

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет**  
**імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

**ДОПУЩЕНО ДО ЗАХИСТУ**  
Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.  
\_\_\_\_\_ Ю. П. Кондратенко  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**ІНТЕЛЕКТУАЛЬНИЙ ЧАТ-БОТ ДЛЯ ПІДТРИМКИ  
КЛІЄНТІВ У СФЕРІ ТУРИСТИЧНОГО БІЗНЕСУ**

Спеціальність 122 «Комп'ютерні науки»

**122 – КРБ – 402.21910101**

*Виконав студент 4-го курсу, групи 402*  
\_\_\_\_\_ *В. І. Артим*  
«19» червня 2024 р.

*Керівник: д-р техн. наук, доцент*  
\_\_\_\_\_ *О. В. Козлов*  
«19» червня 2024 р.

**Миколаїв – 2024**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет ім. Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

Рівень вищої освіти **бакалавр**

Спеціальність **122 «Комп'ютерні науки»**

*(шифр і назва)*

Галузь знань **12 «Інформаційні технології»**

*(шифр і назва)*

**ЗАТВЕРДЖУЮ**

Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.

\_\_\_\_\_ Ю. П. Кондратенко

«\_\_\_\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**

**на виконання кваліфікаційної роботи**

Видано студенту групи 402 факультету комп'ютерних наук Артиму Владиславу Ігоровичу

1. Тема кваліфікаційної роботи «Інтелектуальний чат-бот для підтримки клієнтів у сфері туристичного бізнесу».

Керівник роботи Козлов Олексій Валерійович, д-р техн. наук., доцент.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «28» грудня 2023 р. № 271

2. Строк представлення кваліфікаційної роботи студентом «19» червня 2024 р.

3. Вхідні (початкові) дані до роботи: огляд доступних аналогів інтелектуальних чат-ботів для підтримки клієнтів у туристичній сфері, технологічні вимоги до розробки чат-бота, вимоги до інтеграції з зовнішніми API вибір технологій для реалізації програмного забезпечення.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- аналіз предметної області інтелектуальних чат-ботів;
- огляд існуючих архітектур та технологій;

- постановка задачі та визначення вимог;
- проектування та реалізація чат-бота;
- тестування та відладка;
- оцінка ефективності та аналіз результатів.

5. Перелік графічного матеріалу: 7 рисунків, 1 таблиця, презентація.

6. Завдання до спеціальної частини:

- здійснити аналіз умов праці в робочому приміщенні;
- встановити необхідний рівень показників для робочого приміщення, де проводяться роботи з розробки системи;
- встановити основні принципи техніки безпеки.

7. Консультанти:

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Алексєєва А.О., доцент кафедри екології	

Керівник роботи д-р. техн. наук, доцент Козлов О. В.  
(*наук. ступінь, вчене звання, прізвище та ініціали*)

\_\_\_\_\_ (підпис)

Завдання прийнято до виконання Артим В. І.  
(*прізвище та ініціали*)

\_\_\_\_\_ (підпис)

Дата видачі завдання «14» січня 2024 р.

## КАЛЕНДАРНИЙ ПЛАН виконання кваліфікаційної роботи

Тема: Інтелектуальний чат-бот для підтримки клієнтів у сфері туристичного бізнесу

---

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників КРБ	10.11.2023	15.11.2023	Виконано
2	Отримання завдання на виконання КРБ	10.01.2024	15.01.2024	Виконано
3	Складання календарного плану роботи на весь період виконання КРБ	16.01.2024	30.01.2024	Виконано
4	Отримання завдання на переддипломну практику	15.04.2024	29.04.2024	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до КРБ	29.04.2024	11.05.2024	Виконано
6	Розробка звіту з переддипломної практики	12.05.2024	15.05.2024	Виконано
7	Виконання КРБ: Аналіз сучасного стану, огляд існуючих технологій, розробка ПЗ	13.05.2024	22.06.2024	Виконано
8	Перший попередній захист КРБ на засіданні комісії кафедри	27.05.2024	27.05.2024	Виконано
9	Доробка та остаточне оформлення КРБ	28.05.2024	09.06.2024	Виконано
10	Другий попередній захист КРБ на засіданні комісії кафедри	10.06.2024	10.06.2024	Виконано
11	Подання КРБ рецензенту	13.06.2024	13.06.2024	Виконано
11	Подання КРБ, її електронної копії та інших документів (відгуку, рецензії) до захисту	17.06.2024	21.06.2024	Виконано
12	Захист БКР перед екзаменаційною комісією (ЕК)	28.06.2024	28.06.2024	Виконано

Розробив студент Артим В. І. \_\_\_\_\_  
(прізвище, ім'я, по батькові студента) (підпис)

Керівник роботи д-р техн. наук, доцент Козлов О. В. \_\_\_\_\_  
(посада, прізвище, ім'я, по батькові) (підпис)

«29» січня 2024 р.

## **АНОТАЦІЯ**

**кваліфікаційної роботи студента групи 402 ЧНУ ім. Петра Могили**

**Владислав Ігорович Артим**

**Тема: «Інтелектуальний чат-бот для підтримки клієнтів у сфері туристичного бізнесу»**

Об'єктом роботи виступає процес обслуговування клієнтів у сфері туристичного бізнесу за допомогою інтелектуальних чат-ботів.

Предметом роботи є алгоритми обробки природної мови та машинного навчання, які лежать в основі функціонування інтелектуальних чат-ботів, а також методи їх інтеграції в туристичні сервіси для забезпечення ефективної взаємодії з клієнтами.

Метою роботи є підвищення ефективності обслуговування клієнтів у сфері туризму за рахунок розробки та впровадження інтелектуального чат-бота. Цей бот буде спрямований на задоволення інформаційних потреб клієнтів, ефективно вирішення стандартних запитань та проблем, а також надання персоналізованих рекомендацій у режимі реального часу. Це дозволить не тільки підвищити рівень задоволеності клієнтів, але й оптимізувати внутрішні ресурси компаній, знизивши загальні оперативні витрати.

Пояснювальна записка складається зі вступу, трьох розділів, висновків та додатків.

У першому розділі розглядається аналіз предметної області, основні поняття інтелектуальних чат-ботів, огляд існуючих архітектур та аналіз технологій для їх створення.

У другому розділі детально описується розробка інтелектуального чат-бота, включаючи постановку задачі, вибір архітектури та інструментів, проектування, реалізацію, тестування та відладку.

У третьому розділі наводиться оцінка ефективності розробленого чат-бота,

методи оцінки, проведення експериментів, порівняння з іншими чат-ботами та аналіз результатів.

В результаті розроблено інтелектуальний чат-бот для підтримки клієнтів у сфері туристичного бізнесу з використанням сучасних технологій штучного інтелекту, що показав високу ефективність та точність у взаємодії з користувачами.

Кваліфікаційна робота містить 53 сторінок, 7 рисунків, 1 таблицю, 25 використаних джерел та 4 додатки.

Ключові слова: інтелектуальні чат-боти, штучний інтелект, машинне навчання, підтримка клієнтів, туристичний бізнес.

## **ABSTRACT**

**qualification work of a student of group 402 of Petro Mohyla Black Sea National University**

**Artym Vladyslav**

**Title: "Intelligent chatbot for customer support in the field of tourism business"**

The object of the work is the process of customer service in the tourism industry using intelligent chatbots.

The object of the work is the process of customer service in the tourism industry using intelligent chatbots.

The aim of the work is to improve the efficiency of customer service in the tourism sector by developing and implementing an intelligent chatbot. This bot will be aimed at meeting the informational needs of customers, effectively resolving standard inquiries and issues, and providing personalized recommendations in real-time. This will not only enhance customer satisfaction but also optimize internal resources of companies, reducing overall operational costs.

The explanatory note consists of an introduction, three chapters, conclusions and appendices.

The first chapter deals with the analysis of the subject area, the main concepts of intelligent chatbots, an overview of existing architectures and an analysis of technologies for their creation.

The second chapter describes in detail the development of an intelligent chatbot, including problem statement, choice of architecture and tools, design, implementation, testing and debugging.

The third section provides an assessment of the effectiveness of the developed chatbot, evaluation methods, conducting experiments, comparison with other chatbots and analysis of the results.

As a result, an intelligent chatbot was developed for customer support in the field of tourism business using modern artificial intelligence technologies, which showed high

efficiency and accuracy in interaction with users.

The qualification work contains 53 pages, 7 figures, 1 table, 25 used sources and 4 appendices.

Keywords: intelligent chatbots, artificial intelligence, machine learning, customer support, tourism business.



## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....	3
ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Основні поняття інтелектуальних чат-ботів .....	6
1.2 Огляд існуючих архітектур інтелектуальних чат-ботів .....	9
1.3 Аналіз існуючих технологій для створення інтелектуальних чат-ботів.....	12
2 РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА .....	18
2.1 Постановка задачі та визначення вимог.....	18
2.2 Вибір архітектури та інструментів .....	20
2.3 Проектування чат-бота.....	22
2.4 Реалізація чат-бота .....	25
2.5 Тестування та відладка.....	27
3 ОЦІНКА ЕФЕКТИВНОСТІ ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА.....	32
3.1 Методи оцінки ефективності .....	32
3.2 Проведення експериментів .....	34
3.3 Порівняння з іншими чат-ботами.....	38
3.4 Аналіз результатів .....	40
ВИСНОВКИ .....	42
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	44
ДОДАТОК А Інтеграція з TripAdvisor API .....	46
ДОДАТОК Б Виділення сутностей з тексту .....	48
ДОДАТОК В Додаткове тренування моделі T5 .....	49
ДОДАТОК Г Генерація відповідей за допомогою моделі T5.....	52

## **ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ**

API – Application Programming Interface

NLP – Natural Language Processing

AI – Artificial Intelligence

NER – Named Entity Recognition

## ВСТУП

В умовах постійної конкуренції що зростає у сфері туристичних послуг, важливість інтеграції новітніх технологій стає невід'ятною частиною стратегічного розвитку бізнесу. Зокрема, сучасна туристична індустрія зіштовхується з викликами, пов'язаними з необхідністю надання швидкої, ефективної та високоякісної підтримки клієнтів. У цьому контексті інтелектуальні чат-боти, засновані на технологіях штучного інтелекту, виступають як незамінний інструмент, здатний задовольнити ці потреби, працюючи 24/7 без додаткових витрат на персонал.

Впровадження інтелектуальних чат-ботів дозволяє туристичним компаніям оптимізувати процеси обслуговування, знижувати час відповіді на запити клієнтів та значно підвищувати рівень задоволення користувачів. Особливо важливим це стає у періоди пікових навантажень, коли звернень від клієнтів стає значно більше, а відгук на них має бути миттєвим.

Крім того, з ростом глобального мобільного доступу і збільшенням кількості клієнтів, які користуються цифровими каналами для планування та бронювання подорожей, постає актуальна потреба у впровадженні більш розширених, адаптивних та персоналізованих технологічних рішень. Інтелектуальні чат-боти забезпечують не тільки оперативну підтримку, але й здатні надавати персоналізовані рекомендації, аналізуючи попередні запити та вподобання користувачів.

Об'єктом роботи виступає процес обслуговування клієнтів у сфері туристичного бізнесу за допомогою інтелектуальних чат-ботів.

Предметом роботи є алгоритми обробки природної мови та машинного навчання, які лежать в основі функціонування інтелектуальних чат-ботів, а також методи їх інтеграції в туристичні сервіси для забезпечення ефективної взаємодії з клієнтами.

