, халяльні).

Ця інформація буде отримана з внутрішньої системи управління меню ресторану та регулярно оновлюватиметься для забезпечення точності та актуальності. Дані меню будуть використовуватися чат-ботом для надання користувачам детальної інформації про варіанти страв, відповідей на питання, пов'язані з інгредієнтами та дієтичними вподобаннями, а також для полегшення процесу замовлення. Приклад структури даних, що описує страви з меню наведено в лістингу 3.1.

Лістинг 3.1 – Структура даних, що описує страви з меню

```
{
  "dishes": [
    {
      "id": "d1",
```

```
    "name": "Борщ",
    "description": "Традиційний український борщ із буряком, картоплею та квасолею.",
    "category": "Супи",
    "price": 75,
    "ingredients": ["буряк", "картопля", "квасоля", "капуста", "морква", "цибуля", "часник", "томатна паста", "сметана"]
  },
  {
    "id": "d2",
    "name": "Вареники з картоплею",
    "description": "Домашні вареники з картопляною начинкою, подані зі смаженою цибулею.",
    "category": "Основні страви",
    "price": 95,
    "ingredients": ["борошно", "картопля", "цибуля", "вершкове масло"]
  },
  ...
]
```

Дані клієнта: щоб запропонувати персоналізоване обслуговування та покращити користувацький досвід, чат-бот збиратиме та оброблятиме різні типи даних клієнтів, зокрема:

а) профілі користувачів:

- особиста інформація (наприклад, ім'я, контактні дані);
- дієтичні вподобання та обмеження (наприклад, вегетаріанство, алергія);
- улюблені страви та кухні;
- мовні та комунікаційні вподобання;

б) історія замовлень:

- попередні замовлення та деталі замовлень;
- найчастіше замовлені позиції;
- відгуки та оцінки про страви;

с) відгуки та оцінки:

– відгуки та оцінки користувачів про страви, обслуговування та загальний досвід;

– пропозиції та скарги.

Дані про клієнтів будуть збиратися за допомогою різних каналів, таких як реєстраційні форми користувачів, взаємодія в чаті та інтеграція з існуючими системами управління взаємовідносинами з клієнтами (CRM). Ця інформація буде надійно зберігатися та оброблятися згідно з відповідними правилами захисту даних, такими як Загальний регламент про захист даних (GDPR) та Закон України «Про захист персональних даних». Дані клієнтів будуть використовуватися чат-ботом для надання персоналізованих рекомендацій, пропонування страв на основі дієтичних уподобань та попередніх замовлень, а також для полегшення повторного замовлення улюблених позицій. Крім того, відгуки та рецензії користувачів будуть аналізуватися для покращення пропозицій меню, якості обслуговування та загального клієнтського досвіду. Приклад структури даних, що описує дані клієнта наведено в лістингу 3.2.

Лістинг 3.2 – Структура даних, що описує дані клієнта

```
{
  "customers": [
    {
      "id": "c1",
      "name": "Андрій Петренко",
      "email": "andrii.petrenko@example.com",
      "allergies": ["Milk", "Nuts"],
      "dietaryPreferences": ["Vegan"],
      "orderHistory": [
        {
          "orderId": "o1",
          "date": "2023-05-15",
          "items": [
            {
              "dishId": "d7",
              "quantity": 1
            },
            {
              "dishId": "d12",
              "quantity": 2
            }
          ]
        }
      ]
    }
  ]
}
```

```
    ]
    },
    ...
  ]
},
...
]
```

Зовнішні джерела даних: щоб розширити свої знання та можливості, чат-бот інтегрується з різними зовнішніми джерелами даних, такими як:

a) онлайн-бази даних про продукти харчування та кулінарні ресурси:

- бази даних про харчування;
- бази даних інгредієнтів та алергенів;
- сховища рецептів і кулінарні блоги;
- аналіз харчових тенденцій та звіти;

b) календарі та каталоги місцевих подій:

- інформація про місцеві фестивалі, свята та культурні заходи;
- списки найближчих визначних пам'яток та цікавих місць;

c) дані про погоду і дорожній рух:

– інформація про погоду в реальному часі для планування доставки та розміщення на відкритому повітрі;

– оновлення даних про дорожній рух для оптимізації маршрутів доставки та оцінки часу прибуття;

d) соціальні мережі та платформи відгуків:

- користувацький контент та відгуки про ресторан і його страви;
- аналіз настроїв і моніторинг тенденцій.

Використовуючи ці зовнішні джерела даних, чат-бот може надавати користувачам більш повну інформацію, персоналізовані рекомендації та контекстну обізнаність. Наприклад, він може запропонувати страви, які підходять для певної культурної події, або порекомендувати варіанти сидіння на відкритому повітрі, виходячи з погодних умов.

Використовуючи мовні моделі та бази знань, чат-бот зможе розуміти та генерувати природні, контекстно-залежні розмови українською мовою, а також враховувати специфічні знання та культурні нюанси, що стосуються сфери послуг та місцевого контексту.

Використовуючи цю комплексну інформаційну екосистему, чат-бот може надавати персоналізовані, контекстно-орієнтовані та культурно релевантні послуги українським клієнтам, покращуючи їхній загальний досвід відвідування ресторану та сприяючи довірі та взаємодії з рестораном. Приклад запису до бази знань наведено в лістингу 3.3.

Лістинг 3.3 – Приклад запису до бази знань

```
{
  "id": "kb1",
  "title": "Борщ",
  "content": "Борщ є національною стравою України та однією з найвідоміших українських страв у світі. Це густий овочевий суп із буряком як основним інгредієнтом. Традиційно борщ готується з додаванням картоплі, капусти, моркви, цибулі, часнику та квасолі. Він подається зі сметаною та часниковими грінками. Борщ має багату історію та різноманітні регіональні варіації в Україні. Він вважається не лише смачною, а й корисною стравою завдяки високому вмісту вітамінів та мінералів.",
  "category": "Українська кухня",
  "tags": ["суп", "традиційна страва", "буряк", "овочі"]
}
```

Поєднання даних меню, інформації про клієнта, деталей ресторану та всеосяжної бази знань дозволить чат-боту вести природні розмови, надавати індивідуальні рекомендації, полегшувати процес замовлення та покращувати загальний клієнтський досвід, враховуючи місцеві вподобання та культурні нюанси.

3.1.2 Послуги, які надає чат-бот

Чат-бот, розроблений на базі LLM, надаватиме комплексний набір послуг, спрямованих на покращення взаємодії з клієнтами та оптимізацію операцій в

українському секторі послуг. Використовуючи свої можливості розуміння та генерації природної мови, чат-бот буде вести природні, контекстно-залежні розмови з клієнтами, пропонуючи їм персоналізовану допомогу та підтримку протягом усього їхнього шляху до ресторану. Основні послуги, що надаються чат-ботом, такі:

Розмовне замовлення:

Однією з основних функцій чат-бота є полегшення процесу замовлення за допомогою природної мовної взаємодії. Клієнти можуть робити замовлення, змінювати свій вибір та запитувати про пункти меню, просто розмовляючи з чат-ботом українською мовою. Чат-бот проведе клієнта через процес замовлення, надаючи детальну інформацію про страви, інгредієнти, дієтичні варіанти та ціни.

Навігація по меню та інформація про страви: чат-бот оснащений великою базою знань, що містить вичерпну інформацію про пропозиції меню ресторану. Клієнти можуть запитувати про конкретні страви, їхні інгредієнти, інформацію про алергени, поживну цінність та способи приготування. Чат-бот відповість точною та актуальною інформацією, гарантуючи, що клієнти зроблять усвідомлений вибір з урахуванням своїх дієтичних уподобань та обмежень.

Розміщення та зміна замовлень: за допомогою природної мови клієнти можуть розміщувати замовлення, вказуючи бажані страви та їхню кількість. Чат-бот підтвердить деталі замовлення, включно з будь-якими налаштуваннями чи особливими побажаннями, що полегшить процес оформлення замовлення. Клієнти також можуть змінювати свої замовлення, додаючи, видаляючи або замінюючи позиції, а чат-бот плавно оновить деталі замовлення.

Відстеження та оновлення статусу замовлення: Після розміщення замовлення чат-бот оновлює його статус в режимі реального часу, починаючи з підготовки і закінчуючи доставкою або самовивозом. Клієнти можуть запитати про очікуваний час очікування, а чат-бот буде використовувати інтеграцію з системою управління кухнею ресторану, щоб надати точну та актуальну інформацію.

Персоналізовані рекомендації:

Використовуючи дані та вподобання клієнтів, чат-бот пропонує персоналізовані рекомендації щодо страв, адаптовані до індивідуальних смаків та дієтичних вимог. Ця послуга має на меті покращити досвід відвідування ресторану, знайомлячи клієнтів з новими та цікавими варіантами меню, які відповідають їхнім уподобанням.

Рекомендації на основі вподобань: аналізуючи профілі клієнтів, включаючи дієтичні вподобання, алергії та історію попередніх замовлень, чат-бот пропонує страви, які відповідають їхнім конкретним потребам і смакам. Наприклад, якщо клієнт має непереносимість глютену, чат-бот порекомендує безглютенові варіанти, або якщо клієнт часто замовляє вегетаріанські страви, чат-бот надасть пріоритет вегетаріанським рекомендаціям.

Сезонні та акційні рекомендації: чат-бот в курсі сезонних пропозицій меню ресторану, спеціальних акцій та страв з обмеженим терміном дії. Він проактивно рекомендуватиме їх клієнтам, підкреслюючи їхні унікальні особливості, інгредієнти та смаки, покращуючи загальний досвід відвідування ресторану та заохочуючи клієнтів досліджувати нові кулінарні пригоди.

Контекстні рекомендації: інтегруючись із зовнішніми джерелами даних, такими як місцеві календарі подій та інформація про погоду, чат-бот може надавати контекстно-релевантні рекомендації. Наприклад, під час культурного фестивалю чат-бот може запропонувати традиційні українські страви або напої, пов'язані з цією подією. Аналогічно, у теплий літній день чат-бот може порекомендувати освіжаючі салати або прохолодні напої, які підійдуть для обіду на свіжому повітрі.

Дієтична та алергологічна допомога:

Забезпечення безпеки клієнтів та задоволення різноманітних дієтичних потреб є важливим аспектом послуг чат-бота. Використовуючи свою всеосяжну базу знань та інтеграцію з даними меню, чат-бот надаватиме точну інформацію та рекомендації щодо дієтичних обмежень та алергенів.

Ідентифікація алергенів та попередження: клієнти можуть запитати про конкретні алергени, присутні у стравах, і чат-бот надасть детальну інформацію, включаючи списки інгредієнтів та потенційні ризики перехресного забруднення.

Якщо клієнт обирає страву, яка містить вказаний ним алерген, чат-бот видає чітке попередження, що дозволяє йому прийняти усвідомлене рішення.

Рекомендації згідно дієтичних уподобань клієнта: для клієнтів, які дотримуються певних дієт, наприклад, вегетаріанських, веганських, халяльних або кошерних, чат-бот запропонує індивідуальні рекомендації та підказки. Він виділятиме пункти меню, які відповідають їхнім дієтичним вподобанням, забезпечуючи безперешкодний і приємний обідній досвід.

Інформація про харчування: чат-бот матиме доступ до вичерпних даних про поживність кожного пункту меню, включаючи підрахунок калорій, розбивку на макроелементи та інформацію про конкретні харчові компоненти (наприклад, глютен, лактозу, натрій). Клієнти можуть запитувати цю інформацію, щоб зробити усвідомлений вибір на основі своїх дієтичних цілей або проблем зі здоров'ям.

Відгуки та залучення споживачів:

Сприяння відкритому спілкуванню та збору відгуків клієнтів має вирішальне значення для постійного вдосконалення та покращення загального клієнтського досвіду. Чат-бот слугуватиме платформою, де клієнти зможуть ділитися своїми думками, пропозиціями та відгуками безпосередньо з рестораном.

Оцінки та відгуки: після відвідування ресторану клієнти можуть оцінити та залишити відгук про конкретні страви, якість обслуговування та загальну задоволеність через інтерфейс чат-бота. Ці відгуки будуть надійно збережені та проаналізовані, щоб визначити сфери для вдосконалення та виділити популярні позиції меню або винятковий сервіс.

Робота зі скаргами та пропозиціями: якщо клієнти стикаються з будь-якими проблемами або мають пропозиції щодо покращення, вони можуть безпосередньо спілкуватися з чат-ботом. Чат-бот перенаправляє ці скарги та пропозиції відповідному персоналу або відділам, забезпечуючи швидке вирішення та подальші дії.

Програми лояльності та акції: чат-бот може слугувати каналом для інформування клієнтів про переваги програм лояльності, спеціальні акції та ексклюзивні пропозиції. Використовуючи дані та вподобання клієнтів, чат-бот

може персоналізувати ці повідомлення, підвищуючи залученість та лояльність клієнтів.

Опитування та анкетування: чат-бот може полегшити збір інформації про клієнтів за допомогою опитувань, збираючи цінні дані про вподобання, тенденції та очікування. Ця інформація може бути використана для оновлення меню, покращення обслуговування та прийняття загальних стратегічних рішень для ресторану.

