

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра комп'ютерних наук

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему «Інформаційна технологія створення інтелектуального чат-бота»

Виконав: студент 2 курсу,
групи 2КН-19м
спеціальності 122 «Комп'ютерні науки»

Кудрявцев Д. С.

Керівник: д.т.н., проф. Яровий А.А.

Рецезент: к.т.н., доц. Черноволик Г.О.

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ КН _____

д.т.н., проф.. Яровий А.А.

(підпис)

“ _____ ” _____ 2020 року

ЗАВДАННЯ

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра зі спеціальності:
122 – «Комп'ютерні науки»

08-22.МКР.004.15.054.ПЗ

Магістранта групи 2КН-19м Кудрявцева Дмитра Станіславовича

Тема магістерської кваліфікаційної роботи: «Інформаційна технологія створення інтелектуального чат-бота»

Вхідні дані: Кількість предметних областей для аналізу тексту – не менше 2; розмір вибірок для навчання та тестування – не менше 10000 діалогів; кількість ключових слів для одного діалогу – не менше 10, розмір текстів – 50-600 слів, мова текстів – англійська.

Короткий зміст частин магістерської кваліфікаційної роботи:

1. Графічна: Архітектурна модель інформаційної технології, Схема взаємодії між модулями операційного контролю, модулю попередньої обробки та модулям інтелектуального аналізу.

2. Текстова (пояснювальна записка): вступ, обґрунтування доцільності розробки інформаційної технології створення інтелектуальних чат-ботів, аналіз засобів та методів для реалізації інформаційної технології створення інтелектуальних чат-ботів, програмна реалізація інформаційної технології створення інтелектуальних чат-ботів, економічна частина, висновки, перелік використаних джерел, додатки.

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Обґрунтування доцільності розробки інформаційної технології створення інтелектуального чат-бота			Аналітичний огляд літературних джерел, розділ 1 ПЗ
2	Аналіз методів та засобів для реалізації інформаційної технології створення інтелектуального чат-бота			Розділ 2
3	Програмна реалізація інформаційної технології створення інтелектуального чат-бота			Розділ 3
4	Економічна частина			Розділ 4
5	Апробація та/або впровадження результатів дослідження			Стаття у міжнародному фаховому виданні
6	Оформлення пояснювальної записки, графічного матеріалу та презентації			Записка, графічний матеріал, презентація

Консультанти з окремих розділів магістерської кваліфікаційної роботи

1. Науковий керівник _____
(підпис)
“___” _____ 20р.

Д.Т.Н., професор
наук. ступінь, вчене звання (посада)
А. А. Яровий
ініціали та прізвище

2. Економічна частина _____
(підпис)
“___” _____ 20р.

к.е.н., доцент кафедри ЕПВМ
наук. ступінь, вчене звання (посада)
Бальзан М.В.
ініціали та прізвище

Дата попереднього захисту роботи “___” _____ 20р.

Рецензент _____
(підпис)

к.т.н., доц. кафедри ПЗ
Черноволик Г.О.
наук. ступінь, вчене звання (посада)

ініціали та прізвище

Завдання видав
науковий керівник _____
(підпис)
“___” _____ 20р.

Д.Т.Н., професор
наук. ступінь, вчене звання (посада)
А. А. Яровий
ініціали та прізвище

Завдання отримав магістрант _____
(підпис)
“___” _____ 20р.

Д. С. Кудрявцев
ініціали та прізвище

АНОТАЦІЯ

У даній магістерській кваліфікаційній роботі реалізована інформаційна технологія створення інтелектуальних чат-ботів для задачі проведення діалогів користувача інформаційної технології та створеного чат-бота. Основним призначенням даної технології є формування предметно-спрямованих чат-ботів у сфері підтримки користувача в обраних предметних областях із можливістю автономної роботи.

Під час виконання роботи досліджено особливості реалізації інформаційної технології на основі поєднання інтелектуальної обробки даних та засобів для функціонування чат-боту. Розроблено структурну модель інтелектуального чат-бота, охарактеризовано основні та другорядні вимоги до створення подібних систем-аналогів. Проведено порівняння застосування декількох джерел інформації та вплив обраної тематики на систему під час розпізнавання текстової інформації.