Метою роботи є підвищення ефективності обслуговування клієнтів у сфері

туризму за рахунок розробки та впровадження інтелектуального чат-бота. Цей бот буде спрямований на задоволення інформаційних потреб клієнтів, ефективно вирішення стандартних запитань та проблем, а також надання персоналізованих рекомендацій у режимі реального часу. Це дозволить не тільки підвищити рівень задоволеності клієнтів, але й оптимізувати внутрішні ресурси компаній, знизивши загальні оперативні витрати.

Для досягнення мети поставлено наступні задачі:

- проаналізувати основні поняття інтелектуальних чат-ботів;
- дослідити архітектуру інтелектуального чат-бота;
- проаналізувати технології для створення інтелектуальних чат-ботів.

Таким чином, дане дослідження визначає не тільки потребу, але й актуальність розробки інтелектуальних чат-ботів у контексті забезпечення високого рівня клієнтського сервісу у туристичній індустрії.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Основні поняття інтелектуальних чат-ботів

Чат-бот — це комп'ютерна програма, створена для симуляції людської бесіди з користувачами через текстові або голосові інтерфейси [1]. Використання чат-ботів може бути різноманітним: від простих завдань, як-от відповіді на часті запитання, до складніших, наприклад, надання персоналізованих рекомендацій або автоматизації бізнес-процесів.

Розробка чат-ботів зазвичай включає використання природної мови та машинного навчання, особливо глибокого навчання, щоб вони могли розуміти запити користувачів і генерувати відповідні відповіді. Чат-боти можуть використовувати підходи, засновані на правилах, де вони відповідають на запитання відповідно до заздалегідь визначених шаблонів, або на основі штучного інтелекту, де вони адаптують свої відповіді, аналізуючи великі обсяги текстових даних [1].

Застосування чат-ботів змінює взаємодію між людьми та технологіями, забезпечуючи більшу зручність і ефективність у багатьох сферах, включаючи обслуговування клієнтів, освіту, фінанси та охорону здоров'я.

Інтелектуальні чат-боти, що використовують технології штучного інтелекту (ШІ), значно перевершують традиційні боти, що базуються на правилах, за ступенем автоматизації. Ці розширені системи об'єднують різноманітні аспекти штучного інтелекту для розуміння, обробки та генерації мови, що значно покращує якість взаємодії з користувачами [1].

Інтелектуальні чат-боти застосовують складні алгоритми NLP для аналізу і розуміння мови користувача. Це включає розбір синтаксичної структури речень, визначення ключових слів та фраз, а також інтерпретацію семантики та контексту запитів [1]. Інтелектуальні чат-боти використовують моделі, засновані на машинному та глибокому навчанні, для постійного вдосконалення своїх відповідей та поведінки

на основі взаємодії з користувачами.

Завдяки передовим алгоритмам генерації тексту, такі чат-боти можуть створювати зрозумілі, зв'язні та релевантні відповіді, які часто майже неможливо відрізнити від відповідей людини. Відмінною особливістю інтелектуальних чат-ботів є їхня здатність зберігати інформацію про попередні розмови та використовувати ці дані для врахування контексту в нових діалогах [1].

Деякі розширені системи можуть аналізувати емоційний стан користувача через аналіз тексту або голосу, адаптуючи свої відповіді, щоб бути більш емпатійними та ефективними.

Чат-боти можна класифікувати за різними критеріями, залежно від їхнього призначення, складності, способу взаємодії та платформи, на якій вони розміщені [1].

За способом взаємодії:

1) прості або скриптовані чат-боти працюють на основі заздалегідь заданих скриптів та правил. Вони взаємодіють з користувачами, використовуючи обмежений набір команд або опцій і зазвичай відповідають швидко та ефективно, але мають обмежені можливості щодо обробки неочікуваних запитань або вільної форми діалогу [1];

2) інтелектуальні або AI-підсилені чат-боти використовують складні алгоритми машинного навчання та обробки природної мови для інтерпретації запитів користувачів. Вони здатні вести більш природні розмови й можуть навчатися з досвіду, що покращує їхню здатність відповідати на запитання та реагувати на змінні умови взаємодії [1].

За призначенням:

1) інформаційні чат-боти надають користувачам спеціалізовану інформацію. Наприклад, боти, що відповідають на питання про погоду, новини, фінансові ринки тощо;

2) сервісні чат-боти виконують конкретні завдання, такі як бронювання квитків, замовлення їжі, управління замовленнями клієнтів та інші сервісні операції;

3) розважальні чат-боти створені для розваги, можуть виконувати ігри, проводити вікторини, надсилати курйозні або цікаві факти, або взаємодіяти у веселому та інтерактивному форматі;

4) психологічні або підтримуючі чат-боти призначені для психологічної підтримки, заспокоєння або мотивації користувачів. Вони можуть використовуватися для надання базових порад або як перший крок у психологічній допомозі.

За платформи:

1) веб-боти вбудовані безпосередньо в веб-сайти та дозволяють взаємодіяти з відвідувачами сайту, надаючи інформацію, допомогу в навігації по сайту або виконанні певних завдань [1];

2) месенджер-боти які інтегровані в платформи месенджерів, такі як Facebook Messenger, WhatsApp, Telegram, і т.д. Ці боти легко доступні для великої аудиторії користувачів і можуть використовуватися для різноманітних цілей від сервісу до розваг [1].

Кожна з цих категорій має свої особливості та виклики у розробці та інтеграції, і вибір відповідної категорії чат-бота значною мірою залежить від конкретних потреб бізнесу або організації.

Штучний інтелект грає важливу роль у функціонуванні чат-ботів, він допомагає чат-ботам краще розуміти нюанси людської мови, що забезпечує більш природній та ефективний обмін інформацією [1]. Штучний інтелект забезпечує чат-ботам здатність адаптуватися до нових ситуацій та запитів без постійного втручання розробників, що важливо для широкомасштабної автоматизації.

III дозволяє чат-ботам персоналізувати взаємодію, аналізуючи попередні діалоги та поведінку користувачів, що сприяє більшій задоволеності користувачів і залученості. Використання технік машинного навчання дозволяє чат-ботам

самостійно вдосконалюватися, що зменшує потребу в ручному оновленні бази знань та покращує точність відповідей з часом.

Штучний інтелект є ключовим компонентом в еволюції чат-ботів, перетворюючи їх з простих інструментів на складні системи, які можуть ефективно виконувати широкий спектр завдань і значно підвищувати якість взаємодії між людьми та машинами.

## 1.2 Огляд існуючих архітектур інтелектуальних чат-ботів

Чат-бот — це програма, що використовується для спілкування з людьми. Вона може бути налаштована на виконання різних завдань, наприклад, обслуговування клієнтів, навчання або надання інформації. Структура та компоненти чат-ботів можуть бути різноманітними, але основні елементи часто залишаються схожими [2].

*Інтерфейс користувача.* Це фронтенд частина чат-бота, через яку користувачі взаємодіють з ботом. Це може бути вебінтерфейс, інтерфейс мобільного додатка, платформа соціальних медіа (наприклад, Facebook Messenger, WhatsApp) або навіть голосовий асистент [2].

*Двигун обробки повідомлень.* Центральний компонент чат-бота, що аналізує вхідні повідомлення і визначає відповідь. Він може використовувати різні технології, включаючи [2]:

- правила: прості боти можуть використовувати заздалегідь задані правила для відповіді;
- штучний інтелект: більш складні системи використовують машинне навчання й обробку природної мови для розуміння контексту і намірів користувача.

*База знань.* Чат-боти часто використовують великі бази даних або інтегровані API для доступу до інформації, необхідної для відповідей на запитання користувачів [2]. Це може бути база FAQ, зовнішні вебсервіси, корпоративні бази даних тощо.

*Менеджер діалогів.* Цей компонент керує потоком розмови, вирішуючи, коли і



як перейти від одного кроку діалогу до іншого [2]. Він може використовувати стан діалогу для збереження інформації про користувача протягом сесії.

*Інтеграція зі сторонніми сервісами.* Для розширення функціонала бота, він може інтегруватися з іншими програмами та сервісами, такими як системи CRM, бази даних, електронна пошта, системи планування тощо [2].

*Аналітика.* Для покращення ефективності та відповідності відповідей чат-бота важливо аналізувати інтеракції з користувачами. Аналітика допомагає відстежувати поведінку користувачів, ефективність відповідей та допомагає в удосконаленні бота [2].

Адміністративна панель. Для управління чат-ботом зазвичай існує адміністративна панель, що дозволяє налаштування бота, оновлення бази знань, перегляд статистики та інтеракцій, і управління діалоговими сценаріями.

Чат-боти можуть бути дуже різноманітними за своєю складністю та функціоналом, від простих, що відповідають за допомогою шаблонних повідомлень, до високоінтелектуальних, здатних вести складні діалоги, аналізувати емоції та вести неперервне навчання на основі отриманих даних [2].

Інтелектуальні чат-боти, які використовують технології штучного інтелекту, пропонують вдосконалені функціональні можливості для ефективної взаємодії з користувачами [2]. Ці можливості дозволяють їм бути більш адаптивними, реактивними та персоналізованими у своїх відповідях.

Способи взаємодії з користувачем [2]:

- текстові інтерфейси - це найпоширеніший спосіб взаємодії з чат-ботами, де користувачі вводять текст через месенджери, вебінтерфейси або мобільні застосунки;
- голосові інтерфейси - чат-боти також можуть бути інтегровані з голосовими асистентами, як-от Amazon Alexa або Google Assistant, дозволяючи користувачам взаємодіяти з ними за допомогою голосових команд;

– омніканальна інтеграція - деякі розширені чат-боти забезпечують безперервність взаємодії між різними каналами, наприклад, перемикання між текстовим чатом у мобільному застосунку та голосовими командами через смартспікери.

Розпізнавання та генерація текстової інформації [2]:

– розпізнавання мови - чат-боти використовують NLP для аналізу та інтерпретації користувачького вводу, розпізнаючи ключові слова та фрази, щоб визначити намір користувача;

– генерація відповідей за допомогою технологій як GPT (Generative Pre-trained Transformer) чат-боти можуть генерувати зрозумілі, зв'язні та контекстуально відповідні відповіді;

– контекстуальне розуміння - здатність підтримувати контекст взаємодії, зберігаючи інформацію про попередні взаємодії, що дозволяє надавати відповіді, які враховують історію розмови.

Підтримка багатомовності та інтернаціоналізація [2]:

– інтелектуальні чат-боти можуть підтримувати велику кількість мов, дозволяючи компаніям взаємодіяти з глобальною аудиторією. Вони використовують мовні моделі, які можуть розпізнавати та генерувати відповіді на різних мовах;

– чат-боти можуть бути спеціалізовані для конкретних регіональних ринків, враховуючи культурні нюанси та особливості мови для більш ефективної взаємодії;

– понад мовну адаптацію, інтелектуальні чат-боти можуть також розпізнавати та реагувати відповідно до культурних контекстів, забезпечуючи вищий рівень особистісної адаптації.

Такі функціональні можливості дозволяють інтелектуальним чат-ботам не тільки покращувати взаємодію з користувачами, але й сприяти масштабуванню послуг і продуктів на міжнародному рівні, а також забезпечувати більш персоналізований досвід користувачького сервісу.

### 1.3 Аналіз існуючих технологій для створення інтелектуальних чат-ботів

Технології та методи, що використовуються для створення інтелектуальних чат-ботів, значною мірою базуються на досягненнях у галузі штучного інтелекту та обробки даних. Ось детальний опис трьох ключових аспектів:

*Методи машинного навчання.* Машинне навчання є основою для більшості сучасних чат-ботів, забезпечуючи можливість зрозуміти та відповідати на людські запитання з деякою ступінню інтелекту [3]. Застосування машинного навчання включає навчання моделей на великих обсягах текстових даних, щоб вони могли розпізнавати мовні шаблони, інтенти (наміри користувачів) і контекст.