Інтеграція з операційною діяльністю ресторану:

Для забезпечення безперебійної роботи та ефективного надання послуг чат-бот буде інтегрований з різними системами та процесами в інфраструктурі ресторану. Ця інтеграція уможливить обмін даними в режимі реального часу, автоматизацію рутинних завдань і спростить комунікацію між різними відділами та зацікавленими сторонами.

Інтеграція кухні та управління замовленнями: чат-бот буде підключений до кухні ресторану та системи управління замовленнями, що дозволить безперешкодно розміщувати, відстежувати та виконувати замовлення. Коли замовлення надходять через чат-бота, вони автоматично надсилаються на кухню для приготування, мінімізуючи ручне втручання та зменшуючи ризик помилок.

Управління запасами та ланцюгами поставок: інтегруючись із системами управління запасами, чат-бот може надавати інформацію про наявність позицій меню в режимі реального часу та пропонувати альтернативні варіанти, якщо певні інгредієнти або страви тимчасово відсутні. Ця інтеграція також може полегшити прогнозування замовлень та оптимізацію ланцюжка поставок на основі попиту та вподобань клієнтів, отриманих за допомогою чат-бота.

Інтеграція з системою управління взаємовідносинами з клієнтами (CRM): чат-бот буде взаємодіяти з CRM-системою ресторану, що дозволить безперешкодно управляти даними клієнтів і персоналізувати їх. Профілі клієнтів, історії замовлень, вподобання та відгуки, зібрані за допомогою чат-бота, будуть надійно зберігатися і будуть доступні для авторизованого персоналу, що дозволить

проводити цільові маркетингові кампанії, управляти програмами лояльності та надавати індивідуальні пропозиції послуг.

Інтеграція оплати та доставки: для додаткової зручності чат-бот може бути інтегрований з платіжними шлюзами та службами доставки, що дозволяє клієнтам здійснювати транзакції та домовлятися про доставку або самовивіз безпосередньо через інтерфейс чат-бота. Цей спрощений процес мінімізує тертя та покращує загальний клієнтський досвід.

Надаючи ці комплексні послуги, чат-бот на базі LLM докорінно змінить клієнтський досвід в українському секторі послуг, пропонуючи персоналізовану допомогу, підвищуючи операційну ефективність та сприяючи зміцненню довіри та взаємодії між бізнесом та його клієнтами. Завдяки постійному вдосконаленню та адаптації до мінливих потреб та вподобань українського ринку, чат-бот позиціонуватиме себе як життєво важливий компонент сучасного ресторанного досвіду, встановлюючи нові стандарти якості обслуговування та задоволеності клієнтів. В чат-боті, реалізованому в даній роботі реалізовані функції розмовного замовлення, навігація по меню та надання інформації про страви, персоналізовані рекомендації на основі попередніх замовлень клієнта (враховуються дні оформлення замовлень, патерни обирання страв), ідентифікація алергенів та попередження, рекомендації згідно дієтичних уподобань клієнта.

3.2 Опис алгоритму роботи чат-бота

3.2.1 Сценарії взаємодії з користувачем

Чат-бот на базі LLM буде вести природні, контекстно-орієнтовані розмови з клієнтами, проводячи їх через різні сценарії та надаючи персоналізовану допомогу протягом усього часу перебування в ресторані. Нижче наведено типові сценарії взаємодії з користувачем, з якими працюватиме чат-бот:

Привітання та первинний запит:

Чат-бот привітає клієнта в дружній і культурно прийнятній манері, використовуючи українські мовні конвенції та місцеві ідіоми.

Потім він запитує клієнта про його наміри або про те, як він може допомогти йому, наприклад, переглянути меню, зробити замовлення або отримати рекомендації.

Приклад розмови:

Чат-бот: «Вітаю! Я чат-бот ресторану «Смачна Хата». Чим я можу вам допомогти сьогодні?»

Клієнт: «Доброго дня! Я хотів би переглянути ваше меню.»

Чат-бот: «Звичайно! Ось наше меню з описами страв і цінами. Чи є у вас якісь особливі вподобання або дієтичні обмеження, про які я маю знати?»

Навігація по меню та вибір страв:

Чат-бот презентує клієнту меню, або показуючи його повністю, або проводячи його по різних категоріях та опціях відповідно до його вподобань.

Клієнти можуть запитувати про конкретні страви, інгредієнти, алергени або дієтичну інформацію, а чат-бот надаватиме детальні та точні відповіді.

Якщо клієнт вибере страву, яка містить вказаний ним алерген, чат-бот видасть чітке попередження і запропонує альтернативні варіанти.

Приклад розмови:

Клієнт: «Я хотів би замовити борщ і вареники з картоплею.»

Чат-бот: «Чудовий вибір! Наш борщ готується за традиційним українським рецептом з буряком, картоплею, квасолею та іншими овочами. Вареники з картоплею подаються зі смаженою цибулею. Чи є у вас якісь харчові алергії або дієтичні обмеження, про які я маю знати?»

Клієнт: «Так, я не вживаю молочні продукти.»

Чат-бот: «Зрозуміло. Хочу попередити, що вареники з картоплею готуються з використанням вершкового масла. Можливо, ви хотіли б замовити іншу веганську страву замість вареників?»

Оформлення та модифікація замовлення:

Після того, як клієнт обрав бажані страви, чат-бот проведе його через процес замовлення, підтверджуючи деталі та дозволяючи внести будь-які зміни чи особливі побажання.

Чат-бот також надасть приблизний час приготування та доставки/самовивозу, а також варіанти оплати.

Якщо клієнт захоче змінити своє замовлення, чат-бот плавно оновить деталі замовлення та підтвердить зміни.

Приклад розмови:

Чат-бот: «Дозвольте підтвердити ваше замовлення: борщ і грибна юшка. Чи бажаєте ви додати ще якісь страви або напої?»

Клієнт: «Так, я хотів би також замовити вареники з вишнями».

Чат-бот: «Зрозуміло. Я додав до вашого замовлення вареники з вишнями. Чи є ще якісь зміни або доповнення?»

Клієнт: «Ні, це все.»

Чат-бот: «Дякую! Ваше замовлення буде готове приблизно через 30 хвилин. Ви можете сплатити готівкою або карткою під час отримання замовлення.» (чат бот не надає суму замовлення в поточному вигляді імплементації)

Персоналізовані рекомендації.

Чат-бот використовує дані клієнта, його вподобання та історію попередніх замовлень, щоб надати персоналізовані рекомендації щодо страв, які відповідають індивідуальним смакам та дієтичним потребам.

Він також запропонує сезонні, акційні або культурно релевантні страви, засновані на місцевих подіях, святах або трендах.

Приклад розмови:

Чат-бот: «На основі ваших попередніх замовлень і уподобань, я б порекомендував спробувати нашу нову сезонну страву - гречаники з грибами та цибулевою підливою. Ця страва готується з екологічно чистих продуктів і відповідає вашим вегетаріанським вподобанням.»

Клієнт: «Звучить смачно! Я із задоволенням її спробую.»

Чат-бот: «Чудово! Також хотів би нагадати, що наступного тижня у нас буде святкове меню з нагоди Дня Незалежності України. Ми пропонуватимемо традиційні українські страви, такі як галушки, банош та вергуни. Не пропустіть цю можливість скуштувати автентичні національні смаки!»

Дієтична та алергологічна допомога:

Чат-бот надаватиме точну інформацію та рекомендації щодо дієтичних обмежень та алергенів, щоб клієнти могли зробити усвідомлений вибір відповідно до своїх потреб та вподобань.

Він виділятиме страви, які відповідають певним дієтичним моделям, наприклад, вегетаріанські, веганські, халяльні або кошерні, і надаватиме інформацію про їхню поживну цінність за запитом.

Приклад розмови:

Клієнт: «У мене непереносимість лактози. Чи є у вас безлактозні страви?»

Чат-бот: «Звичайно! Ми маємо широкий вибір безлактозних страв у нашому меню. Наприклад, наш борщ, грибна юшка та овочеве рагу готуються без молочних продуктів. Я також можу порекомендувати вам наші веганські котлети з гречаної каші та гарніром із запечених овочів.»

Клієнт: «Чудово, дякую за інформацію. Я б хотіла замовити веганські котлети з гарніром.»

Чат-бот: «Зрозуміло. Веганські котлети з гречаної каші та запечені овочі додані до вашого замовлення. Чи бажаєте ви отримати інформацію про поживну цінність цієї страви?»

Відгуки та залучення клієнтів:

Після відвідування ресторану чат-бот запропонує клієнтам залишити оцінки, відгуки та відгуки про конкретні страви, якість обслуговування та загальну задоволеність.

Він також полегшить подання скарг, пропозицій або компліментів, забезпечуючи швидке вирішення та подальші дії відповідного персоналу.

Приклад розмови:

Чат-бот: «Дякуємо, що скористалися нашими послугами! Ми цінуємо ваш відгук. Будь ласка, оцініть свій досвід від 1 до 5 зірок і залиште коментар, якщо маєте якісь зауваження або пропозиції.»

Клієнт: «Я б поставив 4 зірки. Їжа була смачною, але обслуговування було трохи повільним.»

Чат-бот: «Дякуємо за чесний відгук! Ми обов'язково передамо вашу пропозицію щодо покращення швидкості обслуговування керівництву ресторану. Ми також будемо раді отримати більш детальні коментарі, якщо ви маєте час їх залишити.»

Інтеграція з операційною системою ресторану:

Чат-бот легко інтегрується з різними системами та процесами в інфраструктурі ресторану, забезпечуючи обмін даними в режимі реального часу, автоматизацію рутинних завдань і налагодження комунікації між різними відділами та стейкхолдерами.

Приклад сценарію:

Клієнт робить замовлення через чат-бота, яке автоматично перенаправляється в систему управління кухнею для приготування.

Чат-бот надає клієнту приблизний час приготування на основі даних з кухні в режимі реального часу.

Коли замовлення готове, чат-бот повідомляє про це клієнта і полегшує процес оплати через інтегровані платіжні шлюзи.

Чат-бот також оновлює систему управління запасами ресторану замовленими позиціями, запускаючи автоматичне повторне замовлення інгредієнтів, яких бракує на складі.

Завдяки здатності розуміти та генерувати природну мову, чат-бот на базі LLM забезпечить безперебійний та персоналізований досвід взаємодії з користувачами в українському секторі послуг. Його здатність вести контекстно-орієнтовані розмови, пропонувати індивідуальні рекомендації та інтегруватися з різними ресторанными системами зробить революцію в обслуговуванні клієнтів та

підвищить операційну ефективність, сприяючи зміцненню довіри та задоволеності серед українських споживачів.

3.2.2 Обробка і реагування на запити користувачів

Основна функціональність чат-бота на базі LLM полягає в обробці запитів користувачів та генеруванні контекстно-релевантних відповідей у режимі реального часу. Цей процес включає кілька ключових компонентів і технологій, які працюють у тандемі для забезпечення безперебійної та інтелектуальної взаємодії з клієнтами:

1) обробка природної мови (NLP): чат-бот використовує передові методи NLP для аналізу та розуміння запитів користувачів українською мовою. Цей процес складається з кількох етапів:

а) попередня обробка тексту: перш ніж аналізувати вхідні дані користувача, чат-бот виконує завдання з попередньої обробки тексту, щоб очистити та нормалізувати його. Це включає токенізацію (розбиття тексту на окремі слова або токени), видалення стоп-слів (загальних слів на кшталт «і», «той» тощо), обробку пунктуації, а також стеммінг/лематизацію (приведення слів до їхньої базової форми);

б) класифікація намірів: чат-бот використовує моделі машинного навчання, такі як машини опорних векторів (SVM) або архітектури глибокого навчання, такі як рекурентні нейронні мережі (RNN) або трансформери, щоб класифікувати наміри користувача на основі його введення. Це передбачає зіставлення запиту користувача із заздалегідь визначеними намірами або категоріями, такими як «розміщення замовлення», «запит меню» або «дієтичні обмеження»;

в) розпізнавання сутностей: на додаток до визначення намірів користувача, чат-бот використовує методи розпізнавання іменованих сутностей (NER) для вилучення відповідних сутностей із введених користувачем даних. Ці

сутності можуть включати назви страв, інгредієнти, алергени, кількість або іншу контекстну інформацію, важливу для розуміння і точної відповіді на запит;

г) аналіз настроїв: щоб надавати більш персоналізовані та емпатичні відповіді, чат-бот може також аналізувати настрої, виражені в повідомленні користувача. Виявляючи такі емоції, як розчарування, задоволення або розгубленість, чат-бот може відповідно змінити тон і мову, сприяючи більш природному та цікавому спілкуванню.

NLP -компонент чат-бота навчається на великих масивах даних українською мовою, включаючи корпуси, бази знань і специфічні ресурси, пов'язані зі сферою обслуговування та кулінарією. Цей процес навчання включає такі методи, як навчання з перенесенням, точне налаштування та доповнення даних для подолання викликів, пов'язаних з обмеженою доступністю високоякісних україномовних даних.