Здійснено програмну реалізацію усіх компонентів інформаційної технології на платформі .NET Core 3.1 із використанням сервісу TensorFlow. Проведено тестування розробленого програмного продукту та проаналізовано його працездатність на наборі даних.

ABSTRACT

In this master's qualification work the information technology of creation of intelligent chatbots for the task of carrying out dialogues of the user of information technology by the created chatbot was realized. The main goal of this technology is the formation of subject-oriented chatbots in the field of user support in selected subject areas with the possibility of autonomous work.

During the research, the information technology implementation based on the combination of intellectual data processing and means for the functioning of the chatbot were investigated. The structural model of the intelligent chatbot was developed, the basic and secondary requirements to the creation of similar systems-analogs were characterized. A comparison of the use of several sources of information and the impact of selected topics on the system during the recognition of textual information were analyzed.

Software implementation of all information technology components on the .NET Core 3.1 platform using the TensorFlow service has been implemented. The developed using of created information technology chatbot was tested and its operability on the data set was analyzed.

ЗМІСТ

ВСТУП	6
<u>1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНИХ ЧАТ-БОТІВ</u>	11
1.1 Аналіз предметної області аналізу текстової інформації.....	11
1.2 Аналіз сучасних технологій створення інтелектуальних чат-ботів	14
1.3 Аналіз об'єкту проектування	16
1.3.1 Постановка задачі створення предметно-орієнтованих чат-ботів	16
1.3.2 Особливості створення предметно-орієнтованих чат-ботів	17
1.3.3 Характеристика та аналіз систем-аналогів	22
1.4 Висновок.....	23
<u>2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНИХ ЧАТ-БОТІВ</u>	25
2.1 Аналіз особливостей вхідної інформації та обґрунтування вибору засобів інформаційної технології.....	25
2.2 Аналіз методів розпізнавання та аналізу текстової інформації для декількох предметних областей.....	27
2.3 Аналіз та обґрунтування вибору зовнішніх інформаційних технологій при реалізації інтелектуальних чат-ботів	39
2.4 Етапи створення інформаційної технології та структурна модель інтелектуального чат-бота.....	41
2.5 Висновок	56
<u>3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНИХ ЧАТ-БОТІВ</u>	58
3.1 Обґрунтування вибору програмно-апаратної платформи.....	58
3.2 Обґрунтування вибору мови програмування.....	60
3.3 Розробка схеми алгоритму взаємодії основних компонентів інформаційної технології.....	65
3.4 Тестування інтелектуального чат-бота в обраних предметних областях та аналіз результатів.....	68
3.5 Висновки	76
<u>4 ЕКОНОМІЧНА ЧАСТИНА</u>	78
<u>ВИСНОВКИ</u>	89
<u>ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ</u>	91
<u>Додаток А. Інструкція користувача</u>	95
<u>Додаток Б. Лістинг роботи програми</u>	96
<u>Додаток В. Сертифікат</u>	105
<u>Додаток Г. Сертифікат учасників міжнародної конференції</u>	106
<u>Додаток Д. Графічна частина</u>	107

ВСТУП

Сучасний світ поступово переходить від сфери індустріального розвитку та матеріальних відносин у еру цифрових технологій та єдиної системи комунікації, що представлена мережею «Інтернет».

Після створення першої ЕОМ у 1945 році дотепер минуло понад 70 років і за цей проміжок часу електронні носії інформації стали найбільш поширеним засобом не лише для зберігання інформації, але й для її аналізу, потреб комунікації [1]. Завдяки стрімкому розвитку технологій обробки даних та збільшення об'єму інформації, натепер постає проблема систематизації накопленої інформації та її розпізнавання для подальшої коректної обробки.