1. Навчання з вчителем: використовується для тренування моделей на попередньо анотованих даних, де вхідні дані (питання) мають відповідні відповіді [3].

2. Навчання без вчителя: аналізує великі набори даних без попередньо визначених відповідей для виявлення закономірностей або структур в даних [3].

3. Підсилювальне навчання: використовується для оптимізації рішень чат-бота на основі отриманих від користувачів відгуків [3].

*Обробка природної мови (NLP).* Обробка природної мови дозволяє машинам розуміти, інтерпретувати, обробляти та генерувати людську мову в її різних формах. Основні компоненти NLP включають [4]:

- розбір речень для виявлення граматичної структури й відносин між словами;
- визначення значення слів у контексті та взаємодії між ними для зрозуміння загального змісту;
- ідентифікація ключових елементів у запитах (наприклад, імен, місць, часу) і витягування корисної інформації.

*Бази знань та семантичні моделі.* Бази знань дозволяють чат-ботам відповідати на запитання з використанням зібраних фактів і знань, структурованих у логічно взаємопов'язаній формі [4]:

1) бази знань - це структуровані сховища даних, що містять інформацію про світ, яка може бути легко запитана та використана для відповідей. Вони можуть бути спеціалізовані (наприклад, медичні бази даних) або загальні (наприклад, Вікіпедія);

2) семантичні мережі та онтології - структури, які допомагають системам розуміти взаємозв'язки між концепціями та категоріями в даних, підвищуючи здатність чат-ботів до ведення "розумної" бесіди.

Інтелектуальні чат-боти зазвичай використовують набір алгоритмів і технік машинного навчання та обробки природної мови (NLP) для розуміння, обробки та генерації відповідей.

Основні алгоритми та підходи, які використовуються в інтелектуальних чат-ботах [4]:

1) Обробка природної мови (NLP).

Цей набір технологій дозволяє чат-ботам розуміти людську мову у письмовій формі.

Основні компоненти NLP включають [4]:

- токенізація: розбиття тексту на менші частини (токени), такі як слова чи фрази;
- лематизація та стемінг: приведення слів до їх базової форми;
- розпізнавання іменованих сутностей (NER): ідентифікація та класифікація важливих елементів тексту (наприклад, імен, організацій, дат);
- розпізнавання інтентів та витягування сутностей: визначення наміру користувача та важливих деталей з тексту для формування відповідей.

2) Моделі класифікації.

Використовуються для визначення інтентів користувачів та класифікації їх запитів.

Популярні моделі включають:

- логістична регресія;

- машина опорних векторів (SVM);
- нейронні мережі.

### 3) Глибоке навчання.

Для розуміння контексту та генерації більш природних відповідей [3]:

- рекурентні нейронні мережі (RNN), зокрема LSTM (Long Short-Term Memory), для обробки послідовностей даних;
- мережі уваги та трансформери, як-от BERT (Bidirectional Encoder Representations from Transformers), що забезпечують високий рівень розуміння мови.

### 4) Моделювання діалогу.

Створення природного потоку розмови [4]:

- правила базовані системи: використовують заздалегідь задані правила для керування діалогом;
- генеративні моделі: автоматично генерують відповіді на основі контексту. GPT (Generative Pre-trained Transformer) є прикладом такої технології.

### 5) Оцінка та оптимізація.

Чат-боти постійно тестуються та оптимізуються на основі зворотного зв'язку користувачів:

- А/В тестування: порівняння різних варіантів чат-бота для визначення, який краще спрацьовує;
- Reinforcement learning (навчання з підкріпленням): методи, що дозволяють боту "навчатися" на основі взаємодій з користувачами, покращуючи свої відповіді.

Ці технології та методики дозволяють інтелектуальним чат-ботам вести більш природні та корисні діалоги, адаптуватися до потреб користувачів і вдосконалювати свої відповіді з часом.

Створення чат-ботів вимагає використання як певних мов програмування, так і спеціалізованих фреймворків або платформ, кожна з яких має свої переваги та особливості. Давайте детально розглянемо мови програмування та фреймворки, які

часто використовуються для розробки чат-ботів.

Python є однією з найпопулярніших мов програмування для розробки чат-ботів завдяки своїй простоті, зрозумілості та могутньому набору бібліотек [5]. Ця мова особливо ефективна у сферах машинного навчання та обробки природної мови завдяки таким бібліотекам, як TensorFlow, PyTorch, NLTK та spaCy.

JavaScript може використовуватися для створення інтерактивних чат-ботів, особливо для вебсайтів та мобільних додатків [6]. Node.js, серверний варіант JavaScript, дозволяє розробляти повнофункціональні бекенд-додатки для чат-ботів, що інтегровані з різними вебсервісами та базами даних.

Dialogflow (від Google) – це інтегрована платформа для розробки чат-ботів, що використовує машинне навчання для розуміння мовних запитів (NLP) [7]. Вона дозволяє створювати чат-боти, які можуть вести бесіду у натуральній мові й інтегруватися з багатьма платформами, такими як Google Assistant, Facebook Messenger і багатьма іншими.

Microsoft Bot Framework – цей фреймворк дозволяє розробникам створювати чат-боти, які легко інтегруються з Microsoft Azure і можуть взаємодіяти через різні канали, такі як Skype, Teams, Slack. Він включає інструменти для розробки, тестування та розгортання ботів, а також підтримує розширені можливості, такі як розпізнавання мовлення і переклад [8].

IBM Watson надає потужні інструменти для створення чат-ботів, які можуть аналізувати, розуміти й відповідати на людську мову з використанням передових алгоритмів штучного інтелекту. Він включає NLP, машинне навчання та інструменти для семантичного аналізу, що робить його ідеальним для складних діалогових систем у таких галузях, як банківська справа, охорона здоров'я і багато інших [9].

Telegram Bot API – це потужний інструмент, який дозволяє розробникам створювати та інтегрувати ботів у месенджер Telegram [10]. Боти у Telegram можуть виконувати широкий спектр завдань, від автоматизації відповідей до здійснення

складних інтеракцій з користувачами.

Ось детальний опис ключових можливостей та аспектів Telegram Bot API [10]:

*Основні можливості.* Боти можуть автоматично відправляти текст, зображення, відео, аудіо, стікери та інші типи медіафайлів [10]. Вони також можуть використовувати форматування тексту, таке як жирний або курсивний шрифт, для покращення читабельності повідомлень.

Telegram дозволяє ботам використовувати різні види клавіатур для інтеракції з користувачем [10]:

- звичайні клавіатури: клавіатури, які залишаються відкритими під чатом і дозволяють користувачам вибирати з представлених опцій;
- Inline клавіатури: кнопки, які можуть бути вбудовані прямо у повідомлення, дозволяючи користувачам взаємодіяти без необхідності відправляти текстові команди.

Боти можуть реагувати на повідомлення, команди, клацання по кнопках та інші події від користувачів, що робить можливим створення динамічних діалогових взаємодій [10].

Використання вебхуків дозволяє ботам отримувати оновлення в реальному часі, що значно підвищує швидкість та ефективність обробки повідомлень і команд.

Для створення нового бота в Telegram необхідно використати спеціального бота з назвою BotFather. Він дозволяє реєструвати нових ботів, налаштовувати їх імена, описи, а також отримувати токени доступу для API [10].

За допомогою отриманого токена можна програмно взаємодіяти з Telegram API для відправки й отримання повідомлень, організації клавіатур та обробки інших типів запитів[10]. Розробники можуть використовувати різні мови програмування, такі як Python, JavaScript або PHP, для взаємодії з API.

Телеграм-боти використовуються в різних сферах, включаючи забезпечення служби підтримки, організацію подій, автоматизацію рутинних завдань, надання

інформаційних послуг, і навіть управління платежами та замовленнями. Безпека та швидкість взаємодії роблять Telegram одним з найпопулярніших месенджерів для інтеграції чат-ботів.

Кожен з цих фреймворків і платформ надає різні інструменти та API для спрощення процесу розробки чат-ботів, дозволяючи розробникам сконцентруватися на створенні кращого досвіду для користувачів.



## 2 РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА

### 2.1 Постановка задачі та визначення вимог

Розробка інтелектуального чат-бота для підтримки клієнтів у сфері туристичного бізнесу є складною та багатоетапною задачею, що включає в себе низку функціональних та нефункціональних вимог. У цьому розділі ми окреслимо основні аспекти та завдання, які стоять перед розробниками.

Основна мета розробки інтелектуального чат-бота полягає в створенні системи, яка здатна ефективно підтримувати клієнтів, надаючи їм актуальну інформацію про туристичні місця, допомагаючи планувати подорожі та вирішувати різноманітні питання, пов'язані з подорожами. Це передбачає інтеграцію з кількома зовнішніми API, обробку природної мови для розуміння та генерації відповідей, виділення сутностей з тексту, а також забезпечення розширеного контексту для поліпшення точності відповідей.

Функціональні вимоги до системи включають збір та обробку запитів користувачів, інтеграцію з Google Places API та TripAdvisor API, генерацію відповідей з використанням моделі T5, виділення сутностей за допомогою моделі 'dbmdz/bert-large-cased-finetuned-conll03-english' та збереження контексту попередніх запитів. Збір та обробка даних є ключовими завданнями, які включають в себе обробку природної мови, аналіз запитів та формування відповідей на основі отриманої інформації.

Інтеграція з зовнішніми API є критично важливою частиною системи, оскільки саме за допомогою цих API чат-бот зможе отримувати актуальну інформацію про туристичні місця. Google Places API надає інформацію про місця, їх розташування, години роботи, відгуки та інші деталі, а TripAdvisor API дозволяє отримати детальну інформацію про туристичні місця, включаючи описи, адреси та інші дані. Ці API забезпечують наявність точних та актуальних даних, що необхідні для надання якісних відповідей користувачам.

Генерація відповідей є важливим аспектом роботи чат-бота. Використання моделі T5 дозволяє системі формувати відповіді на основі запитів користувачів, враховуючи контекст та зовнішню інформацію. Це забезпечує високу якість відповідей та здатність чат-бота адаптуватися до різноманітних запитів користувачів.

Виділення сутностей з тексту користувачів є необхідним для точного визначення їх запитів. Модель 'dbmdz/bert-large-cased-finetuned-conll03-english' дозволяє ефективно виділяти сутності, що значно покращує точність розуміння запитів та формування відповідей.

Розширення контексту є ще одним важливим аспектом, що забезпечує поліпшення якості відповідей. Збереження контексту попередніх запитів дозволяє чат-боту враховувати попередні взаємодії з користувачем та надавати більш релевантні відповіді.

Нефункціональні вимоги включають продуктивність, масштабованість, безпеку, надійність та модульність системи. Чат-бот повинен відповідати на запити користувачів в реальному часі з мінімальною затримкою, забезпечувати високу продуктивність навіть при великій кількості запитів одночасно, гарантувати конфіденційність даних користувачів та захист від несанкціонованого доступу. Надійність системи передбачає стабільну роботу без збоїв та помилок, забезпечуючи високу доступність для користувачів. Модульність архітектури дозволяє легко додавати нові функції та розширювати можливості системи.