2) Генерування відповідей: після того, як наміри користувача та відповідні сутності були визначені за допомогою NLP, чат-бот генерує відповідну відповідь. У цьому процесі використовуються великі мовні моделі (ВММ), спеціально розроблені та навчені для українського мовного та культурного контексту:

а) архітектура LLM : чат-бот може використовувати LLM на основі трансформерів , таких як GPT (Generative Pre-trained Transformer - генеративний попередньо навчений трансформатор) або PaLM (Pathways Language Model). Ці моделі попередньо навчаються на великій кількості українських текстових даних, що дозволяє їм вловлювати нюанси та складнощі мови, а також включати специфічні знання, пов'язані зі сферою послуг.

б) оперативний інжиніринг: щоб генерувати релевантні та узгоджені відповіді, чат-бот використовує методи швидкого інжинірингу. Це передбачає створення ретельно продуманих підказок, які надають LLM необхідний контекст, включаючи наміри користувача, витягнуті сутності та будь-яку додаткову інформацію з бази даних або зовнішніх джерел;

в) умовна генерація: на основі наданої підказки LLM генерує контекстно-релевантну відповідь, адаптовану до запиту користувача. Ця відповідь може

включати інформацію про страви, рекомендації, підтвердження замовлення або будь-яку іншу релевантну інформацію, необхідну для ефективного виконання наміру користувача;

г) доопрацювання відповіді: сформована відповідь може пройти додаткові етапи уточнення, щоб забезпечити її якість, узгодженість і відповідність заздалегідь визначеним обмеженням або інструкціям. Це може включати такі методи, як фільтрація невідповідного або нерелевантного контенту, дотримання стилістичних або граматичних правил, а також включення додаткової контекстної інформації з бази даних або зовнішніх джерел.

LLM-компонент чат-бота постійно оновлюється і допрацьовується на основі нових даних, щоб підвищити його продуктивність і адаптуватися до нових мовних патернів, уподобань клієнтів і знань у конкретній галузі.

3) Інтеграція з базою даних: для надання точної та актуальної інформації про меню ресторану, страви, вподобання клієнтів та історію замовлень чат-бот інтегрується з системою управління базами даних (СУБД). Ця інтеграція дозволяє чат-боту отримувати та зберігати релевантні дані в міру необхідності під час розмови:

а) схема бази даних: схема бази даних призначена для зберігання інформації про страви, інгредієнти, алергени, профілі клієнтів, замовлення та розмови. Ця схема забезпечує ефективне зберігання, пошук та управління даними, представлена на рис.3.1;

б) рівень доступу до даних: додаток чат-бота включає рівень доступу до даних, який абстрагується від зв'язку з базою даних. Цей рівень виконує такі завдання, як виконання запитів, отримання даних і оновлення записів, забезпечуючи послідовний і безпечний інтерфейс для взаємодії з базою даних.

в) виконання запитів: під час розмови чат-бот генерує та виконує запити до бази даних на основі запитів користувача та інформації, необхідної для формулювання відповіді. Наприклад, якщо користувач запитує про певну страву, чат-бот зробить запит до бази даних, щоб отримати інформацію про неї, інгредієнти та алергени;

г) персистентність даних: коли користувач робить замовлення або залишає відгук, чат-бот зберігає відповідну інформацію в базі даних. Сюди входять деталі замовлення, вподобання клієнтів, рейтинги та будь-які інші дані, необхідні для подальшого використання або аналізу.

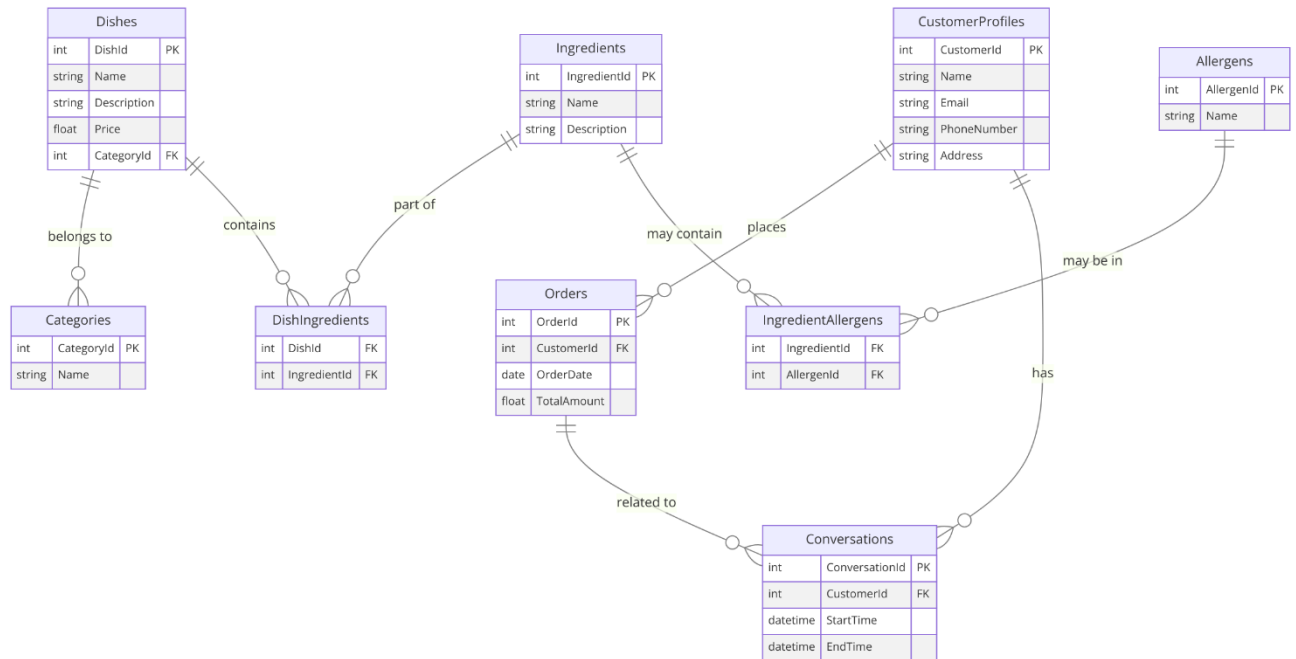


Рисунок 3.1 – Логічна схема бази даних

Компонент інтеграції з базою даних гарантує, що чат-бот має доступ до точної та актуальної інформації, що дозволяє йому надавати персоналізовані та обґрунтовані відповіді клієнтам.

1) Інтеграція зовнішніх даних: щоб розширити знання та можливості чат-бота, він інтегрується з різними зовнішніми джерелами (меню ресторану) даних та API. Ця інтеграція дозволяє чат-боту отримати доступ до додаткової інформації та контексту, що виходить за межі внутрішніх даних ресторану;

а) онлайн-бази даних про їжу: чат-бот може інтегруватися з онлайн-базами даних продуктів харчування та кулінарними ресурсами, щоб отримати детальну інформацію про інгредієнти, поживну цінність і профілі алергенів. Ця інформація може бути використана для надання більш вичерпних відповідей клієнтам з особливими дієтичними вимогами або проблемами зі здоров'ям;

б) місцеві календарі та каталоги подій: інтегруючись з місцевими календарями та каталогами подій, чат-бот може бути в курсі майбутніх фестивалів, культурних заходів або свят в українському контексті. Ці знання можна використовувати, щоб пропонувати відповідні страви, напої або акції, пов'язані з цими подіями, покращуючи клієнтський досвід і підвищуючи культурну релевантність;

в) погодні та дорожні API: інтеграція з погодними та транспортними API може дозволити чат-боту надавати контекстно-релевантну інформацію та рекомендації. Наприклад, у спекотний літній день чат-бот може запропонувати освіжаючі салати або холодні напої, які підходять для обіду на свіжому повітрі. Аналогічно, він може враховувати умови дорожнього руху, оцінюючи час доставки або рекомендуючи варіанти самовивозу;

г) соціальні мережі та платформи відгуків: відстежуючи соціальні мережі та платформи відгуків, чат-бот може збирати створений користувачами контент, огляди та відгуки про ресторан і його пропозиції. Ця інформація може бути використана для аналізу настроїв, моніторингу тенденцій і поліпшення загального клієнтського досвіду на основі зворотного зв'язку в реальному часі;

Інтеграція із зовнішніми джерелами даних розширює базу знань чат-бота, дозволяючи йому надавати більш комплексні, контекстні та релевантні відповіді клієнтам в українському секторі послуг.

2) Інтерфейс користувача та канали взаємодії: чат-бот призначений для взаємодії з клієнтами через різні канали, забезпечуючи безперебійний та зручний користувацький досвід. Ці канали можуть включати:

а) веб-інтерфейс: чат-бот може бути інтегрований на веб-сайт ресторану, що дозволить клієнтам ініціювати розмови та взаємодіяти з ботом безпосередньо зі свого веб-браузера;

б) мобільні додатки: можна розробити спеціальні мобільні додатки для забезпечення нативного спілкування з чат-ботом на платформах iOS та Android. Ці додатки можуть використовувати такі функції, як push-повідомлення, служби

визначення місцезнаходження та інтегровані варіанти оплати для більш захоплюючого та зручного користувацького досвіду;

в) платформи для обміну повідомленнями: Чат-бот може бути інтегрований з популярними платформами обміну повідомленнями, такими як Telegram, Viber або WhatsApp, використовуючи знайомість і зручність цих широко використовуваних в Україні каналів зв'язку;

г) голосові інтерфейси: щоб задовольнити клієнтів, які віддають перевагу голосовій взаємодії, чат-бот може бути інтегрований з голосовими помічниками або віртуальними агентами, що дозволяє вести розмову за допомогою голосових команд і відповідей.

Незалежно від каналу взаємодії, користувацький інтерфейс чат-бота розроблений таким чином, щоб бути інтуїтивно зрозумілим, зручним та оптимізованим для української мови та культурного контексту. Це включає використання відповідної мови, термінології та візуальних елементів, які резонують з українськими клієнтами.

3) Постійне навчання та вдосконалення: щоб чат-бот залишався актуальним та ефективним, він включає в себе механізми постійного навчання та вдосконалення. Цей процес включає в себе:

а) аналіз відгуків користувачів: чат-бот збирає та аналізує відгуки, оцінки та рецензії користувачів, щоб визначити сфери для вдосконалення, такі як прогалини в його знаннях, непорозуміння або сфери, де його відповіді були неадекватними або недоречними;

б) журналювання та аналіз розмов: Усі розмови з користувачами записуються та аналізуються, щоб виявити закономірності, проблеми, що повторюються, або можливості для покращення можливостей чат-бота. Цей аналіз може виявити нові наміри, сутності або розмовні потоки, які потребують вирішення;

в) доопрацювання та перенавчання моделі: на основі інформації, отриманої з відгуків користувачів і аналізу розмов, базові моделі NLP і LLM чат-бота допрацьовуються або перенавчаються на нових даних, щоб поліпшити їхню

продуктивність. Цей процес включає в себе такі методи, як трансферне навчання, доповнення даних і безперервна оптимізація моделі;

г) оновлення бази знань: база знань чат-бота, яка включає інформацію про страви, інгредієнти, алергени та культурний контекст, регулярно оновлюється та розширюється, щоб забезпечити точність та актуальність інформації, що надається клієнтам;

д) співпраця з експертами домену: розробка та вдосконалення чат-бота передбачає співпрацю з експертами, такими як кулінари, дієтологи та культурологи. Їхні думки та відгуки допомагають вдосконалити знання, відповіді та загальну ефективність чат-бота в українському секторі послуг.

Завдяки постійному навчанню та адаптації чат-бот завжди реагує на мінливі потреби та вподобання українських клієнтів, забезпечуючи незмінно високу якість та цікавий діалоговий досвід.

Процес обробки та відповіді на запити користувачів у чат-боті на базі LLM передбачає складну взаємодію технік NLP , архітектури LLM, інтеграції з базами даних, зовнішніми джерелами даних та інтерфейсами користувача. Використовуючи передові технології та адаптуючи їх до українського мовного та культурного контексту, чат-бот може надавати персоналізований, точний та цікавий діалоговий досвід клієнтам у сфері послуг. Крім того, здатність чат-бота безперервно навчатися та вдосконалюватися забезпечує його довгострокову актуальність та ефективність у задоволенні мінливих потреб українських споживачів.

4 ТЕХНІЧНА РЕАЛІЗАЦІЯ

4.1 Вибір та налаштування фреймворку

Фреймворк Semantic Kernel став потужним та універсальним рішенням для розробки передових додатків для обробки природної мови (NLP), включаючи чат-ботів на основі великих мовних моделей (LLM). Фреймворк розроблено переважно для мови програмування C# [34], за допомогою якого і буде розроблено даний чат-бот. У контексті українського сектору послуг використання цього фреймворку має низку переваг, таких як сумісність з українськими мовними моделями, простота інтеграції з існуючими системами та масштабованість для пристосування до майбутнього зростання та вимог, що змінюються.

Одним із ключових чинників вибору фреймворку Semantic Kernel є його здатність безперешкодно інтегруватися з різними LLM. Цей фреймворк надає уніфікований інтерфейс для взаємодії з різними LLM, дозволяючи розробникам використовувати їхні сильні сторони та можливості без прив'язки до конкретної моделі чи постачальника. Коли з'являються нові та вдосконалені моделі української мови, фреймворк Semantic Kernel може легко інтегрувати їх, гарантуючи, що чат-бот залишатиметься на передовій у сфері розуміння та генерації мовлення.