Незважаючи на різноманіття інформації та її видів, електронна інформація поділяється на текстову (символьну) та графічну у форматах аудіо- та відеозаписів. Зважаючи на засоби для аналізу даних типів інформації найбільш поширеними є технології інтелектуального аналізу даних у вигляді застосування машинного навчання, нейронних мереж, технології агрегації даних, використання розподілених обчислень. При застосуванні даних технологій рівень розпізнавання інформації та її класифікація за тематикою та предметною областю має досить відносний характер. Для підвищення рівня розпізнавання текстової інформації в класичному випадку обирається одна предметна область, та на її основі формується контекстуальна цінність інформації у порівнянні із обраною предметною областю. Ще одним із яскравих прикладів розпізнавання текстової інформації застосовуються технології обробки природної мови, що містять набір предметних областей та їх комбінування. Кінцевим продуктом використання даних технологій є створення програмних додатків для розпізнавання текстової інформації чи чат-ботів, що вдало зарекомендували себе, як засоби для комунікації, системи підтримки прийняття рішень та інтелектуальні помічники. Основним недоліком даних систем є використання обмеженої бази знань та обмеження однією предметною областю [2]. Завдяки

цьому, сучасні рішення в даному напрямку є новими та містять наукову новизну для подальших досліджень.

Під час обрання теми магістерської кваліфікаційної роботи основну увагу було направлено на можливість дослідження оригінального підходу до вирішення задачі надання ефективної підтримки в обраних сферах чи предметних областях, а саме створення інтелектуальних чат-ботів із можливостями аналізу декількох предметних областей. Подібно до людини, що має можливість вільно оперувати багатьма предметними областями, кожна предметна область донині потребує окремого розгляду, а це формує занадто складну модель розпізнавання інформації. При застосуванні декількох нейронних мереж для кожної з предметних областей принцип розпізнавання не змінюється, але з'являється необхідність використання порогу адекватності розпізнавання. Створення подібних систем та програмних модулів у вигляді інтелектуальних чат-ботів та програмних додатків є досить складним процесом та потребує чималої кількості даних для перевірки їх функціональності. Повний цикл створення інтелектуальних чат-ботів є основою створюваної інформаційної технології.

Актуальність. Класифікація текстової інформації за допомогою програмних засобів та алгоритмів обробки даних відіграє дедалі важливіше значення в сучасному світі завдяки надзвичайній кількості створеного контенту, який потребує сортування та аналізу його вмісту. При автоматичному створенні текстової інформації її контекст зазвичай містить однотипну інформацію, що легко відрізнити від автентичного тексту з точки зору експерта. Але експерт має обмежений час для аналізу. Тому задача автоматичного аналізу тексту та визначення тематики текстової інформації є актуальною задачею для дослідження в магістерській кваліфікаційній роботі.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка спеціалізованих засобів штучного інтелекту на

основі інтелектуального аналізу даних та машинного навчання» та плану наукової та навчально-методичної роботи кафедри.

Мета і завдання досліджень. Метою магістерської кваліфікаційної роботи є підвищення точності та швидкодії розпізнавання тексту для декількох предметних областей.

Для досягнення поставленої мети потрібно вирішити наступні завдання:

- 1) здійснити аналіз предметної області інтелектуального аналізу тексту;
- 2) здійснити аналіз методів та засобів для реалізації інформаційної технології створення інтелектуальних чат-ботів;
- 3) розробити алгоритм функціонування усіх програмних модулів;
- 4) виконати програмну реалізацію інформаційної технології та створити на її основі інтелектуальний чат-бот для щонайменше двох предметних областей;
- 5) провести тестування інтелектуального чат-бота на чистому наборі даних та проаналізувати отримані результати.

Об'єктом дослідження – процес розпізнавання тексту інтелектуальним чат-ботом для декількох предметних областей.

Предмет дослідження – програмні засоби створення інтелектуальних чат-ботів та аналізу тексту.

Методи дослідження. У роботі використані наступні методи наукових досліджень: інтелектуального аналізу даних, теорії штучних нейронних мереж для реалізації інформаційної технології створення інтелектуальних чат-ботів, об'єктно-орієнтованого програмування.

Наукова новизна отриманих результатів.

1. Удосконалено інформаційну модель семантичного аналізу тексту, яка на відміну від існуючих містить введений оригінальний набір правил обробки та формування списку стоп-слів, що забезпечило підвищення точності семантичного аналізу тексту за рахунок зменшення кінцевого розміру семантичного ядра.
2. Розроблено інформаційну технологію створення інтелектуального чат-бота, яка на відміну від існуючих містить удосконалену інформаційну модель

семантичного аналізу тексту, що забезпечило підвищення точності та швидкодії розпізнавання тексту для декількох предметних областей.

Практичне значення одержаних результатів полягає в тому, що на основі проведених досліджень розроблено програмне забезпечення для аналізу текстової інформації.