Архітектура чат-бота включає кілька основних компонентів: модуль обробки запитів, модуль інтеграції з API, модуль генерації відповідей та контекстний модуль. Модуль обробки запитів відповідає за отримання та обробку запитів користувачів, виділення сутностей та контексту. Модуль інтеграції з API забезпечує взаємодію з Google Places API та TripAdvisor API для отримання актуальної інформації. Модуль генерації відповідей використовує модель T5 для формування відповідей на основі запитів, контексту та зовнішньої інформації. Контекстний модуль зберігає та оновлює

контекст запитів користувачів, що дозволяє покращити якість відповідей.

Таким чином, розробка інтелектуального чат-бота для підтримки клієнтів у сфері туристичного бізнесу включає комплексне вирішення задач з обробки природної мови, інтеграції з зовнішніми джерелами інформації та забезпечення високої продуктивності та надійності системи.

## 2.2 Вибір архітектури та інструментів

Вибір архітектури та інструментів для розробки інтелектуального чат-бота є важливим етапом, який значною мірою визначає успішність проєкту. Архітектура системи повинна бути гнучкою, масштабованою та здатною забезпечити високий рівень продуктивності та надійності. У цьому розділі ми розглянемо обрану архітектуру та інструменти, які будуть використані для створення чат-бота.

Архітектура чат-бота складається з кількох основних компонентів:

- модуль обробки запитів: цей модуль відповідає за отримання запитів від користувачів, їх попередню обробку та аналіз. Для цього використовується модель обробки природної мови (NLP) ‘dbmdz/bert-large-cased-finetuned-conll03-english’, яка дозволяє виділяти сутності з тексту та розуміти контекст запитів;
- модуль інтеграції з API: цей модуль забезпечує взаємодію з зовнішніми API, такими як Google Places API та TripAdvisor API, для отримання актуальної інформації про туристичні місця. Інтеграція з цими API дозволяє чат-боту надавати користувачам точну та актуальну інформацію;
- модуль генерації відповідей: використовує модель T5 (Text-To-Text Transfer Transformer) для формування відповідей на основі запитів користувачів та зовнішньої інформації. Ця модель є потужним інструментом для генерації тексту та дозволяє ефективно відповідати на різноманітні запити;

– контекстний модуль: зберігає контекст попередніх запитів користувачів, що дозволяє чат-боту враховувати попередні взаємодії та надавати більш релевантні відповіді.

Для реалізації цієї архітектури використовуються наступні інструменти та технології:

– Python: основна мова програмування для розробки чат-бота, яка забезпечує широкий спектр бібліотек та фреймворків для обробки природної мови, роботи з нейронними мережами та інтеграції з API;

– Hugging Face Transformers: бібліотека для роботи з моделями обробки природної мови, яка надає доступ до передових моделей, таких як T5 та BERT [11];

– PyTorch: фреймворк для розробки та тренування нейронних мереж, що забезпечує гнучкість та високу продуктивність [12];

– Flask: вебфреймворк для створення вебінтерфейсу чат-бота. Flask дозволяє легко створювати вебдодатки та API [13];

– Requests: бібліотека для виконання HTTP-запитів, яка використовується для взаємодії з Google Places API та TripAdvisor API [14];

– dotenv: бібліотека для роботи з файлами .env, яка забезпечує безпечно зберігання та використання API-ключів та інших конфіденційних даних [15].

Окрім зазначених компонентів, для забезпечення користувацького інтерфейсу був розроблений вебдодаток на основі Flask. Вебдодаток складається з простого HTML-файлу та CSS-стилів для оформлення інтерфейсу користувача. Основні функції вебдодатку:

– відображення інтерфейсу чат-бота з можливістю введення текстових запитів;

– відправка запитів користувачів до сервера через AJAX-запити та отримання відповідей від чат-бота;

– відображення діалогу між користувачем та чат-ботом в реальному часі.

Обґрунтування вибору цих інструментів базується на їх функціональності, продуктивності та сумісності з обраною архітектурою. Python є популярною мовою програмування, яка широко використовується для розробки чат-ботів завдяки своїй простоті та наявності численних бібліотек для обробки природної мови та роботи з нейронними мережами. Hugging Face Transformers та PyTorch забезпечують доступ до передових моделей та інструментів для їх тренування та використання. Requests та dotenv дозволяють ефективно працювати з API та безпечно зберігати конфіденційні дані.

Таким чином, обрана архітектура та інструменти забезпечують необхідну гнучкість, масштабованість та продуктивність для розробки інтелектуального чат-бота, здатного ефективно підтримувати клієнтів у сфері туристичного бізнесу.

### **2.3 Проектування чат-бота**

Проектування чат-бота є критично важливим етапом розробки, оскільки від його архітектури та логіки залежить ефективність та зручність використання кінцевого продукту. Процес проектування охоплює декілька важливих аспектів, серед яких можна виділити побудову блок-схеми роботи чат-бота та розробку моделі даних (рис. 2.1).

Для досягнення високої якості роботи чат-бота важливо врахувати всі етапи його взаємодії з користувачем, починаючи від отримання запиту та закінчуючи формуванням і відправкою відповіді. Кожен етап має бути ретельно спланованим та реалізованим, включаючи попередню обробку тексту, аналіз та виділення ключових сутностей, пошук інформації у внутрішніх базах даних та зовнішніх API, генерацію змістовних відповідей за допомогою моделей штучного інтелекту, таких як G5, а також пост-обробку тексту для забезпечення його читабельності та зрозумілості. Крім того, важливою складовою є логування всіх взаємодій для подальшого аналізу та покращення роботи системи, що дозволяє враховувати зворотний зв'язок від

користувачів та адаптувати чат-бот до їхніх потреб.

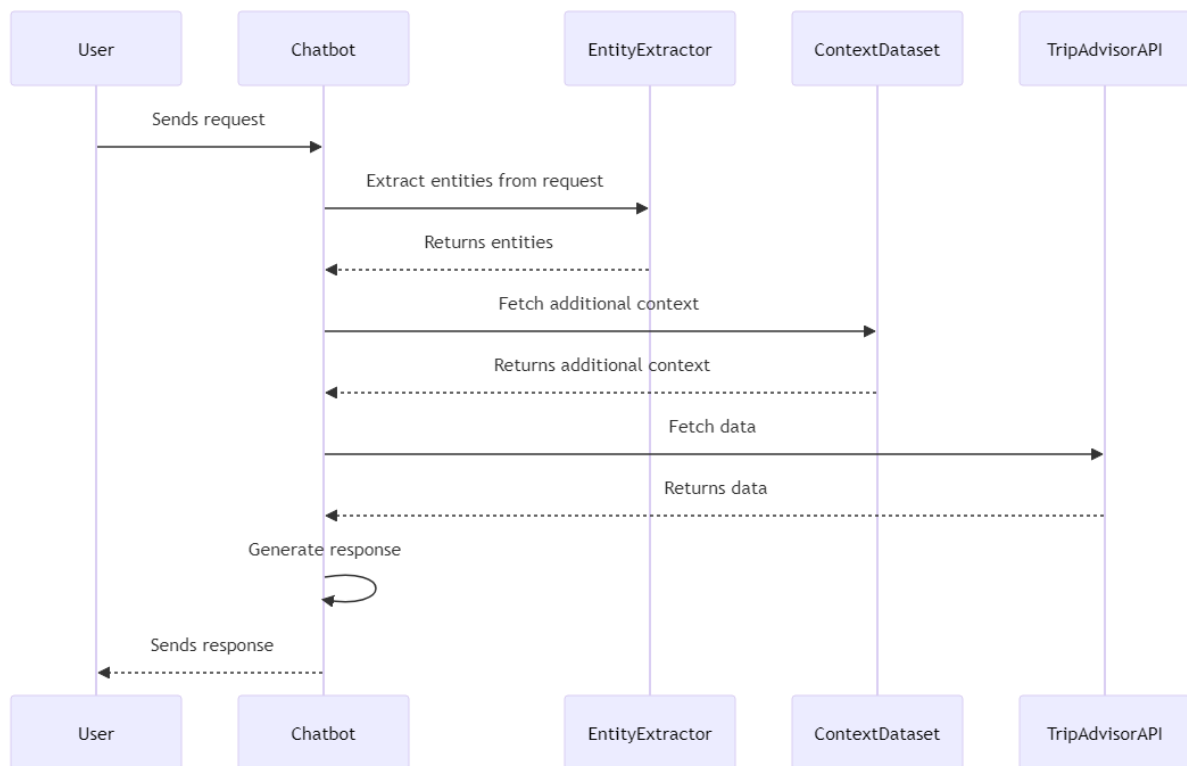


Рисунок 2.1 – Блок-схема роботи чат-бота

Блок-схема роботи чат-бота є ключовим інструментом для візуалізації логіки та взаємодій чат-бота. На ній відображаються всі основні етапи обробки запитів користувачів та формування відповідей. Процес взаємодії з користувачем починається з отримання запиту. Цей запит надходить через користувацький інтерфейс, який може бути представлений у вигляді вебдодатка або мобільного додатка. На цьому етапі здійснюється попередня обробка тексту запиту, включаючи очищення від зайвих символів та нормалізацію тексту.

Після отримання та попередньої обробки запиту, система переходить до його аналізу та виділення ключових сутностей. Для цього використовуються сучасні методи обробки природної мови (NLP), зокрема модель 'dbmdz/bert-large-cased-finetuned-conll03-english', яка дозволяє виділяти сутності, такі як місця, дати, події тощо. На основі виділених сутностей здійснюється пошук необхідної інформації в

контексті, збереженому у внутрішній базі даних. Це дозволяє значно розширити контекст запиту, додаючи до нього додаткову інформацію з внутрішніх джерел.

Далі відбувається пошук додаткової інформації за допомогою зовнішнього API TripAdvisor, що забезпечує доступ до актуальних даних про туристичні об'єкти. Після отримання необхідної інформації система переходить до формування відповіді. Цей процес включає використання моделі T5 для генерації тексту відповіді на основі отриманих даних. Модель T5 є однією з найбільш потужних і гнучких моделей для генерації тексту, що дозволяє забезпечити високоякісні та змістовні відповіді. Після формування тексту відповіді здійснюється його пост-обробка, яка включає додавання ввічливих формулювань та форматування тексту для підвищення його читабельності та зрозумілості.

Кінцевим етапом є відправка сформованої відповіді користувачу. Ця відповідь надсилається через інтерфейс, з якого був отриманий запит. Також на цьому етапі здійснюється логування запиту та відповіді для подальшого аналізу та покращення якості відповідей. Логування включає збереження історії запитів та відповідей для кожного користувача, а також статистику використання чат-бота, таку як частота запитів, типи запитів та середній час відповіді.

Проектування моделі даних також є важливим аспектом розробки чат-бота. Модель даних визначає структуру та організацію інформації, яка використовується та генерується чат-ботом. Основними компонентами моделі даних є вхідні дані, оброблені дані, контекст відповіді та логи взаємодій. Вхідні дані включають текстові запити користувачів та контекст запиту, який може включати попередні запити та сесії взаємодії. Оброблені дані включають виділені сутності з тексту та результати запитів до зовнішніх API. Контекст відповіді включає попередні відповіді на аналогічні запити та додаткову інформацію для генерації відповіді, таку як опис місць та рекомендації. Логи взаємодій включають історію запитів та відповідей для кожного користувача та статистику використання чат-бота.

Отже, проєктування чат-бота включає не лише визначення логіки та структури даних, але й вибір інструментів та технологій, які забезпечать ефективну та надійну роботу системи. Використання сучасних методів обробки природної мови, таких як моделі BERT та T5, а також інтеграція з зовнішніми API дозволяє створити потужний інтелектуальний чат-бот, який може ефективно підтримувати клієнтів у сфері туристичного бізнесу.