Крім того, фреймворк Semantic Kernel пропонує модульну та розширювану архітектуру, що дозволяє розробникам налаштовувати та розширювати функціональність чат-бота відповідно до унікальних потреб українського сектору послуг. Ця модульність дозволяє інтегрувати специфічні для домену бази знань, кастомні джерела даних та спеціалізовані компоненти, пристосовані до конкретних галузей або випадків використання.

Процес налаштування фреймворку Semantic Kernel складається з кількох ключових кроків:

- встановлення та конфігурація: першим кроком є встановлення пакета Semantic Kernel та його залежностей. Це можна зробити за допомогою менеджера пакетів Visual Studio, через термінал, або шляхом клонування офіційного

репозиторію з проекту Semantic Kernel. Після встановлення фреймворк потрібно налаштувати для роботи з потрібним LLM та будь-якими додатковими компонентами чи інтеграціями, необхідними для програми чат-бота;

– інтеграція мовної моделі: Фреймворк Semantic Kernel підтримує широкий спектр LLM. Щоб інтегрувати LLM, розробникам потрібно налаштувати фреймворк для завантаження відповідної моделі та встановити необхідні параметри та налаштування. Це може включати такі завдання, як завантаження попередньо навчених ваг моделі, налаштування відповідних токенизаторів та конфігурування моделі для конкретних завдань, таких як генерація тексту або відповіді на запитання;

– налаштування бази знань: однією з ключових переваг фреймворку Semantic Kernel є його здатність включати специфічні для конкретної галузі бази знань. Для українського чат-бота для сфери послуг розробники можуть створити або інтегрувати існуючі бази знань, що містять інформацію про місцеву кухню, культурні традиції та практики сфери послуг. Ці бази знань можна завантажити у фреймворк і використовувати для покращення розуміння чат-бота та його здатності генерувати відповіді;

– кастомізація та розширення: Фреймворк Semantic Kernel надає багатий набір API та точок розширення, що дозволяє розробникам налаштовувати та розширювати функціональність чат-бота за потреби. Це може включати створення кастомних компонентів для вирішення конкретних завдань, таких як розуміння природної мови (NLU), управління діалогами або розпізнавання намірів користувача. Крім того, розробники можуть інтегрувати зовнішні джерела даних, API та сервіси, щоб розширити можливості чат-бота та надати більш повні та персоналізовані відповіді;

– розгортання та масштабованість: після встановлення та налаштування фреймворку Semantic Kernel додаток чат-бота можна розгорнути у відповідному хостинговому середовищі, наприклад, на хмарній платформі або виділеному сервері. Фреймворк розроблений з можливістю масштабування, що дозволяє чат-

боту справлятися зі зростаючим користувацьким навантаженням і адаптуватися до мінливого попиту без шкоди для продуктивності або швидкості реагування.

У процесі налаштування та конфігурації важливо враховувати унікальні культурні та мовні нюанси українського контексту. Це може включати такі завдання, як включення специфічних для України мовних ресурсів, адаптація відповідей чат-бота до місцевих стилів спілкування та ідіом, а також забезпечення дотримання відповідних правил захисту даних та етичних принципів.

Однією з ключових переваг фреймворку Semantic Kernel є його активна спільнота розробників та обширна документація. Українські розробники та дослідники можуть використовувати ці ресурси, щоб бути в курсі останніх досягнень та найкращих практик, гарантуючи, що чат-боти залишатимуться передовими та ефективними в обслуговуванні потреб українського сектору послуг.

Прийнявши фреймворк Semantic Kernel, українські компанії та організації можуть скористатися надійною та гнучкою платформою для розробки чат-ботів на базі LLM, адаптованих до місцевої мови, культури та специфічних галузевих вимог. Модульність, масштабованість та безперешкодна інтеграція з українськими мовними моделями роблять фреймворк ідеальним вибором для створення інтелектуальних та цікавих діалогових інтерфейсів, які покращують клієнтський досвід та стимулюють інновації у сфері послуг.

4.2 Реалізація збереження контексту та підказок

4.2.1 Методи зберігання діалогів у базах даних

Ефективне зберігання та пошук діалогів мають вирішальне значення для збереження контексту та надання узгоджених відповідей у взаємодії з чат-ботами. У контексті українського сектору послуг, де персоналізовані та культурно релевантні діалоги є важливими, впровадження надійних механізмів зберігання даних [31] стає критично важливим компонентом системи чат-ботів. У цьому

підрозділі розглядаються різні методи зберігання діалогів у базах даних з урахуванням унікальних вимог та обмежень українського ринку.

Одним із головних міркувань при розробці схеми зберігання діалогів є здатність ефективно фіксувати та зберігати контекст розмови. Цей контекст включає такі елементи, як наміри користувача, настрої, попередні повідомлення та будь-які релевантні метадані, які можуть вплинути на відповіді чат-бота. Зберігаючи цю інформацію, чат-бот може підтримувати цілісне розуміння потоку розмови і надавати більш релевантні та персоналізовані відповіді.

Реляційні бази даних [33], такі як MySQL або PostgreSQL, пропонують структурований і добре налагоджений підхід до зберігання діалогів. Ці бази даних використовують таблиці із заздалегідь визначеними схемами, що дозволяє ефективно зберігати і знаходити структуровані дані [32]. У контексті зберігання діалогів типова схема може включати таблиці для зберігання інформації про користувача, сеансів розмови, повідомлень і будь-яких пов'язаних з ними метаданих.

Наведемо код для створення таблиць зберігання діалогів у базі даних SQL:

```
CREATE TABLE users (  
    user_id SERIAL PRIMARY KEY,  
    name VARCHAR(100),  
    email VARCHAR(100),  
    preferences JSON  
);  
  
CREATE TABLE conversations (  
    conversation_id SERIAL PRIMARY KEY,  
    user_id INT REFERENCES users(user_id),  
    start_time TIMESTAMP,  
    end_time TIMESTAMP  
);  
  
CREATE TABLE messages (  
    message_id SERIAL PRIMARY KEY,  
    conversation_id INT REFERENCES  
conversations(conversation_id),  
    sender VARCHAR(10),  
    message_text TEXT,  
    timestamp TIMESTAMP  
);
```

```
CREATE TABLE metadata (  
    metadata_id SERIAL PRIMARY KEY,  
    conversation_id INT REFERENCES  
conversations(conversation_id),  
    key VARCHAR(100),  
    value TEXT  
);
```

У цьому коді створюються таблиці users, conversations, messages та metadata. Таблиця users зберігає інформацію про користувача та його налаштування, таблиця conversations відстежує сеанси розмов, таблиця messages зберігає окремі повідомлення, якими обмінюються під час розмови, а таблиця метаданих дозволяє зберігати додаткову контекстно-залежну інформацію.

Бази даних SQL мають кілька переваг для зберігання діалогів, зокрема:

- цілісність даних: реляційні бази даних забезпечують цілісність даних за допомогою обмежень, зовнішніх ключів і транзакцій, гарантуючи, що збережені дані залишаються послідовними і надійними;

- оптимізація запитів: бази даних SQL надають потужні можливості запитів, що дозволяє ефективно знаходити та фільтрувати дані діалогу на основі різних критеріїв;

- масштабованість: багато баз даних SQL підтримують методи шардеризації та реплікації, що дозволяє здійснювати горизонтальне масштабування для обробки великих обсягів діалогових даних.

Однак бази даних SQL можуть зіткнутися з проблемами при роботі з високодинамічними або неструктурованими даними [32], такими як складні вкладені об'єкти або схеми, що швидко розвиваються. У таких випадках бази даних NoSQL можуть запропонувати більшу гнучкість і масштабованість.

NoSQL бази даних [33], такі як MongoDB або Couchbase, забезпечують документно-орієнтований підхід до зберігання даних, який може особливо добре підходити для зберігання діалогів та пов'язаного з ними контексту. Ці бази даних зберігають дані у вигляді гнучких, напівструктурованих документів, що дозволяє легко еволюціонувати та адаптуватися до мінливих вимог [32].

Приклад структури документа для зберігання діалогів у базі даних NoSQL (MongoDB) наведено в лістингу 4.1.

Лістинг 4.1 – Структура документа для зберігання діалогів у базі даних NoSQL

```
{
  «_id»: ObjectId(«63a8d7b9f3d3d8b8cd47ef35»),
  «user»: {
    «id»: «u1»,
    «name»: «Андрій Петренко»,
    «preferences»: {
      «dietary»: [«Vegan»],
      «language»: «uk»
    }
  },
  «conversation»: {
    «id»: «c1»,
    «startTime»: ISODate(«2023-06-25T14:30:00Z»),
    «endTime»: ISODate(«2023-06-25T15:15:00Z»)
  },
  «messages»: [
    {
      «sender»: «user»,
      «text»: «Добрий день! Я хочу замовити щось поїсти.»,
      «timestamp»: ISODate(«2023-06-25T14:30:00Z»)
    },
    {
      «sender»: «bot»,
      «text»: «Добрий день! Що б ви хотіли замовити?»,
      «timestamp»: ISODate(«2023-06-25T14:31:00Z»)
    },
    ...
  ],
  «metadata»: {
    «intent»: «order_food»,
    «sentiment»: «positive»
  }
}
```

У цьому прикладі кожна розмова зберігається як документ у колекції, що містить інформацію про користувача, деталі розмови, повідомлення, якими він обмінювався, і будь-які відповідні метадані. Така гнучка структура дозволяє легко еволюціонувати та адаптуватися, оскільки нові поля або вкладені об'єкти можна додавати без зміни загальної схеми.

Бази даних NoSQL мають кілька переваг для зберігання діалогів:

- гнучкість: безсхемний характер баз даних NoSQL дозволяє легко адаптуватися до мінливих вимог і структур даних, що розвиваються;
- масштабованість: NoSQL бази даних призначені для горизонтального масштабування, що робить їх придатними для обробки великих обсягів діалогових даних і високого рівня паралелізму;
- продуктивність: NoSQL-бази даних часто забезпечують швидші операції читання та запису, що може бути корисним для взаємодії з чат-ботами в режимі реального часу.

Однак у базах даних NoSQL можуть бути відсутні деякі з розширених можливостей запитів і функцій цілісності даних, які можна знайти в базах даних SQL, що вимагає додаткових зусиль для забезпечення узгодженості та надійності даних.

У деяких випадках для зберігання діалогів може бути корисним гібридний підхід [33], що поєднує бази даних SQL та NoSQL. Це може передбачати використання бази даних SQL для зберігання структурованих даних, таких як профілі користувачів і метадані розмов, а також використання бази даних NoSQL для зберігання власне діалогових повідомлень і пов'язаного з ними контексту.

Застосовуючи гібридний підхід, розробники можуть скористатися перевагами обох типів баз даних, балансує між необхідністю зберігання структурованих даних і гнучкістю та масштабованістю, необхідними для обробки динамічних даних діалогу.

Незалежно від обраного підходу до бази даних, важливо враховувати такі фактори, як конфіденційність даних, безпека та відповідність відповідним нормам в українському контексті. Це може передбачати впровадження шифрування, контролю доступу та безпечного зберігання даних для захисту конфіденційної інформації користувачів та забезпечення дотримання законів про захист даних, таких як Загальний регламент про захист даних (GDPR) та Закон України «Про захист персональних даних».

Крім того, для забезпечення ефективного зберігання та пошуку даних діалогів можуть знадобитися методи оптимізації продуктивності, такі як

індексування, кешування та балансування навантаження, особливо у сценаріях з високим рівнем паралелізму або при роботі з великими об'ємами даних.

Завдяки впровадженню надійних механізмів зберігання діалогів, пристосованих до конкретних потреб та обмежень українського сектору послуг, чат-бот може підтримувати цілісне розуміння контексту розмови, що дозволяє йому забезпечувати більш персоналізовану, культурно релевантну та цікаву взаємодію з клієнтами. В чат-боті, описаному в даній роботі використовується реляційний підхід до зберігання даних, в якості СУБД обрано PostgreSQL.

4.2.2 Методи збереження контексту розмови.

Підтримання зв'язного та контекстуального потоку розмови має вирішальне значення для забезпечення безперебійного та персоналізованого користувацького досвіду у взаємодії з чат-ботами. В українському секторі послуг, де значну роль відіграють культурні нюанси та місцеві вподобання, збереження контексту розмови стає ще більш важливим. У цьому підрозділі розглядаються різні методи ефективного управління та збереження контексту в системі чат-ботів, які гарантують, що бот може надавати релевантні та цікаві відповіді на основі історії розмови та інформації про користувача.

Однією з основних технік збереження контексту розмови є впровадження управління сесіями та використання контекстних об'єктів. Сесія - це окремий екземпляр розмови між користувачем і чат-ботом, який фіксує всю релевантну інформацію та стан, пов'язані з цією взаємодією.

У контексті українського чат-бота для сфери послуг об'єкт сесії може містити наступну інформацію:

– профіль користувача: сюди входять такі дані, як ім'я користувача, контактна інформація, мовні уподобання, дієтичні обмеження та будь-які інші відповідні персональні дані. Зберігаючи та отримуючи доступ до цієї інформації, чат-бот може персоналізувати свої відповіді та рекомендації на основі індивідуальних потреб та уподобань користувача.

– історія розмов: хронологічний запис повідомлень, якими обмінювалися користувач і чат-бот протягом поточного сеансу. Ця історія дозволяє чат-боту розуміти контекст розмови, повертатися до попередніх висловлювань і надавати послідовні та релевантні відповіді.

– відстеження намірів і настроїв: чат-бот може використовувати методи обробки природної мови для виявлення та зберігання намірів користувача (наприклад, замовлення їжі, бронювання, запит інформації) та настроїв (наприклад, позитивних, негативних, нейтральних) протягом розмови. Ця інформація допомагає чат-боту відповідно адаптувати свою мову і тон, сприяючи більш природній і цікавій взаємодії.

– контекстуальні змінні: залежно від потоку розмови, чат-боту може знадобитися зберігати та отримувати певні контекстні змінні або прапори, щоб підтримувати відповідний контекст. Наприклад, якщо користувач робить замовлення, чат-бот може зберігати змінні, пов'язані з обраними стравами, кількістю та будь-якими спеціальними інструкціями.

Зберігаючи ці об'єкти сеансу, чат-бот може ефективно керувати станом розмови та надавати послідовні та релевантні відповіді, виходячи з поточного контексту. Крім того, управління сеансами дозволяє реалізувати такі функції, як призупинення та відновлення розмови, що дає змогу користувачам безперешкодно продовжувати взаємодію між кількома сеансами або пристроями. Програмний код реалізації чат-контролера наведено в лістингу 4.2.

Лістинг 4.2 – Програмний код реалізації чат-контролера

```
using Microsoft.AspNetCore.Http;  
using Microsoft.AspNetCore.Mvc;  
using RestoLm.Application.Services;  
  
namespace RestoLm.Controllers.Chat;  
  
[ApiController]  
[Route("api/[controller]")]  
public class ChatController : ControllerBase  
{  
    private readonly IConversationService _conversationService;  
  
    public ChatController(IConversationService conversationService)
```

```

    {
        _conversationService = conversationService;
    }

    [HttpPost]
    public IActionResult Chat([FromForm] string message)
    {
        var context =
HttpContext.Session.GetObject<ChatContext>("context") ?? new
ChatContext
    {
        UserProfile = new Dictionary<string, string>(),
        ConversationHistory = new List<string>(),
        Intent = null,
        Sentiment = null,
        Variables = new Dictionary<string, string>()
    };

    // Update the conversation history
    context.ConversationHistory.Add(message);

    // Process the message and update the context
    var processedMessage =
_conversationService.ProcessMessage(message, context);
    context.Intent = processedMessage.intent;
    context.Sentiment = processedMessage.sentiment;
    foreach (var variable in processedMessage.variables)
    {
        context.Variables[variable.Key] = variable.Value;
    }

    // Store updated context in session
    HttpContext.Session.SetObject("context", context);

    // Generate and return the bot's response
    var botResponse =
_conversationService.GenerateResponse(response, context);
    context.ConversationHistory.Add(botResponse);

    return Ok(botResponse);
}
}

```

У цьому прикладі об'єкт session використовується для зберігання та отримання контексту розмови. Функція ProcessMessage, задекларована в ConversationService сервісі, аналізує повідомлення користувача і відповідно оновлює контекст, а функція GenerateResponse, що також задекларована в ConversationService сервісі, генерує відповідь бота на основі оновленого контексту

[30]. В свою чергу сервіс ConversationService може робити виклики на окреме API, розроблене на Python задля більш простої і гнучкої роботи з моделями. Це обумовлено більшою кількістю спеціалізованих бібліотек саме для мови програмування Python.

Хоча управління сеансами та контекстні об'єкти забезпечують структурований спосіб зберігання та отримання контексту розмови, великі мовні моделі (LLM) також можуть бути використані для підтримки контексту в неявному вигляді через їхні внутрішні уявлення та механізми уваги.

Такі моделі, як GPT-3, Gemini та інші, навчаються на великих обсягах текстових даних, що дозволяє їм розвивати розуміння мовних моделей, семантичних зв'язків і контекстних підказок. Включаючи історію розмови у вхідний запит для LLM, модель може генерувати відповіді, які враховують попередні обміни, зберігаючи зв'язність і релевантність.

Наприклад, у даному чат-боті для сфери послуг, основний промпт (запит) для LLM може бути структурований так, як це наведено в лістингу 4.3.

Лістинг 4.3 – Приклад запиту для LLM

```
You are an expert restaurant assistant AI designed to interact with customers, understand their preferences, and provide personalized recommendations. Your tasks include:
```

1. **Recommending Dishes**: Based on the customer's preferences, recommend suitable dishes from the menu.
2. **Describing Dishes**: Provide detailed descriptions of the recommended dishes if the customer asks for more information.
3. **Answering Questions**: Address any questions the customer may have about the menu, ingredients, preparation methods, or dietary concerns.

```
Instructions:
```

- Think step by step.
- Always maintain a polite and friendly tone.
- Provide clear and concise information.
- Be proactive in offering assistance and additional information.
- Answer like real waiter, in human-like manner, no need to greet customer in every message

```
[PROVIDED DATA]
```

```
conversation history:
```

```
{{ $conversationHistory }}
-----
customer preferences and dietary restrictions:
{{ $preferencesAndRestrictions }}
-----
Recommendations from system:
{{ $systemRecommendations }}
[END OF PROVIDED DATA]

Your Answer:
```

У цьому прикладі запит на введення включає інформацію про поточні бажання клієнта і його дієтичні обмеження, інформацію від системи рекомендацій (що також за допомогою запиту до LLM аналізує патерни клієнта) та історію розмов до поточного моменту. Потім LLM може згенерувати відповідь, яка враховує дієтичні вподобання користувача, мову [21] та контекст розмови.

Цей підхід використовує здатність LLM розуміти і генерувати природну мову, неявно зберігаючи контекст через внутрішні репрезентації та механізми уваги. Однак важливо зазначити, що хоча LLM може певною мірою керувати контекстом, для забезпечення послідовної та надійної роботи їм все одно можуть бути корисними явні методи управління контекстом, такі як управління сеансами та об'єктами контексту.

Хоча традиційні LLM можуть певною мірою фіксувати контекст, їхня здатність підтримувати довготривалий контекст і узгодженість у тривалих розмовах може бути обмеженою. Щоб вирішити цю проблему, дослідники запропонували моделі мови з розширеною пам'яттю, які включають компоненти зовнішньої пам'яті для зберігання та вилучення відповідної інформації з історії розмови.

Одним з таких підходів є використання мереж пам'яті, які поєднують нейронні мережі з компонентом довготривалої пам'яті. Мережа пам'яті може зберігати і витягувати релевантну інформацію з історії розмови, дозволяючи моделі підтримувати контекст і генерувати більш узгоджені і релевантні відповіді впродовж тривалої взаємодії.

Іншим перспективним методом є використання мовних моделей, доповнених пошуком, які включають механізм пошуку для отримання відповідної інформації із зовнішніх джерел знань, таких як бази даних або бази знань. Цей підхід може бути особливо корисним в українському секторі послуг, де чат-бот може потребувати доступу до інформації з різних джерел, таких як дані про меню, вподобання клієнтів та бази культурних знань.

Приклад реалізації на Python з використанням пошуково-доповненої мовної моделі наведено в лістингу 4.4.

Лістинг 4.4 – Реалізація коду на Python з використанням пошуково-доповненої мовної моделі

```
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM

# Load the pre-trained language model and tokenizer
tokenizer = AutoTokenizer.from_pretrained("your_pretrained_model")
model =
AutoModelForCausalLM.from_pretrained("your_pretrained_model")

# Define the retrieval function
def retrieve_relevant_info(query, context):
    # Implement your retrieval logic here
    # This could involve searching a database, knowledge base, or
    other sources
    # based on the query and conversation context
    relevant_info = ...
    return relevant_info

# Generate a response with the retrieval-augmented model
def generate_response(input_text, context):
    input_ids = tokenizer.encode(input_text, return_tensors='pt')
    outputs = model.generate(input_ids, max_length=200,
early_stopping=True, pad_token_id=tokenizer.eos_token_id)
```

```
response = tokenizer.decode(outputs[0],
skip_special_tokens=True)

# Retrieve relevant information based on the input and context
relevant_info = retrieve_relevant_info(input_text, context)

# Incorporate the retrieved information into the response
augmented_response = f"{response} {relevant_info}"

return augmented_response
```

У цьому прикладі функція `retrieve_relevant_info` отримує релевантну інформацію із зовнішніх джерел на основі введених користувачем даних і контексту розмови. Потім отримана інформація включається у відповідь, що генерується, надаючи користувачеві додатковий контекст та інформацію.

Іншим підходом до збереження контексту розмови є використання контекстних вбудовувань і механізмів уваги в архітектурі мовної моделі. Ці методи дозволяють моделі фіксувати та враховувати релевантну інформацію з історії розмови, що дає змогу надавати більш послідовні та контекстно-орієнтовані відповіді.

Контекстні вбудовування, такі як ті, що використовуються в таких моделях, як BERT і XLNet, представляють слова або лексеми на основі навколишнього контексту, а не розглядають їх як ізольовані одиниці. Таке контекстне представлення дозволяє моделі краще розуміти значення і нюанси мови на основі розмовного контексту.

Механізми уваги, з іншого боку, дозволяють моделі вибірково фокусуватися на релевантних частинах вхідної послідовності під час генерування відповіді. У контексті розмови з чат-ботом механізм уваги може допомогти моделі звернути увагу на важливу інформацію з історії розмови, таку як попередні висловлювання користувача або ключові контекстні підказки.

Поєднуючи контекстні вбудовування та механізми уваги, мовні моделі можуть краще вловлювати та використовувати контекст розмови, що призводить

до більш послідовних та релевантних відповідей. Це особливо важливо в українській сфері послуг, де культурні нюанси, місцеві ідіоми та контекстні посилання відіграють вирішальну роль в ефективній комунікації.

Приклад реалізації з використанням контекстних вбудовувань та механізмів уваги наведено в лістингу 4.5.

Лістинг 4.5 – Реалізація коду на Python з використанням контекстних вбудовувань та механізмів уваги

```
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM

# Load the pre-trained language model and tokenizer
tokenizer = AutoTokenizer.from_pretrained("your_pretrained_model")
model =
AutoModelForCausalLM.from_pretrained("your_pretrained_model")

# Generate a response with contextual embeddings and attention
def generate_response(input_text, context):
    input_ids = tokenizer.encode(input_text, return_tensors='pt')
    context_ids = tokenizer.encode(context, return_tensors='pt')

    # Combine the input and context sequences
    combined_ids = torch.cat([context_ids, input_ids], dim=1)

    # Generate the response with attention to the context
    outputs = model.generate(combined_ids, max_length=200,
early_stopping=True, pad_token_id=tokenizer.eos_token_id,
use_cache=True)
    response = tokenizer.decode(outputs[0],
skip_special_tokens=True)

    return response
```

У цьому прикладі вхідний текст і контекст розмови об'єднуються в єдину послідовність, яка потім подається в мовну модель. Модель використовує контекстні вбудовування та механізми уваги, щоб генерувати відповідь, яка враховує контекст розмови, що призводить до більш зв'язного та релевантного виводу.

Хоча розглянуті вище методи можуть бути реалізовані окремо, поєднання їх у гібридному підході може ще більше посилити здатність чат-бота підтримувати та ефективно використовувати контекст розмови. Інтегруючи управління сеансами, контекстні об'єкти, моделі з розширеною пам'яттю та контекстні вбудовування, чат-бот може скористатися сильними сторонами кожного з підходів, що призведе до більш надійного та узгодженого спілкування.

Крім того, дослідники та розробники вивчають контекстно-орієнтовані архітектури, спеціально розроблені для розмовних систем ШІ. Ці архітектури включають спеціальні компоненти для відстеження контексту, управління пам'яттю та генерації контекстно-залежних відповідей, що забезпечує більш плавне та ефективне збереження контексту під час розмови.

Одним з таких прикладів є архітектура розмовного ШІ (Conversational AI Architecture, САА), запропонована дослідниками САА - це модульна та розширювана архітектура, яка включає компоненти для розуміння природної мови, управління діалогом, відстеження контексту та генерації відповідей, які працюють у тандемі для ефективної підтримки та використання розмовного контексту.

САА включає в себе такі технології, як управління сеансами, контекстні об'єкти та мовні моделі з розширеною пам'яттю, а також нові компоненти для обробки контекстних перемикачів, управління довготривалою пам'яттю та адаптації до вподобань користувача і стилів спілкування. Поєднуючи ці елементи, САА має на меті забезпечити надійну та масштабовану основу для побудови контекстно-орієнтованих систем розмовного ШІ, адаптованих до українського ринку та культурного контексту.

Приклад високорівневої архітектури розмовного ШІ (САА) наведено на рисунку 4.1.