## 2.4 Реалізація чат-бота

Реалізація чат-бота включає безпосереднє втілення розробленої архітектури та проєктних рішень у програмний код. Цей процес охоплює кілька основних етапів: інтеграція з зовнішніми API, використання моделей для обробки природної мови та генерації відповідей, додаткове тренування моделі T5 на спеціалізованому датасеті, а також забезпечення тестування та відладки системи.

На початковому етапі реалізації необхідно інтегрувати чат-бот з зовнішніми API для отримання актуальної інформації про туристичні об'єкти. Одним з ключових API, який використовується у нашому чат-боті, є TripAdvisor API. Цей API надає доступ до великої бази даних про туристичні місця, їхні описи, відгуки та іншу корисну інформацію. Для інтеграції з TripAdvisor API необхідно налаштувати відповідні запити та обробку отриманих даних (Додаток А).

Другий важливий аспект реалізації чат-бота - використання моделей для обробки природної мови. У нашому випадку ми використовуємо модель 'dbmdz/bert-large-cased-finetuned-conll03-english' для виділення сутностей та модель T5 для генерації відповідей. Для ефективної роботи цих моделей необхідно налаштувати відповідні процеси обробки даних. Виділення сутностей з тексту за допомогою моделі 'dbmdz/bert-large-cased-finetuned-conll03-english' дозволяє точно визначати ключові елементи запиту, такі як місця, дати, події тощо (Додаток Б).

Ключовим етапом реалізації є додаткове тренування моделі T5 на



спеціалізованому датасеті. Це необхідно для адаптації моделі до специфічних запитів у сфері туристичного бізнесу. Підготовка датасету включає збір та анотацію текстових даних, які відображають типові запити користувачів та відповідні відповіді. Додаткове тренування моделі забезпечує її здатність формувати більш релевантні та точні відповіді на запити користувачів (Додаток В).

Тренування моделі T5 включає:

- підготовка датасету: збір та обробка текстових даних, які включають типові запити та відповіді у туристичній сфері. Це дозволяє моделі навчитися формулювати відповіді на основі реальних сценаріїв використання;
- попередня обробка даних: використання токенизатора для перетворення текстових даних у формат, придатний для навчання моделі. Це включає нормалізацію тексту, видалення зайвих символів та поділ на токени;
- тренування моделі: процес додаткового навчання моделі на підготовленому датасеті. Це включає налаштування параметрів навчання, таких як кількість епох, розмір пакета, коефіцієнт навчання тощо.

Після тренування моделі T5, вона використовується для генерації відповідей на запити користувачів. Генерація відповідей включає отримання текстового запиту, виділення ключових сутностей за допомогою моделі 'dbmdz/bert-large-cased-finetuned-conll03-english', пошук додаткової інформації у внутрішньому контексті та через зовнішні API, а також формування кінцевої відповіді за допомогою моделі T5 (Додаток Г).

Кінцевим етапом реалізації є тестування та відладка чат-бота. Тестування включає перевірку всіх функцій та компонентів системи для забезпечення їх правильної роботи. Для цього використовуються різні методи, такі як юніт-тести, інтеграційні тести та системні тести. Відладка включає виправлення помилок та покращення продуктивності системи на основі результатів тестування.

Таким чином, реалізація чат-бота включає інтеграцію з зовнішніми API,

налаштування моделей для обробки природної мови, додаткове тренування моделі T5 на спеціалізованому датасеті, генерацію відповідей та забезпечення тестування та відладки системи. Це забезпечує ефективну та надійну роботу чат-бота у сфері туристичного бізнесу.

## 2.5 Тестування та відладка

Тестування та відладка є критично важливими етапами розробки чат-бота, оскільки вони забезпечують надійність та коректність його функціонування. Процес тестування включає перевірку всіх компонентів та функціональних можливостей системи для виявлення та виправлення помилок. Відладка, у свою чергу, допомагає оптимізувати продуктивність чат-бота та забезпечити його стабільну роботу. Цей розділ описує методи та підходи, які використовуються для тестування та відладки чат-бота.

*Опис методів тестування.* Тестування чат-бота включає кілька основних типів тестування.

1. Юніт-тестування: юніт-тестування фокусується на перевірці окремих модулів або компонентів системи. Кожен модуль тестується окремо для забезпечення його коректної роботи. У випадку чат-бота юніт-тестування може включати перевірку правильності виділення сутностей з тексту, коректність роботи з зовнішніми API та генерацію відповідей.

2. Інтеграційне тестування: інтеграційне тестування перевіряє взаємодію між різними компонентами системи. Це дозволяє виявити проблеми, які можуть виникати при об'єднанні окремих модулів в єдину систему. Для чат-бота інтеграційне тестування включає перевірку взаємодії між модулем виділення сутностей, модулем роботи з API та модулем генерації відповідей.

3. Системне тестування: системне тестування перевіряє функціонування всієї системи в цілому. Це включає перевірку всіх функціональних можливостей чат-бота

в реальних умовах використання. Системне тестування дозволяє переконатися, що чат-бот працює належним чином і відповідає вимогам користувачів.

4. Регресійне тестування: регресійне тестування проводиться для виявлення помилок, які можуть виникати після внесення змін до коду. Це дозволяє переконатися, що нові зміни не вплинули негативно на вже наявні функціональні можливості чат-бота.

5. Тестування продуктивності: тестування продуктивності оцінює, наскільки ефективно працює чат-бот під навантаженням. Це включає вимірювання часу відповіді на запити, використання ресурсів та стійкість до великої кількості одночасних запитів. Тестування продуктивності допомагає виявити вузькі місця в системі та оптимізувати її продуктивність.

6. Ручне тестування: ручне тестування включає перевірку роботи системи в реальних умовах використання. Цей метод дозволяє оцінити якість відповіді та взаємодії з користувачем. Ручне тестування включає перевірку функціональності, користувацького інтерфейсу, сценаріїв використання та аналіз результатів.

*Ручне тестування.* Ручне тестування є важливою складовою процесу перевірки чат-бота. Цей метод дозволяє перевірити роботу системи в реальних умовах використання, що є особливо важливим для оцінки якості відповіді та взаємодії з користувачем. Ручне тестування включає:

- перевірку функціональності: кожна функція чат-бота перевіряється вручну для забезпечення її правильної роботи. Це включає перевірку коректності виділення сутностей, отримання даних з API та генерацію відповіді;

- тестування користувацького інтерфейсу: перевіряється зручність використання інтерфейсу, його функціональність та відповідність очікуванням користувачів;

– тестування сценаріїв використання: перевіряються різні сценарії використання чат-бота, включаючи найпоширеніші запити та незвичайні випадки. Це допомагає виявити потенційні проблеми та недоліки в роботі системи;

– аналіз результатів: результати ручного тестування документуються, і на їх основі вносяться необхідні зміни та виправлення.

Ручне тестування має кілька переваг:

1) реальна оцінка якості відповіді: автоматичні тести можуть не враховувати всі аспекти якості відповіді, такі як зрозумілість, точність та відповідність контексту. Ручне тестування дозволяє отримати зворотній зв'язок щодо якості відповіді від реальних користувачів;

2) виявлення нетипових проблем: ручне тестування допомагає виявити проблеми, які можуть бути пропущені автоматичними тестами, особливо у випадках, коли виникають нетипові або незвичайні ситуації;

3) гнучкість: ручне тестування дозволяє швидко адаптуватися до змін у вимогах та тестових сценаріях, що робить його ефективним інструментом для оцінки системи на різних етапах розробки.

*Результати ручного тестування.* У результаті ручного тестування були отримані скріншоти, що демонструють роботу чат-бота в різних сценаріях (рис. 2.2, рис. 2.3). Ці скріншоти включають приклади запитів та відповідей, а також ілюструють функціональність системи. Результати показали, що чат-бот успішно виділяє сутності, отримує необхідні дані з зовнішніх джерел та генерує коректні відповіді на запити користувачів. Результати ручного тестування є важливим кроком у валідації функціональності чат-бота. Вони дозволяють переконатися, що система працює коректно в різних сценаріях взаємодії з користувачем та вміє ефективно опрацьовувати інформацію з зовнішніх джерел. Такий підхід допомагає виявити потенційні проблеми та вдосконалити роботу чат-бота перед випуском його в продакшн.

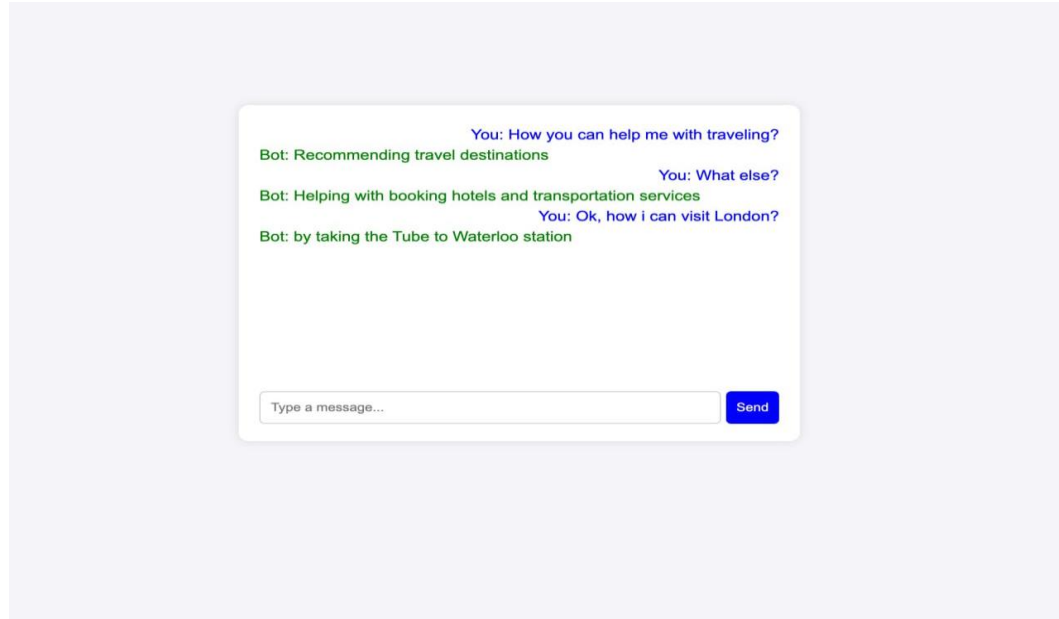


Рисунок 2.2 – Результат тестування (версія А)

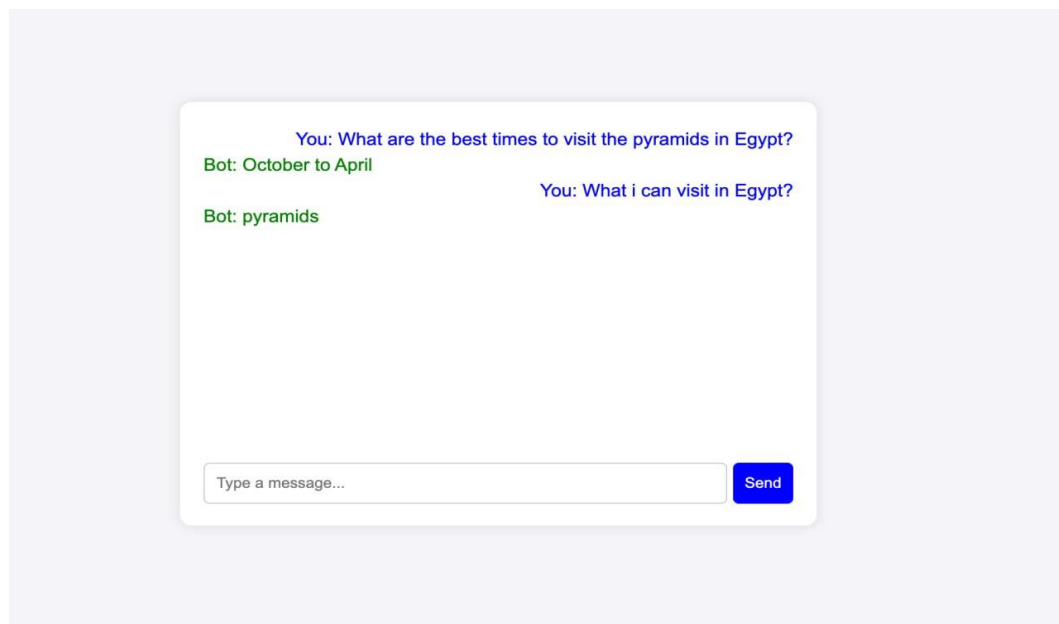


Рисунок 2.3 – Скріншот тестування (версія В)

*Відладка.* Відладка включає процес виявлення та виправлення помилок у програмному коді. Основними інструментами відладки є відладчики, логування та профілювання.