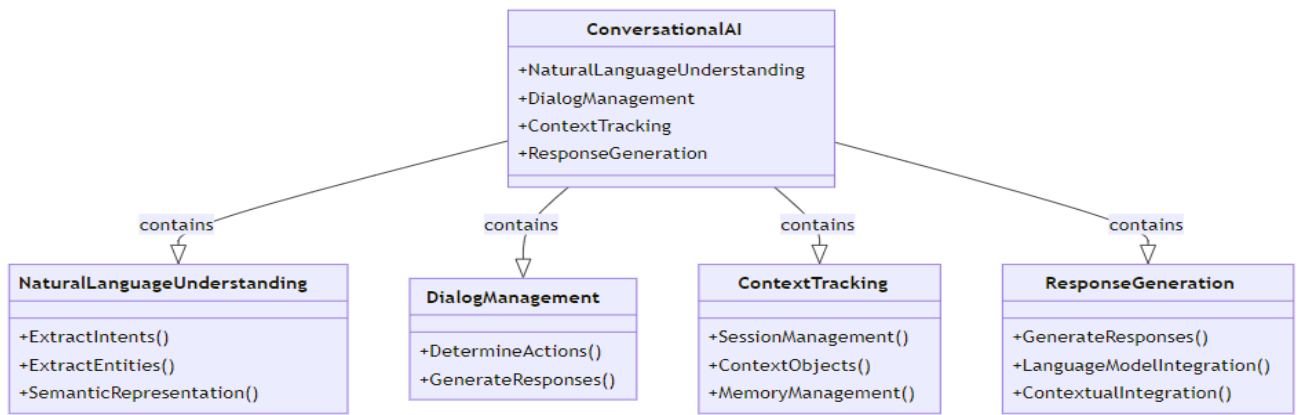


Рисунок 4.1 – Високорівнева архітектура розмовного ШІ (CAA)

У цій архітектурі компонент «Розуміння природної мови» обробляє вхідні дані користувача і витягує відповідну інформацію, таку як наміри, сутності та семантичні уявлення. Компонент управління діалогом керує потоком розмови, визначаючи відповідні дії та відповіді на основі поточного контексту.

Компонент Context Tracking відповідає за підтримку та управління контекстом розмови, включаючи управління сеансами, об'єктами контексту та управління пам'яттю. Він взаємодіє з іншими компонентами для оновлення та отримання відповідної контекстної інформації протягом розмови.

Нарешті, компонент генерації відповідей використовує мовну модель, таку як велика мовна модель або модель з розширеною пам'яттю, щоб генерувати відповідні відповіді на основі введених даних, поточного контексту і будь-якої додаткової інформації, отриманої із зовнішніх джерел або баз знань.

При розробці даного чат-боту був розроблений алгоритм включення історії розмови в запит до LLM. Цей підхід гарно себе показав, дозволяючи боту продовжувати вести діалог згідно з контекстом розмови. В ситуаціях, коли розмір діалогу перевищує допустиму кількість токенів для запиту в LLM, доцільним є скорочення історії повідомлень шляхом «прибирання» найстаріших повідомлень або за допомогою ще одного спеціального запиту до LLM, який повинен цей діалог скоротити, лишивши найважливіші деталі.

5 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

5.1 Тестування чат-бота з різними LLM

Щоб ретельно оцінити продуктивність і можливості чат-бота, було проведено комплексне тестування з використанням декількох великих мовних моделей (LLM). Зокрема, було перевірено інтеграцію та функціональність чат-бота з GPT-3.5-turbo, GPT-4 та моделлю Gemini.

Методологія тестування

Для комплексної оцінки обрано кілька репрезентативних сценаріїв взаємодії з чат-ботом, які охоплюють основні функціональні можливості, необхідні для забезпечення виняткового користувацького досвіду. Потім ці сценарії були протестовані на всіх трьох LLM, щоб полегшити порівняльний аналіз їхньої продуктивності та якості відповідей.

До процесу тестування були включені наступні сценарії:

Сценарій 1: Вітальне повідомлення. У цьому сценарії користувач ініціює розмову з чат-ботом, надсилаючи привітання або вітальне повідомлення. Таким чином перевіряється здатність чат-бота розпізнавати наміри користувача та відповідати на них відповідним привітанням або вступним повідомленням.

Сценарій 2: Запит інформації про страву. У цьому сценарії користувач запитує інформацію про конкретну страву з меню ресторану. Оцінюється здатність чат-бота знайти та надати точну інформацію про інгредієнти страви, алергени, дієтичну придатність та інші релевантні деталі.

Сценарій 3: Оформлення замовлення. У цьому сценарії користувач висловлює намір замовити їжу. Чат-бот повинен провести користувача через процес замовлення, підтвердити обрані страви та надати приблизний час приготування або вікно доставки.

Сценарій 4: Виявлення алергенів. У цьому сценарії тестується здатність чат-бота виявляти потенційні алергени у стравах на основі дієтичних обмежень або

вподобань користувача. Чат-бот повинен надавати чіткі попередження та пропонувати відповідні альтернативи, якщо це необхідно.

5.1.1 Тестування з GPT-3.5-turbo

Розпочато процес тестування, інтегрувавши модель GPT-3.5-turbo в систему розробленого чат-боту. Нижче наведено результати та спостереження, отримані під час тестування кожного сценарію з цією моделлю.

Сценарій 1: Вітальне повідомлення. Результат наведено на рис. 5.1.

```
User: Добрий день!  
Chatbot: Вітаю! Я чат-бот ресторану "Смачна Хата". Як я можу  
вам допомогти сьогодні?
```

Рисунок 5.1 – Результат тестування вітального повідомлення з моделлю
GPT-3.5-turbo

Спостереження: модель GPT-3.5-turbo успішно розпізнала привітання користувача і відповіла ввічливим і контекстуально доречним вітальним повідомленням українською мовою.

Сценарій 2: Запит інформації про страву. Результат наведено на рис. 5.2.

```
User: Розкажіть мені про борщ.  
Chatbot: Борщ – це традиційна українська страва, густий овочевий суп  
з буряком як основним інгредієнтом. До нього також зазвичай додають  
картоплю, капусту, моркву, цибулю, часник та квасолю. Борщ подається  
зі сметаною та часниковими грінками.
```

Рисунок 5.2 – Результат тестування запиту інформації про страву з моделлю
GPT-3.5-turbo

Спостереження: чат-бот надав детальний і точний опис традиційної української страви «борщ», включно з основними інгредієнтами та типовими доповненнями.

Сценарій 3: Оформлення замовлення. Результат наведено на рис. 5.3.

User: Я хотів би замовити борщ та вареники з картоплею.
Chatbot: Ваше замовлення на борщ та вареники з картоплею прийнято. Дякуємо за замовлення! Очікуваний час приготування – близько 30 хвилин.

Рисунок 5.3 – Результат тестування оформлення замовлення з моделлю GPT-3.5-turbo

Спостереження: чат-бот коректно обробив замовлення користувача на «борщ» і «вареники з картоплею» та надав орієнтовний час приготування, продемонструвавши свою здатність обробляти замовлення та надавати релевантну інформацію.

Сценарій 4: Виявлення алергенів. Результат наведено на рис. 5.4.

User: Чи містить борщ яйця?
Chatbot: Ні, традиційний борщ не містить яєць. Основними інгредієнтами є буряк, картопля, квасоля, капуста, морква, цибуля, часник та томатна паста.

Рисунок 5.4 – Результат тестування виявлення алергенів з моделлю GPT-3.5-turbo

Спостереження: чат-бот точно визначив, що «борщ» не містить яєць, продемонструвавши свою здатність виявляти потенційні алергени на основі інгредієнтів страви.

5.1.2 Тестування з GPT-4

Потім було протестовано інтеграцію чат-бота з моделлю GPT-4, використовуючи той самий набір сценаріїв. Результати та спостереження такі:

Сценарій 1: Вітальне повідомлення. Результат наведено на рис. 5.5.

User: Добрий день!

Chatbot: Вітаю вас у нашому ресторані! Я чат-бот, створений для того, щоб допомогти вам з будь-якими запитаннями чи потребами, пов'язаними з нашим меню та послугами. Як я можу вам допомогти сьогодні?

Рисунок 5.5 – Результат тестування вітального повідомлення з моделлю GPT-4

Спостереження: модель GPT-4 надала більш персоналізоване та детальне вітальне повідомлення, задаючи тон корисній та цікавій взаємодії.

Сценарій 2: Запит інформації про страву. Результат наведено на рис. 5.6.

User: Розкажіть мені про борщ.

Chatbot: Борщ – це традиційна українська страва, густий овочевий суп з буряком як основним інгредієнтом. Він готується з додаванням картоплі, капусти, моркви, цибулі, часнику та квасолі. Борщ має багату історію та різноманітні регіональні варіації в Україні. Його вважають не лише смачною, а й корисною стравою завдяки високому вмісту вітамінів та мінералів. Чи бажаєте дізнатися більше про інші страви в нашому меню?

Рисунок 5.6 – Результат тестування запиту інформації про страву з моделлю GPT-4

Спостереження: на додаток до надання детальної інформації про страву, модель GPT-4 запропонувала додатковий контекст і довідку про культурне значення «борщу» в Україні. Вона також проактивно запитувала про зацікавленість користувача дізнатися більше про інші пункти меню.

Сценарій 3: Оформлення замовлення. Результат наведено на рис. 5.7.

User: Я хотів би замовити борщ та вареники з картоплею.

Chatbot: Дякую за ваше замовлення! Ви замовили борщ та вареники з картоплею. Чи бажаєте додати до замовлення якісь напої або десерти? Також хочу повідомити, що очікуваний час приготування вашого замовлення – близько 25 хвилин. Будь ласка, повідомте мене, якщо у вас є будь-які додаткові запити чи уподобання

Рисунок 5.7 – Результат тестування оформлення замовлення з моделлю GPT-4

Спостереження: модель GPT-4 не тільки підтвердила замовлення і надала приблизний час приготування, але й проактивно запропонувала додати додаткові елементи, такі як напої або десерти. Вона також заохочувала користувача поділитися будь-якими додатковими побажаннями чи вподобаннями, демонструючи більш проактивний та персоналізований підхід до обробки замовлень.

Сценарій 4: Виявлення алергенів. Результат наведено на рис. 5.8.

User: Чи містить борщ яйця?

Chatbot: Ні, традиційний український борщ не містить яєць. Основними інгредієнтами є буряк, картопля, квасоля, капуста, морква, цибуля, часник та томатна паста. Проте, якщо у вас є алергія на якісь інші інгредієнти, будь ласка, повідомте мене, і я з радістю допоможу вам підібрати безпечну для вас страву з нашого меню.

Рисунок 5.8 – Результат тестування виявлення алергенів з моделлю GPT-4

Спостереження: як і модель GPT-3.5-турбо, GPT-4 точно визначив, що «борщ» не містить яєць. Однак він пішов далі, проактивно запропонувавши користувачеві допомогу в пошуку відповідної страви без алергенів, якщо у нього є якісь інші дієтичні обмеження або алергії.

5.1.3 Тестування з Gemini

Нарешті, було протестовано інтеграцію чат-бота з Gemini. Пропоную вашій увазі результати та спостереження.

Сценарій 1: Вітальне повідомлення. Результат наведено на рис. 5.9.

Спостереження: модель «Gemini» надала тепле і заохочувальне вітальне повідомлення, яке демонструвало її досвід в українській кухні. Вона позиціонувала себе як обізнаного гіда, готового допомогти користувачеві з рекомендаціями щодо меню та запитаннями.

User: Добрий день!

Chatbot: Вітаю вас у нашому затишному ресторані! Я – чат-бот, створений для того, щоб допомогти вам зробити ваше замовлення та відповісти на будь-які запитання, пов'язані з нашим меню та послугами. Як справжній знавець української кухні, я з радістю порекомендую вам найсмачніші страви та напої, які задовольнять навіть найвибагливіші смаки. Чим можу вам допомогти сьогодні?

Рисунок 5.9 – Результат тестування вітального повідомлення з моделлю Gemini

Сценарій 2: Запит інформації про страву. Результат наведено на рис. 5.10.

User: Розкажіть мені про борщ.

Chatbot: О, ви запитали про справжню гордість української кухні – борщ! Це густий овочевий суп з буряком як основним інгредієнтом, який готується за традиційним рецептом з додаванням картоплі, квасолі, капусти, моркви, цибулі, часнику та томатної пасту. Борщ подається зі сметаною та часниковими грінками, що надає йому неперевершеного смаку та аромату. Ця страва є невід'ємною частиною української культури та передається з покоління в покоління. Борщ не лише смачний, а й корисний завдяки великій кількості овочів та вітамінів. Чи бажаєте дізнатися більше про інші традиційні українські страви в нашому меню?

Рисунок 5.10 – Результат тестування запиту інформації про страву з моделлю Gemini

Спостереження: модель «Gemini» надала детальний і культурно багатий опис «борщу», підкресливши його значення в українській кухні та традиціях. Вона підкреслила поживні властивості страви та запропонувала поділитися інформацією про інші традиційні українські страви, продемонструвавши своє глибоке розуміння місцевого кулінарного ландшафту.

Сценарій 3: Оформлення замовлення. Результат наведено на рис. 5.11.

User: Розкажіть мені про борщ.

Chatbot: О, ви запитали про справжню гордість української кухні – борщ! Це густий овочевий суп з буряком як основним інгредієнтом, який готується за традиційним рецептом з додаванням картоплі, квасолі, капусти, моркви, цибулі, часнику та томатної пасты. Борщ подається зі сметаною та часниковими грінками, що надає йому неперевершеного смаку та аромату. Ця страва є невід'ємною частиною української культури та передається з покоління в покоління. Борщ не лише смачний, а й корисний завдяки великій кількості овочів та вітамінів. Чи бажаєте дізнатися більше про інші традиційні українські страви в нашому меню?

Рисунок 5.11 – Результат тестування оформлення замовлення з моделлю Gemini

Спостереження: модель Gemini не лише підтвердила замовлення, але й надала додаткові деталі та контекст щодо автентичності страв та способу їх приготування. Вона проактивно запропонувала додати напої або десерти, щоб доповнити їжу, продемонструвавши всебічне розуміння обіднього досвіду.

Сценарій 4: Виявлення алергенів. Результат наведено на рис. 5.12.

User: Чи містить борщ яйця?

Chatbot: Ні, традиційний український борщ не містить яєць. Основними інгредієнтами є буряк, картопля, квасоля, капуста, морква, цибуля, часник та томатна паста. Проте, якщо у вас є будь-які харчові алергії або дієтичні обмеження, будь ласка, повідомте мене про них, і я з радістю допоможу вам підібрати безпечні та смачні страви з нашого меню, які відповідатимуть вашим потребам.

Рисунок 5.12 – Результат тестування виявлення алергенів з моделлю Gemini

Спостереження: як і попередні моделі, Gemini точно визначив, що «борщ» не містить яєць. Крім того, він проактивно запропонував допомогти користувачеві знайти відповідні страви, які відповідають будь-яким дієтичним обмеженням або алергіям, які він може мати, демонструючи своє прагнення забезпечити персоналізований та інклюзивний досвід харчування.

5.2 Методи виправлення помилок та оптимізації продуктивності

Для забезпечення надійності та ефективності системи чат-ботів дуже важливо впровадити надійні механізми обробки помилок і методи оптимізації продуктивності. У контексті українського сектору послуг, де задоволеність клієнтів та операційна ефективність мають першорядне значення, ці методи відіграють життєво важливу роль у забезпеченні безперебійного та швидкого реагування на запити користувачів.

Стратегії обробки помилок:

1. Обробка винятків та ведення журналів: належна обробка винятків має важливе значення для виявлення та усунення помилок під час виконання та неочікуваних сценаріїв. Реалізувавши комплексний механізм обробки винятків, система чат-ботів може ефективно обробляти помилки, реєструвати відповідну інформацію для цілей налагодження та надавати користувачам відповідний зворотній зв'язок.

Програмний код реалізації обробки помилок на C# наведено в лістингу 5.1.

Лістинг 5.1 – Програмний код реалізації обробки помилок на C#

```
using System;
using System.IO;

namespace RestoLMChatbot
{
    public class ErrorHandlingMiddleware
    {
        private readonly RequestDelegate _next;
        private static readonly log4net.ILog _log =
log4net.LogManager.GetLogger(typeof(ErrorHandlingMiddleware));

        public ErrorHandlingMiddleware(RequestDelegate next)
        {
            _next = next;
        }

        public async Task Invoke(HttpContext context)
        {
            try
            {
                await _next(context);
            }
        }
    }
}
```

```

        }
        catch (Exception ex)
        {
            await HandleExceptionAsync(context, ex);
        }
    }

    private static async Task HandleExceptionAsync(HttpContext
context, Exception exception)
    {
        _log.Error($"An error occurred: {exception.Message}",
exception);

        context.Response.StatusCode =
StatusCodes.Status500InternalServerError;
        context.Response.ContentType = "application/json";

        await context.Response.WriteAsync(new ErrorDetails
        {
            StatusCode = context.Response.StatusCode,
            Message = "An error occurred while processing your
request."
        }.ToString());
    }
}
}
}

```

У цьому прикладі клас `ErrorHandlingMiddleware` перехоплює винятки та реєструє їх за допомогою фреймворку логування (`log4net`). Крім того, він надає клієнту стандартизовану відповідь на помилку, гарантуючи, що користувачі отримають відповідний зворотній зв'язок у разі виникнення помилки.

2. Перевірка даних: впровадження надійних механізмів перевірки даних має вирішальне значення для запобігання помилкам, спричиненим недійсними або неправильно оформленими вхідними даними користувача. Перевіряючи вхідні дані на різних етапах конвеєра обробки запитів, система чат-ботів може виявляти та вирішувати потенційні проблеми до того, як вони поширяться далі, підвищуючи загальну стабільність та надійність.

Програмний код реалізації валідації вхідних даних на `C#`, поданий у лістингу 5.2.

Лістинг 5.2 – Програмний код реалізації валідації вхідних даних

```

using FluentValidation;

namespace RestoLMChatbot.Validators
{
    public class OrderValidator : AbstractValidator<Order>
    {
        public OrderValidator()
        {
            RuleFor(order => order.CustomerName)
                .NotEmpty()
                .WithMessage("Customer name is required.");

            RuleFor(order => order.Items)
                .NotEmpty()
                .WithMessage("At least one item must be ordered.");

            RuleForEach(order => order.Items)
                .SetValidator(new OrderItemValidator());
        }
    }

    public class OrderItemValidator : AbstractValidator<OrderItem>
    {
        public OrderItemValidator()
        {
            RuleFor(item => item.DishId)
                .NotEmpty()
                .WithMessage("Dish ID is required.");

            RuleFor(item => item.Quantity)
                .GreaterThan(0)
                .WithMessage("Quantity must be greater than 0.");
        }
    }
}

```

У цьому прикладі класи `OrderValidator` і `OrderItemValidator` визначають правила валідації для об'єктів замовлення і позиції замовлення за допомогою бібліотеки `FluentValidation`. Ці правила гарантують, що обов'язкові поля присутні, а певні обмеження (наприклад, кількість більше 0) виконані перед обробкою замовлення.

3. Механізми повторної спроби: у сценаріях, де чат-бот взаємодіє із зовнішніми сервісами або API, можуть виникати тимчасові збої або проблеми з мережею. Реалізація механізмів повторних спроб може допомогти пом'якшити ці

проблеми, автоматично повторюючи невдалі операції певну кількість разів або дотримуючись певної політики повторних спроб.

Програмний код реалізації механізму повторної спроби на C# наведено в лістингу 5.3.

Лістинг 5.3 – Програмний код реалізації механізму повторної спроби

```
using Polly;
using Polly.Retry;

namespace RestoLMChatbot.Services
{
    public class MenuService
    {
        private readonly AsyncRetryPolicy _retryPolicy;

        public MenuService()
        {
            _retryPolicy = Policy
                .HandleResult<HttpResponseMessage>(r =>
!r.IsSuccessStatusCode)
                .WaitAndRetryAsync(
                    retryCount: 3,
                    sleepDurationProvider: retryAttempt =>
                    TimeSpan.FromSeconds(Math.Pow(2, retryAttempt)),
                    onRetry: (exception, sleepDuration, retryCount,
                    context) =>
                    {
                        // Log or handle retry attempt
                    });
        }

        public async Task<IEnumerable<Dish>> GetMenuAsync()
        {
            return await _retryPolicy.ExecuteAsync(async () =>
            {
                // Call external API or service to retrieve menu
                data
                var response = await
                _httpClient.GetAsync("https://api.restaurant.com/menu");
                response.EnsureSuccessStatusCode();

                return await
                response.Content.ReadAsAsync<IEnumerable<Dish>>();
            });
        }
    }
}
```

У цьому прикладі клас `MenuService` використовує бібліотеку `Polly` для реалізації політики повторних спроб. Політика визначає, що якщо HTTP-запит завершився невдало (код статусу неуспіху), його буде повторено до 3 разів з експоненціальною стратегією відступу. Це може допомогти зменшити перехідні збої та підвищити загальну надійність системи при взаємодії із зовнішніми сервісами.

Методи оптимізації продуктивності:

1. Кешування: впровадження механізмів кешування може значно підвищити продуктивність і швидкість реакції системи чат-ботів, зменшуючи необхідність багаторазового отримання або обчислення даних. Кешування може застосовуватися на різних рівнях, наприклад, кешування часто використовуваних даних меню, налаштувань користувача або навіть попередньо обчислених відповідей на поширені запити.

Код реалізації кешування меню, поданий у лістингу 5.4.

Лістинг 5.4 – Код реалізації кешування меню

```
using Microsoft.Extensions.Caching.Memory;

namespace RestoLMChatbot.Services
{
    public class MenuService
    {
        private readonly IMemoryCache _cache;

        public MenuService(IMemoryCache cache)
        {
            _cache = cache;
        }

        public async Task<IEnumerable<Dish>> GetMenuAsync()
        {
            var cacheKey = "menu";
            if (!_cache.TryGetValue(cacheKey, out IEnumerable<Dish>
cachedMenu))
            {
                cachedMenu = await FetchMenuFromDatabaseAsync();
                _cache.Set(cacheKey, cachedMenu,
                TimeSpan.FromHours(1)); // Cache for 1 hour
            }

            return cachedMenu;
        }
    }
}
```

```

        private async Task<IEnumerable<Dish>>
FetchMenuFromDatabaseAsync()
    {
        // Fetch menu data from the database
        return await _dbContext.Dishes.ToListAsync();
    }
}

```

У цьому прикладі клас `MenuService` використовує механізм кешування в пам'яті, що надається інтерфейсом `IMemoryCache`. Коли викликається метод `GetMenuAsync`, він спочатку перевіряє, чи доступні дані меню в кеші. Якщо ні, він отримує дані з бази даних і кешує їх на одну годину. Наступні запити до даних меню протягом цієї години будуть обслуговуватися з кешу, що зменшує навантаження на базу даних і покращує час відгуку.

2. Асинхронне програмування: впровадження методів асинхронного програмування може значно покращити загальну швидкість реагування та масштабованість системи чат-ботів. Використовуючи асинхронні операції, система може обробляти кілька запитів одночасно, не блокуючи потоки, що призводить до кращого використання ресурсів і підвищення пропускної здатності.

Код з використанням асинхронних операцій, що не блокують потоки було наведено в лістингу 5.5.

Лістинг 5.5 – Код реалізації використання асинхронних операцій

```

using System.Threading.Tasks;

namespace RestoLMChatbot.Services
{
    public class OrderService
    {
        public async Task<Order> PlaceOrderAsync(Order order)
        {
            // Perform order validation
            await ValidateOrderAsync(order);

            // Process order asynchronously
            var processedOrder = await ProcessOrderAsync(order);

            // Send order confirmation asynchronously
            await SendOrderConfirmationAsync(processedOrder);
        }
    }
}

```

```

        return processedOrder;
    }

    private async Task ValidateOrderAsync(Order order)
    {
        // Validate order data
    }

    private async Task<Order> ProcessOrderAsync(Order order)
    {
        // Process order and save to database
        return await _dbContext.Orders.AddAsync(order);
    }

    private async Task SendOrderConfirmationAsync(Order order)
    {
        // Send order confirmation email or notification
    }
}
}

```

У цьому прикладі клас `OrderService` реалізує метод `PlaceOrderAsync` з використанням методів асинхронного програмування. Кожен крок процесу оформлення замовлення, такий як валідація, обробка та підтвердження, реалізовано як асинхронний метод. Це дозволяє системі обробляти декілька запитів на замовлення одночасно, не блокуючи потоки, покращуючи загальну швидкість реагування та масштабованість.

3. Навантажувальне тестування та моніторинг: проведення навантажувального тестування та впровадження механізмів моніторингу мають вирішальне значення для виявлення вузьких місць у роботі та відповідної оптимізації системи чат-ботів. Навантажувальне тестування імітує реальні сценарії з різним рівнем одночасних запитів, допомагаючи виявити потенційні проблеми та сфери для вдосконалення.

Приклад навантажувального тестування за допомогою `Apache JMeter` наведено в лістингу 5.6.

Лістинг 5.6 – Код налаштування навантажувального тестування

```
# Load Test Plan
```

```

# Test the performance of the chatbot system under different load
scenarios

# Тест план
TestPlan ->
  # Thread Group
  ThreadGroup ->
    # Sampler (HTTP Request)
    HTTPSampler ->
      HTTPSamplerProxy
      Domain: "api.restaurant.com"
      Path: "/chatbot/message"
      Method: "POST"
      # Request data (e.g., user message)

# Load Profile
ThreadGroup ->
  # Ramp-up period (30 seconds)
  RampUp: 30
  # Number of threads (users)
  NumThreads: 1000
  # Loop count (unlimited)
  LoopCount: -1

# Listeners
ViewResultsFullVisualizer
SummaryReport
ResponseTimesOverTime

```

У цьому прикладі Apache JMeter використовується для створення плану навантажувального тесту для системи чат-бота. План визначає групу потоків, що представляє одночасних користувачів, з періодом наростання 30 секунд і загальною кількістю 1000 потоків (імітованих користувачів). Тест надсилає HTTP POST-запити до кінцевої точки чат-бота з імітованими повідомленнями користувачів. Різні слухачі налаштовані для фіксації та аналізу часу відгуку, пропускної здатності та інших показників продуктивності.