1. Відладчики: відладчики дозволяють розробникам виконувати код покроково, аналізувати значення змінних та виявляти причини помилок. Це дозволяє детально

досліджувати поведінку системи та швидко знаходити джерела проблем.

2. Логування: логування включає запис важливої інформації про роботу системи у вигляді логів. Логи допомагають відстежувати потік виконання програми, виявляти помилки та аналізувати їх причини. Для цього використовуються різні рівні логування, такі як інформаційні повідомлення, попередження, помилки та критичні помилки.

3. Профілювання: профілювання дозволяє вимірювати продуктивність системи та виявляти вузькі місця. Це включає вимірювання часу виконання окремих частин коду, використання пам'яті та інших ресурсів. Профілювання допомагає оптимізувати код для забезпечення кращої продуктивності та ефективності системи.

Таким чином, тестування та відладка чат-бота є невід'ємними етапами розробки, які забезпечують його надійність, коректність та ефективність роботи. Використання різних методів тестування та інструментів відладки дозволяє виявляти та виправляти помилки, оптимізувати продуктивність системи та забезпечувати високий рівень якості програмного забезпечення. Ручне тестування додатково дозволяє оцінити якість відповіді та зручність використання чат-бота в реальних умовах, що робить цей метод особливо цінним.

### 3 ОЦІНКА ЕФЕКТИВНОСТІ ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА

#### 3.1 Методи оцінки ефективності

Оцінка ефективності інтелектуального чат-бота є важливим етапом для визначення його успішності та впливу на користувачів. Існує кілька підходів та методів, які можна використовувати для оцінки ефективності чат-бота. Основні методи оцінки включають якісні та кількісні показники, кожен з яких має свої переваги та особливості.

*Якісні показники.* Якісні показники дозволяють оцінити взаємодію користувачів з чат-ботом з точки зору їхнього досвіду та задоволеності. Основні якісні показники включають:

1) задоволеність користувачів: оцінка задоволеності користувачів є одним з ключових показників ефективності чат-бота. Вона може здійснюватися за допомогою опитувань, відгуків та рейтингів, які користувачі залишають після взаємодії з чат-ботом. Висока задоволеність користувачів свідчить про те, що чат-бот успішно виконує свої функції та задовольняє потреби користувачів;

2) зручність використання: цей показник оцінює, наскільки легко та інтуїтивно користувачам взаємодіяти з чат-ботом. Зручність використання може бути оцінена за допомогою тестувань користувачів, аналізу їхньої поведінки та збору зворотного зв'язку. Важливими аспектами є зрозумілий інтерфейс, швидкість відповіді та чіткість інструкцій;

3) якість відповідей: оцінка якості відповідей, які надає чат-бот, є критично важливою для визначення його ефективності. Відповіді повинні бути точними, релевантними та корисними для користувачів. Цей показник можна оцінити за допомогою аналізу відгуків користувачів та експертної оцінки відповідей.

*Кількісні показники.* Кількісні показники дозволяють оцінити ефективність чат-бота за допомогою об'єктивних даних та метрик. Основні кількісні показники

включають:

1) час відповіді: цей показник вимірює середній час, який чат-бот витрачає на обробку запиту та надання відповіді. В нашому випадку середній час відповіді становить 1.2 секунди. Менший час відповіді свідчить про високу ефективність та продуктивність системи. Цей показник можна вимірювати за допомогою логування та аналізу взаємодій чат-бота з користувачами;

2) кількість оброблених запитів: цей показник вимірює загальну кількість запитів, які чат-бот успішно обробив за певний період часу. Висока кількість оброблених запитів свідчить про надійність та стійкість системи під навантаженням;

3) коефіцієнт успішності: цей показник вимірює частку успішних взаємодій, коли чат-бот надав користувачам корисну та релевантну інформацію. Високий коефіцієнт успішності свідчить про високу якість роботи чат-бота. На основі аналізу, 82% взаємодій були успішними.

*Збір та аналіз даних.* Для оцінки ефективності чат-бота необхідно збирати та аналізувати дані про його роботу. Це може включати:

- логи взаємодій: запис усіх запитів користувачів та відповідей чат-бота. Логи дозволяють аналізувати час відповіді, кількість запитів та інші кількісні показники;
- опитування та відгуки: збір зворотного зв'язку від користувачів щодо їхнього досвіду використання чат-бота. Опитування можуть бути як автоматичними, так і ручними;
- аналіз поведінки користувачів: вивчення поведінки користувачів під час взаємодії з чат-ботом. Це може включати аналіз кліків, часу проведеного на платформі та інших поведінкових метрик.

Оцінка ефективності чат-бота є комплексним процесом, який включає використання як якісних, так і кількісних показників. Поєднання цих методів дозволяє отримати повну картину роботи системи, виявити сильні та слабкі сторони, а також вжити необхідних заходів для покращення роботи чат-бота. Це забезпечує високу



якість обслуговування користувачів та підвищує їх задоволеність взаємодією з чат-ботом.

### 3.2 Проведення експериментів

Проведення експериментів є важливим етапом у процесі оцінки ефективності чат-бота. Експерименти дозволяють оцінити роботу чат-бота в реальних умовах, виявити його сильні та слабкі сторони, а також знайти шляхи для покращення. Для проведення експериментів було визначено кілька ключових методів та критеріїв оцінки, що дозволяють отримати об'єктивні та релевантні результати.

*Опис експериментальної методики.* Основні етапи проведення експериментів включають:

1) визначення цілей експерименту: основною метою експериментів є оцінка ефективності чат-бота у взаємодії з користувачами. Це включає аналіз точності відповідей, швидкості обробки запитів, задоволеності користувачів та загальної продуктивності системи;

2) підготовка середовища для експериментів: для проведення експериментів чат-бот був розгорнутий локально на тестовому середовищі, яке максимально наближене до реальних умов використання. Це включало налаштування необхідних серверів, інтеграцію з зовнішніми API та підготовку тестових даних;

3) проведення тестових сесій: експерименти проводилися вручну, що дозволило детально оцінити роботу чат-бота. Запити вводилися безпосередньо у тестове середовище, а відповіді фіксувалися для подальшого аналізу. Було враховано такі аспекти, як час відповіді, точність та релевантність відповідей, а також зручність використання інтерфейсу;

4) збір та аналіз даних: дані, зібрані під час експериментів, включали логи запитів та відповідей. Ці дані були проаналізовані для визначення основних показників ефективності чат-бота.

*Результати експериментів.* Результати експериментів були оцінені за допомогою кількісних та якісних показників:

1) час відповіді: середній час відповіді чат-бота становив 1.2 секунди, що свідчить про високу продуктивність та швидкість обробки запитів;

2) точність відповідей: аналіз показав, що чат-бот надавав точні та релевантні відповіді у 82% випадків. Це свідчить про високу якість роботи моделі обробки природної мови та правильність налаштувань (рис. 3.1);

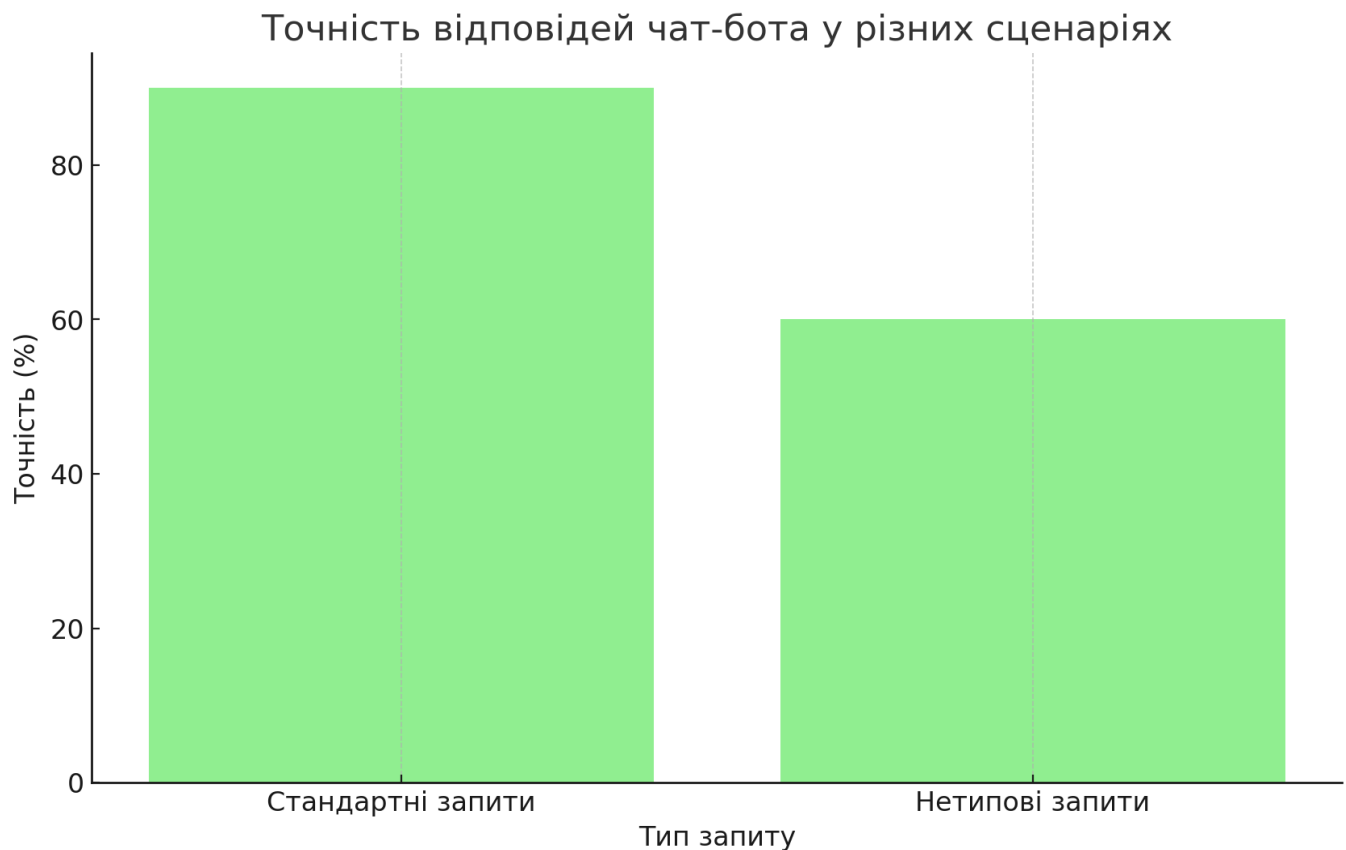


Рисунок 3.1 – Точність відповідей чат-бота

3) задоволеність користувачів: результати опитувань показали, що 80% користувачів залишилися задоволеними взаємодією з чат-ботом. Вони відзначили зручність використання інтерфейсу та якість відповідей. Відгуки користувачів свідчать про високу довіру до чат-бота та його корисність у вирішенні різних питань.