Проводячи навантажувальне тестування і моніторинг продуктивності системи при різних сценаріях навантаження, можна виявити потенційні вузькі місця і впровадити відповідні стратегії оптимізації. Це можуть бути такі методи, як індексування бази даних, оптимізація запитів, балансування навантаження або масштабування інфраструктури системи.

4. Профілювання та оптимізація коду: профілювання коду системи чат-ботів може допомогти виявити вузькі місця в продуктивності та сфери для оптимізації. Аналізуючи час виконання різних шляхів коду, розробники можуть виявити та усунути неефективні алгоритми, зайві обчислення або неоптимальні структури даних.

Код налаштування профайлеру за допомогою Visual Studio Profiling Tools наведено в лістингу 5.7.

Лістинг 5.7 – Код налаштування профайлеру

```
using System.Diagnostics;

namespace RestoLMChatbot.Services
{
    public class ResponseGenerationService
    {
        public string GenerateResponse(string userMessage)
        {
            // Profiling code block
            using (var profiler = new ProfilerSession())
            {
                profiler.Start();

                // Code to be profiled
                var response = ProcessUserMessage(userMessage);

                profiler.Stop();

                // Analyze profiling data
                var report = profiler.GetReport();
                // ...
            }

            return response;
        }

        private string ProcessUserMessage(string userMessage)
        {
            // Message processing logic
        }
    }
}
```

У цьому прикладі метод `GenerateResponse` у класі `ResponseGenerationService` профілюється за допомогою Visual Studio Profiling Tools. Клас `ProfilerSession`

використовується для запуску та зупинки сеансу профілювання, збору даних про продуктивність для блоку коду, що профілюється. Після профілювання зібрані дані можна проаналізувати, щоб виявити вузькі місця, такі як повільні методи, надмірне виділення пам'яті або неефективні структури даних.

Поєднуючи ці стратегії обробки помилок та методи оптимізації продуктивності, система чат-ботів може забезпечити надійний, швидкий та ефективний користувацький досвід, гарантуючи задоволеність клієнтів та операційну досконалість в українському секторі послуг.

ВИСНОВКИ

Розробка та інтеграція чат-бота на основі великої мовної моделі (LLM) для українського сектору послуг була складною, але корисною справою. У цій кваліфікаційній роботі було досліджено різні аспекти цього проекту, від теоретичних основ до етапів практичної реалізації та тестування.

В першому розділі було надано всебічний огляд поточного ландшафту технологій чат-ботів та їх застосування на українському ринку. Заглиблення в тонкощі великих мовних моделей, таких як GPT і Gemini, допомогло оцінити їхній потенціал для революції в обробці природної мови та розмовного ШІ. Крім того, було розглянуто унікальні виклики та міркування, які виникають при розробці рішень для чат-ботів, адаптованих до українського контексту, включаючи лінгвістичні складнощі, культурні нюанси та обмеження в доступі до даних.

В другому розділі було сформульовано основні вимоги до функціональності чат-бота, включаючи обробку природної мови, управління діалогами, відстеження контексту та генерацію відповідей. Було запропоновано надійний фреймворк для розробки чат-бота на рівні LLM, спеціально розробленого для українського сектору послуг. Цей фреймворк використовує передові технології, такі як семантичне ядро, та включає специфічні для України мовні моделі, бази знань та зовнішні джерела даних, що дозволяє забезпечити персоналізовану, контекстно-орієнтовану та культурно-релевантну взаємодію з клієнтами.

В третьому розділі було описано етапи реалізації, що включали ретельний відбір та налаштування фреймворку Semantic Kernel, який уможливив безперешкодну інтеграцію з різними LLM, зокрема GPT-3.5, GPT-4 та Gemini. Було досліджено методи збереження контексту розмови, такі як управління сеансами, контекстні об'єкти, моделі з розширеною пам'яттю та контекстні вбудовування. Ці методи дозволили чат-боту підтримувати зв'язну та цікаву бесіду, адаптуючись до вподобань користувача та мінливого потоку розмови.

В четвертому розділі було проведено комплексне тестування чат-бота за участю декількох LLM і репрезентативних сценаріїв взаємодії. Результати цих

тестів висвітлили сильні та слабкі сторони кожної моделі, надаючи цінну інформацію для майбутніх удосконалень та оптимізацій. Зокрема, модель GPT-4 продемонструвала чудову продуктивність з точки зору точності відповідей, швидкості обробки та загальної стабільності, що робить її перспективним вибором для інтеграції з українським чат-ботом у сфері послуг.

В п'ятому розділі приділялась значна увага стратегіям обробки помилок і методам оптимізації продуктивності. Впровадивши надійну обробку винятків, перевірку даних і механізми повторних спроб, було забезпечено надійність і відмовостійкість системи чат-боту. Крім того, досліджено такі методи, як кешування, асинхронне програмування, навантажувальне тестування та профілювання коду, щоб підвищити швидкість реагування, масштабованість та загальну ефективність системи. Протягом усього процесу розробки приділялась значна увага стратегіям обробки помилок і методам оптимізації продуктивності. Впровадивши надійну обробку винятків, перевірку даних і механізми повторних спроб, забезпечили надійність і відмовостійкість системи чат-ботів.

Важливо визнати поточні виклики та майбутні напрямки досліджень. Доступність високоякісних україномовних даних залишається обмежувальним фактором, і слід докласти зусиль для кураторства та розширення існуючих корпусів і баз знань.

Крім того, етичні міркування, такі як конфіденційність даних, безпека та зменшення упередженості, повинні бути на передньому плані майбутніх зусиль з розвитку систем взаємодії з користувачем за допомогою великих мовних моделей. Оскільки чат-боти все більше інтегруються в різні аспекти нашого життя, важливо забезпечити, щоб вони працювали в рамках прозорості, підзвітності та поваги до прав людини і суспільних цінностей.

Незважаючи на ці виклики, потенційні переваги чат-ботів, розроблених за допомогою LLM, для українського сектору послуг є величезними. Забезпечуючи персоналізоване, ефективне та культурно релевантне обслуговування клієнтів, ці чат-боти можуть стимулювати інновації, підвищувати операційну ефективність та сприяти зміцненню довіри та взаємодії між бізнесом та його клієнтами. Оскільки

український ринок продовжує розвиватися та впроваджувати цифрову трансформацію, інтеграція передових розмовних рішень зі штучним інтелектом буде ставати все більш життєво важливою для збереження конкурентних переваг та надання виняткового сервісу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Державна служба статистики України. URL: <https://www.ukrstat.gov.ua/>.
2. Худолій Ю.С., Халєвіна М.О. Фінтехнології в банківському бізнесі: реалії та перспективи. Проблеми економіки. 2021. № 1. С. 134–142.
3. Чат-боти на базі штучного інтелекту у сфері фінансів: 5 способів досягти успіху в економіці вражень і примножити прибуток. URL: <https://www.gms-worldwide.com/uk/blog/ai-chatbots-in-finance-5-ways-to-win-the-experience-economy-and-increase-profits/>
4. Що таке велика мовна модель (Large Language Model, LLM). TheTransmitted. 2023. URL: <https://thetransmitted.com/adlucem/shho-take-velyka-movna-model-large-language-model-llm/>
5. Наливайко, О. О. Перспективи використання нейромереж у вищій освіті України. Інформаційні технології і засоби навчання. 2023. 97(5). С. 1–17.
6. Слінкін О. В. Розробка чат-боту для сфери послуг на основі великої мовної моделі. 28-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті»: зб. матеріалів форуму. Т. 6. Конференція «Інформаційні інтелектуальні системи» (м. Харків, 16-18 квітня 2024р.). Харків, 2024. С. 203-205.
7. Khudolii Y., Kosolapenko V. Особливості застосування чат-ботів на основі штучного інтелекту у фінансовій сфері. ЕКОНОМІКА І РЕГІОН Науковий вісник. 2023. № 3(90). С. 97–103.
8. Прокопенко Т. О., Обойщик О. Б. Особливості використання чатботів для бізнесу у сучасних месенджер чатах. Вісник Черкаського державного технологічного університету. Серія: технічні науки. 2019. № 1. С. 11–16.
9. Кобилянська О., Єсіна М., Горбенко Ю. Порівняльний аналіз штучного інтелекту на основі існуючих чат-ботів // Комп'ютерні науки та кібербезпека. 2024. № 2. С. 26-32.
10. Знахур С. В., Знахур Л. В. Прогнозування пунктуації тексту на основі моделі BERT. Системи обробки інформації. 2020. № 1(160),. С. 94–99. URL: <https://doi.org/10.30748/soi.2020.160.12>

11. Крупа А. Технологія чат-бот як чинник комп'ютерно-посередницької комунікації цифрового суспільства. *Humanities studies*. 2022. № 12(89). С. 130–141.
12. Діалоговий потік. Документація діалогового потоку. URL: <https://cloud.google.com/dialogflow/docs>
13. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: 142 Proceedings of the 2019 Conference of the North American Chapter of the AssMinneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186.
14. Sytnik N. I. The customer experience management as a strategy for business development. *Business inform.* 2021. Т. 9, № 524. С. 216–224. URL: <https://doi.org/10.32983/2222-4459-2021-9-216-224>
15. Exploring transfer learning with T5: the text-to-text transfer transformer. Google Research - Explore Our Latest Research in Science and AI. URL: <https://research.google/blog/exploring-transfer-learning-with-t5-the-text-to-text-transfer-transformer/>
16. Хорошун О. О. Машинний переклад: історичний огляд. *Нова філологія*. 2021. № 82. С. 333–337
17. Argumentation models and their use in corpus annotation: practice, prospects, and challenges / H. Lopes Cardoso et al. *Natural language engineering*. 2023. P. 1–38.
18. Kübler N., Mestivier A., Pecman M. Teaching specialised translation through corpus linguistics: translation quality assessment and methodology evaluation and enhancement by experimental approach. *Meta*. 2019. Vol. 63, no. 3. P. 807–825. URL: <https://doi.org/10.7202/1060174ar>
19. Kurguzov A., Kompaniets L. The Ukrainian language and culture in global polyphony: current state and prospects. *Skhid*. 2017. No. 5(151). P. 78–82. URL: [https://doi.org/10.21847/1728-9343.2017.5\(151\).113284](https://doi.org/10.21847/1728-9343.2017.5(151).113284)
20. Dobryanska N., Lagodiienko V., Torishnya L. Prospects for the use of renewable energy sources in ukraine. *Ukrainian journal of applied economics*. 2020. Vol. 5, no. 2. P. 206–213.

21. Linguistic corpora technology as a didactic tool in training future translators / N.Ye. Lemish. Information technologies and learning tools. 2020. Vol. 79, no. 5. P. 242–259.
22. Tsyra O., Punchenko N., Frazе-Frazenko O. Features of construction and basic directions of development of virtual digital assistants. Cybersecurity: education, science, technique. 2020. Vol. 1, no. 9. P. 140–148
23. OpenAI (2023). GPT-4. URL: <https://www.openai.com/product/gpt-4>
24. Капраль О. Р., Велика М. Б. Роль чат-ботів в епоху діджиталізації. Вісник Херсонського національного технічного університету. 2023. № 3(82). С. 53–58.
25. ПриватБанк. (2020). Чат Приват24: Ваш банківський помічник 24/7. URL: <https://privatbank.ua/privat24-chat>
26. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. Improving Language Understanding by Generative Pre-training. 2018.
27. Троянов С. Штучний інтелект написав статтю для The Guardian. На chasi. 2020. URL: <https://nachasi.com/2020/09/10/the-guardian-ai/>
28. Spasiteleva S. O., Buriachok V. L. Perspectives for development of blockchain applications in Ukraine. Cybersecurity: education, science, technique. 2018. No. 1. P. 35–48.
29. Multi-source, Multilingual Information Extraction and Summarization. Poibeau Th. et al. Berlin: Springer-Verlag. 233 p.
30. Голуб Б.М. С#. Концепція та синтаксис. Львів: Видавничий центр ЛНУ імені Івана Франка, 2006. 136 с.
31. Пасічник В.В. Організація баз даних та знань: підручник для ВНЗ/ В.В. Пасічник, В.А. Резніченко. К.: Видавнича група ВНУ, 2006. 384с.
32. Литвин В.В. Методи та засоби інженерії даних та знань: навч. посіб. для студ. вищ. навч. закл. / В. В. Литвин // М-во освіти і науки, молоді та спорту України. Л.: Магнолія 2006, 2012. 248 с.
33. Nisa B. U. A comparison between relational databases and nosql databases. International journal of trend in scientific research and development. 2018. Volume-2,

Issue-3. P. 845–848. URL: <https://doi.org/10.31142/ijtsrd11214> (дата звернення: 23.05.2024).

34. Пахомов Б. И.. *C# для начинающих.* / Б. И. Пахомов. – Санкт-Петербург: «БХВ-Петербург», 2014. – 432 с.