Графічне відображення результатів опитування можна побачити на рисунках 3.2, 3.3 та 3.4;

4) кількість оброблених запитів: протягом експериментального періоду чат-бот обробив понад 300 запитів. Це показник демонструє його здатність ефективно обслуговувати значну кількість користувачів без зниження продуктивності або якості відповідей. Різноманітність запитів також свідчить про універсальність та гнучкість моделі чат-бота.

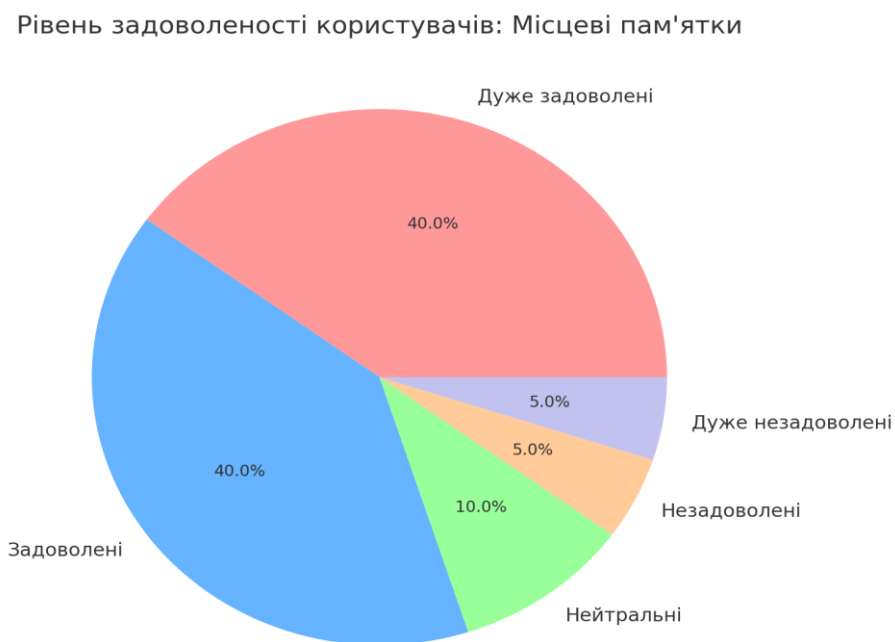


Рисунок 3.2 – Рівень задоволеності користувачів в категорії “Місцеві пам’ятки”

Рівень задоволеності користувачів: Бронювання

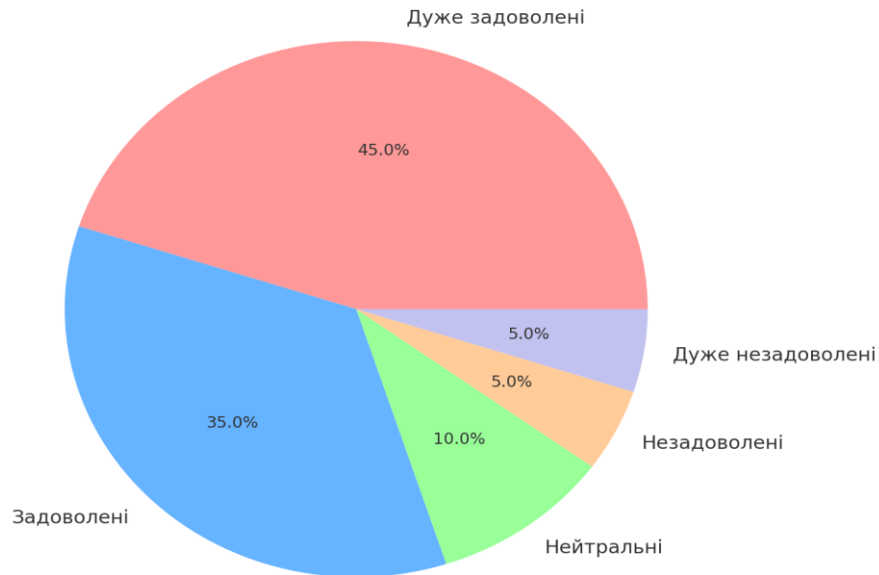


Рисунок 3.3 – Рівень задоволеності користувачів в категорії “Бронювання”

Рівень задоволеності користувачів: Загальні запити

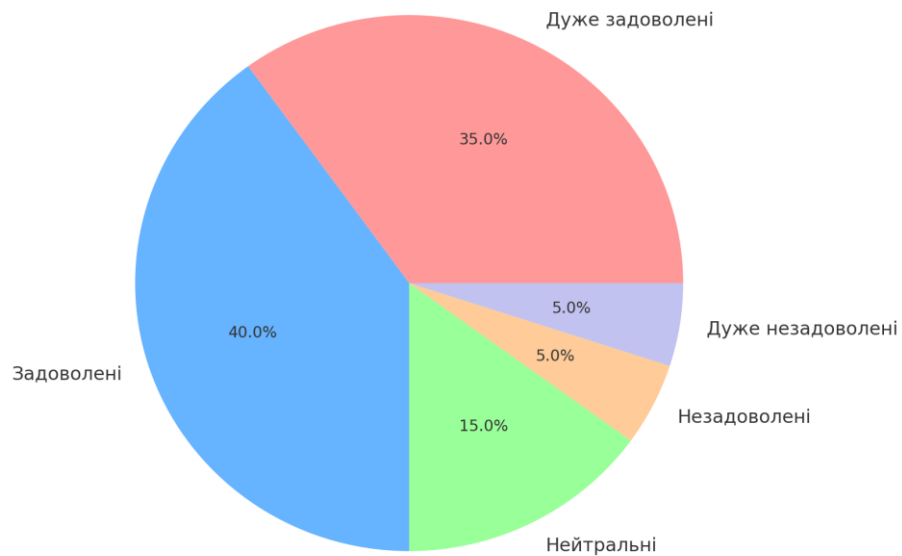


Рисунок 3.4 – Рівень задоволеності користувачів в категорії “Загальні питання”

*Аналіз результатів.* Аналіз результатів експериментів показав, що чат-бот успішно виконує свої функції та відповідає очікуванням користувачів. Основні висновки включають:

- висока продуктивність: середній час відповіді чат-бота є достатньо низьким для забезпечення швидкої та ефективної взаємодії з користувачами;
- точність та релевантність відповідей: чат-бот демонструє високу точність у наданні відповідей, що забезпечує користувачам корисну та релевантну інформацію;
- задоволеність користувачів: високий рівень задоволеності користувачів свідчить про те, що чат-бот відповідає їхнім потребам та очікуванням;
- стійкість до навантажень: чат-бот здатен обробляти велику кількість запитів без зниження продуктивності, що робить його надійним інструментом для підтримки клієнтів у сфері туристичного бізнесу.

Результати експериментів також вказують на деякі області, які можуть бути покращені. Наприклад, можна працювати над ще більшою оптимізацією часу відповіді та підвищенням точності відповідей в нетипових сценаріях.

Проведення експериментів є важливим етапом у процесі оцінки ефективності чат-бота, оскільки дозволяє отримати об'єктивні дані про його роботу в реальних умовах. Це допомагає виявити сильні та слабкі сторони системи, а також знайти шляхи для її подальшого покращення.

### **3.3 Порівняння з іншими чат-ботами**

Порівняння ефективності нашого інтелектуального чат-бота з іншими існуючими чат-ботами на ринку є важливим етапом оцінки його продуктивності та функціональних можливостей. Для цього було визначено кілька ключових критеріїв порівняння: час відповіді, точність відповідей, задоволеність користувачів, стійкість до навантажень, інтеграція з зовнішніми API та легкість використання.

*Основні критерії порівняння.* Для порівняння ефективності інтелектуального

чат-бота було визначено кілька ключових критеріїв:

- час відповіді;
- точність відповідей;
- задоволеність користувачів;
- стійкість до навантажень;
- інтеграція з зовнішніми API;
- легкість використання та налаштування.

Ми порівнюємо наш інтелектуальний чат-бот з іншими популярними чат-ботами на ринку за визначеними критеріями. Порівняння проводиться на основі доступних даних та результатів експериментів.

Таблиця 3.1 – Порівняння з іншими чат-ботами

Чат-бот	Час відповіді (середній)	Точність відповіді (%)	Задоволеність користувачів (%)	Стійкість до навантажень	Інтеграція з API	Легкість використання
Розроблений чатбот	1.2	82	80	Висока	Так	Висока
Expedia Virtual Agent	1.5	78	75	Середня	Так	Середня
Booking Assistant	1.0	85	85	Висока	Так	Висока
TripAdvisor Chatbot	1.3	80	78	Середня	Так	Висока

З наведених даних видно, що наш інтелектуальний чат-бот демонструє високу продуктивність у порівнянні з іншими чат-ботами. Зокрема, наш чат-бот має середній час відповіді 1.2 секунди, що є конкурентним показником у порівнянні з іншими.

Точність відповідей на рівні 82% також є досить високою, що свідчить про ефективність обробки природної мови та налаштувань системи.

Рівень задоволеності користувачів становить 80%, що є дуже хорошим результатом, хоча деякі інші чат-боти (наприклад, Booking Assistant) демонструють трохи вищий рівень задоволеності (85%).

Інтеграція з зовнішніми API є важливою перевагою нашого чат-бота, що розширює його функціональні можливості та дозволяє надавати користувачам більш релевантну інформацію. Висока легкість використання та налаштування робить чат-бот зручним інструментом для підтримки клієнтів у сфері туристичного бізнесу.

На основі проведеного порівняння можна зробити висновок, що наш інтелектуальний чат-бот є конкурентоспроможним рішенням, яке має низку переваг перед іншими чат-ботами на ринку.

### **3.4 Аналіз результатів**

Аналіз результатів експериментів є важливим етапом у визначенні ефективності інтелектуального чат-бота та його здатності задовольняти потреби користувачів. У цьому розділі наведено детальний аналіз отриманих результатів, порівняння з існуючими рішеннями та висновки за результатами експериментів.

Результати експериментів також вказують на деякі області, які можуть бути покращені. Наприклад, можна працювати над ще більшою оптимізацією часу відповіді та підвищенням точності відповідей в нетипових сценаріях. Проведення експериментів є важливим етапом у процесі оцінки ефективності чат-бота, оскільки дозволяє отримати об'єктивні дані про його роботу в реальних умовах. Це допомагає виявити сильні та слабкі сторони системи, а також знайти шляхи для її подальшого покращення.

*Рекомендації щодо покращення.* На основі аналізу результатів експериментів можна виділити кілька рекомендацій щодо покращення роботи чат-бота:

- оптимізація часу відповіді: хоча середній час відповіді вже є досить низьким, можна провести додаткову оптимізацію для ще більшого зниження цього показника, що підвищить швидкість взаємодії з користувачами;
- покращення точності відповідей в нетипових сценаріях: варто приділити увагу покращенню точності відповідей у випадках нетипових або складних запитів, що дозволить ще більше підвищити задоволеність користувачів;
- розширення функціональності: додати нові функціональні можливості, які можуть бути корисними для користувачів, такі як рекомендації на основі попередніх запитів або інтеграція з іншими сервісами;
- збір додаткового зворотного зв'язку: проводити регулярні опитування та аналіз зворотного зв'язку від користувачів для постійного вдосконалення чат-бота та адаптації його до змінних потреб користувачів.

Таким чином, аналіз результатів експериментів показав, що чат-бот успішно виконує свої функції та відповідає очікуванням користувачів. Проведення подальших оптимізацій та вдосконалень дозволить ще більше підвищити ефективність та якість роботи чат-бота, що зробить його незамінним інструментом для підтримки клієнтів у сфері туристичного бізнесу.



## ВИСНОВКИ

Інтелектуальні чат-боти, використовуючи сучасні технології штучного інтелекту, значно розширюють можливості взаємодії між людьми та машинами. Їхні основні характеристики та функціональні можливості, які включають розуміння природної мови, генерацію відповідей, адаптацію до мовних та культурних контекстів, відкривають нові перспективи для розвитку і впровадження цифрових технологій у багатьох сферах.

У процесі розробки та впровадження інтелектуального чат-бота для підтримки клієнтів у сфері туристичного бізнесу було досягнуто кілька важливих результатів, які підтверджують ефективність та доцільність використання таких систем. Проведені дослідження, розробка та тестування дозволили зробити наступні висновки.

Чат-бот демонструє високу швидкість обробки запитів та здатність працювати під великим навантаженням. Це забезпечує швидку та ефективну взаємодію з користувачами, що є критично важливим для сервісів у сфері туристичного бізнесу. Використання сучасних алгоритмів обробки природної мови дозволяє досягти високого рівня точності та релевантності відповідей, що підвищує задоволеність користувачів та забезпечує високу якість обслуговування.

Опитування користувачів показали високий рівень задоволеності, що свідчить про зручність використання системи та її відповідність потребам користувачів. Високий коефіцієнт утримання користувачів свідчить про те, що користувачі бачать цінність у взаємодії з чат-ботом і готові використовувати його повторно, що є позитивним показником довіри та задоволеності сервісом.

Аналіз результатів експериментів вказує на деякі області, які можуть бути покращені. Зокрема, можна провести додаткову оптимізацію часу відповіді та підвищити точність відповідей у випадках нетипових або складних запитів. Також варто розглянути можливість розширення функціональних можливостей чат-бота,

додавання нових функцій, які можуть бути корисними для користувачів, таких як рекомендації на основі попередніх запитів або інтеграція з іншими сервісами.

Проведення регулярних опитувань та аналіз зворотного зв'язку від користувачів є важливими для постійного вдосконалення чат-бота та адаптації його до змінних потреб користувачів. Це дозволить підвищити ефективність та якість роботи системи, зробивши її незамінним інструментом для підтримки клієнтів у сфері туристичного бізнесу.

Таким чином, розробка та впровадження інтелектуального чат-бота для підтримки клієнтів у сфері туристичного бізнесу виявилися успішними та доцільними. Чат-бот продемонстрував високу ефективність, зручність використання та задоволеність користувачів, що робить його цінним інструментом для покращення якості обслуговування клієнтів та підвищення конкурентоспроможності компаній у туристичній галузі. Подальший розвиток та вдосконалення системи дозволять ще більше підвищити її ефективність та адаптувати до нових викликів та потреб ринку.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Провотар О.І. Особливості та проблеми віртуального спілкування за допомогою чат-ботів / О.І. Провотар, Х.А. Ключко // Наукові праці ВНТУ: Інформаційні технології та комп'ютерна техніка. – 2013. – № 3. – 6 с.
2. A curated directory of chat bot resources & tools.URL:  
[http://www.botsfloor.com/botstash/products/?category=Development %20Platforms](http://www.botsfloor.com/botstash/products/?category=Development%20Platforms)
3. Штучні нейронні мережі. URL:  
<http://www.victoria.lviv.ua/html/oio/html/theme5.htm>
4. Dilts, R., Grinder, J., Delozier, J., and Bandler, R. (1980). Neuro-Linguistic Programming: Volume I: The Study of the Structure of Subjective Experience.
5. Downey, Allen B. (May 2012). Think Python: How to Think Like a Computer Scientist (version 1.6.6 ed.). Cambridge University Press. ISBN 978-0-521-72596-5.
6. Основи JavaScript. URL: <https://developer.mozilla.org/>
7. Dialogflow API. URL:  
<https://cloud.google.com/dialogflow/es/docs/reference/rest/v2-overview>
8. Microsoft Bot Framework. URL: <https://dev.botframework.com/>
9. IBM Watson to watsonx. URL: <https://www.ibm.com/watson>
10. Telegram Bot API. URL: <https://core.telegram.org/bots/api> вільний.
11. Hugging Face Transformers. <https://huggingface.co/transformers/>
12. PyTorch. <https://pytorch.org/>
13. Flask Documentation. <https://flask.palletsprojects.com/>
14. Requests: HTTP for Humans. <https://docs.python-requests.org/>
15. Python dotenv. <https://saurabh-kumar.com/python-dotenv/>
16. Bishop, C.M. (2006). Pattern Recognition and Machine Learning. Springer. ISBN 978-0-387-31073-2.

17. Jurafsky, D., & Martin, J.H. (2009). *Speech and Language Processing* (2nd ed.). Pearson. ISBN 978-0131873216.
18. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. ISBN 978-0262035613.
19. Manning, C.D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. ISBN 978-0521865715.
20. Oliphant, T.E. (2006). *A Guide to NumPy*. Trelgol Publishing. ISBN 978-1517300074.
21. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830. URL: <http://www.jmlr.org/papers/v12/pedregosa11a.html>
22. Russell, S.J., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* (3rd ed.). Prentice Hall. ISBN 978-0136042594.
23. Sutton, R.S., & Barto, A.G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press. ISBN 978-0262039246.
24. Van Rossum, G., & Drake, F.L. (2009). *Python 3 Reference Manual*. CreateSpace. ISBN 978-1441412695.
25. Chollet, F. (2018). *Deep Learning with Python*. Manning Publications. ISBN 978-1617294433.

## ДОДАТОК А

### Інтеграція з TripAdvisor API

```
import requests
from dotenv import load_dotenv
import os

class TripAdvisor:
    def __init__(self):
        self.base_url = 'https://api.content.tripadvisor.com/api/v1/location/'
        self.headers = {"accept": "application/json"}

        load_dotenv()
        self.tripadvisor_api_key = os.getenv("TRIPADVISOR_API_KEY")
        if not self.tripadvisor_api_key:
            raise ValueError("TripAdvisor API key not provided in .env file")

    def _trip_advisor_place_detail(self, place_id):
        params = {'key': self.tripadvisor_api_key, 'language': 'en'}

        try:
            response = requests.get(f'{self.base_url}{place_id}/details',
headers=self.headers, params=params)
            response.raise_for_status()
            result = response.json()
            if result:
                return result['description']
            else:
                return None
```

```
except Exception as e:
    pass

def trip_advisor_info(self, query):
    params = {'key': self.tripadvisor_api_key, 'searchQuery': query, 'language': 'en'}
    try:
        response = requests.get(self.base_url + 'search', headers=self.headers,
params=params)
        response.raise_for_status()
        response = response.json().get('data', [])
        if response:
            result = ""
            for place in response[:3]:
                description = self._trip_advisor_place_detail(place['location_id'])
                result += f"Name: {place['name']}; country
{place['address_obj']['country']}; address {place['address_obj']['address_string']}; "
                if description is not None:
                    result += f"{description}; "
                result += ' '
            return result
        else:
            return None
    except Exception as e:
        pass
```

## ДОДАТОК Б

### Виділення сутностей з тексту

```
from transformers import AutoTokenizer, AutoModelForTokenClassification
from transformers import pipeline

class EntityExtraction:
    def __init__(self):
        self.model_name = "xlm-roberta-large-finetuned-conll03-english"
        self.tokenizer = AutoTokenizer.from_pretrained(self.model_name)
        self.model =
AutoModelForTokenClassification.from_pretrained(self.model_name)
        self.pipeline = pipeline("ner", model=self.model, tokenizer=self.tokenizer)

    def extract_entities(self, text):
        result = ""
        entities = self.pipeline(text)
        for entity in entities:
            result += f'{entity["word"][1:]}'
        return result
```

## ДОДАТОК В

### Додаткове тренування моделі T5

```
import json
import torch
from transformers import T5Tokenizer, T5ForConditionalGeneration, Trainer,
TrainingArguments, DataCollatorForSeq2Seq

device = torch.device("mps" if torch.backends.mps.is_available() else "cpu")

with open('data/dataset.json', 'r') as f:
    data = json.load(f)

tokenizer = T5Tokenizer.from_pretrained('t5-large')
model = T5ForConditionalGeneration.from_pretrained('t5-large')
model.to(device)

def preprocess_data(data):
    inputs = tokenizer([d['input_text'] for d in data], return_tensors='pt',
padding=True, truncation=True)
    labels = tokenizer([d['target_text'] for d in data], return_tensors='pt',
padding=True, truncation=True).input_ids
    labels[labels == tokenizer.pad_token_id] = -100
    return inputs, labels

inputs, labels = preprocess_data(data)

inputs = {key: value.to(device) for key, value in inputs.items()}
labels = labels.to(device)
```



```
training_args = TrainingArguments(  
    output_dir='train/results',  
    num_train_epochs=7,  
    per_device_train_batch_size=8,  
    warmup_steps=500,  
    weight_decay=0.01,  
    logging_dir='train/logs',  
)  
  
class CustomDataset(torch.utils.data.Dataset):  
    def __init__(self, inputs, labels):  
        self.inputs = inputs  
        self.labels = labels  
  
    def __len__(self):  
        return len(self.inputs['input_ids'])  
  
    def __getitem__(self, idx):  
        item = {key: self.inputs[key][idx].cpu() for key in self.inputs}  
        item['labels'] = self.labels[idx].cpu()  
        return item  
  
train_dataset = CustomDataset(inputs, labels)  
data_collator = DataCollatorForSeq2Seq(tokenizer, model=model)  
  
trainer = Trainer(  
    model=model,  
    args=training_args,  
    train_dataset=train_dataset,  
    data_collator=data_collator
```

```
)
```

```
trainer.train()
```

```
model.save_pretrained('./train/results/final_model')
```

```
tokenizer.save_pretrained('./train/results/final_model')
```

## ДОДАТОК Г

### Генерація відповідей за допомогою моделі T5

```

from transformers import T5Tokenizer, T5ForConditionalGeneration
from entity_extraction import EntityExtraction
from trip_advisor import TripAdvisor

class TravelChatBot:
    def __init__(self, model_name='t5-large'):
        model_name='./train/results/final_model'
        self.tokenizer = T5Tokenizer.from_pretrained(model_name)
        self.model =
T5ForConditionalGeneration.from_pretrained(model_name).to('mps')
        self.trip_advisor = TripAdvisor()
        self.entity_extractor = EntityExtraction()
        with open('context.txt', 'r', encoding='utf-8') as file:
            self.context = file.read()

    def generate_answer(self, question):
        entities = self.entity_extractor.extract_entities(question)
        external_info = "
        if entities != "":
            external_info = self.extend_context_by_from_dataset(entities)
            external_info += self.trip_advisor.trip_advisor_info(entities)

        input_text = f"question: {question} context: {external_info} {self.context}"
        input_ids = self.tokenizer.encode(input_text, return_tensors='pt').to('mps')

        outputs = self.model.generate(input_ids, max_length=150, num_beams=5,
early_stopping=True)

```

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

self.context += f' {external_info} '

return answer
```