Створена інформаційна технологія сприяє покращенню процесу аналізу текстової інформації, зокрема:

- Розроблено алгоритм роботи програмного забезпечення для усіх компонентів інформаційної технології;
- Розроблено програмні засоби для інтелектуального аналізу текстової інформації на основі рекурентної нейронної мережі із застосуванням методу довгої короткочасної пам'яті. А також запропоновано набір правил для семантичного аналізу тексту включаючи покращення стандартного семантичного аналізу.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології створення інтелектуальних чат-ботів.

Особистий внесок здобувача. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, написаних у співавторстві, здобувачу належать: аналіз методів попередньої обробки текстів, інтелектуальний аналіз текстів, удосконалення методу семантичного аналізу тексту.

Апробація результатів роботи. Результати досліджень було апробовано на XXI конференції «Інтернет-Освіта-Наука» (ІОН-2020) [3] та на XV міжнародній науковій технічній конференції «Комп'ютерні науки та інформаційні технології» у 2020 році [4]. Отримано диплом 2-ого ступеня на всеукраїнському конкурсі студентських наукових робіт з напрямку «Інформатика та кібернетика» за роботу пов'язану із темою досліджень [5].

Публікації. За результатами досліджень опубліковано тези регіональної конференції ВНТУ [6], отримано авторське свідоцтво на комп'ютерну програму [7], а також три наукові статті, із них дві опубліковані у іноземних фахових виданнях, що входять до науково-метричної бази SCOPUS [8, 9], третя стаття опублікована у фаховому виданні з технічних наук [10].

1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНИХ ЧАТ-БОТІВ

1.1 Аналіз предметної області аналізу текстової інформації

Обробка природної мови (ОПМ) – одна з задач комп'ютерних наук та штучного інтелекту, пов'язана з взаємодією між комп'ютерами та людськими (природними) мовами, зокрема, створення програмних засобів для обробки та аналізу великих обсягів даних природних мов [11].

Історія обробки природної мови, почалась в 1950-х роках, хоча перші напрацювання можна знайти з більш ранніх періодів. У 1950 році Алан Тьюрінг опублікував статтю під назвою "Інтелект", яка запропонувала тест Тьюрінга як критерій інтелекту [11].

Перші системи обробки текстової інформації були розроблені шляхом ручного кодування набору правил, написання граматик або розробки евристичних правил для виведення. Проте цього не вистачало для природної мовної варіації. Наприкінці 1980-х і середини 1990-х років більшість досліджень, пов'язаних з обробкою природних мов, значною мірою спиралися на машинне навчання. Парадигма машинного навчання вимагає замість використання статистичного висновку, автоматичне вивчення таких правил шляхом аналізу великих наборів типових реальних прикладів (множини - це набір документів, можливо, з анотаціями для людини або комп'ютера) [12].

Багато різноманітних класів алгоритмів машинного навчання були застосовані до задач обробки природних мов та більшість з них не виправдали сподівання. Дані алгоритми приймають в якості вхідних даних великий набір "особливостей", які генеруються з вхідних даних. Деякі з найбільш ранніх алгоритмів, такі як дерева рішень, виробляли системи жорстких правил if-then, подібні до систем рукописних правил, які тоді були загально-прийнятими. Проте дослідження все частіше зосереджувалися на статистичних моделях, які роблять м'які ймовірнісні рішення, засновані на приєднанні реальних ваг до кожної

вхідної функції. Такі моделі мають перевагу в тому, що вони можуть висловлювати відносну впевненість у багатьох різних можливих відповідях, а не тільки в одній, виробляючи більш надійні результати, коли така модель включена як складова потужнішої системи [13].

Системи, що базуються на алгоритмах машинного навчання, мають багато переваг перед правилами, що додаються вручну.

Процедури навчання, що використовуються під час машинного навчання, автоматично фокусуються на найбільш поширених випадках, тоді як при написанні правил людиною здебільшого зовсім не очевидно, яка лексема має перевагу в реченні [12].

Процедури автоматичного навчання можуть використовувати алгоритми статистичного висновку для створення моделей, які є надійними до невідомих вхідних даних (наприклад, містять слова або структури, які раніше не були помічені), і помилкового введення (наприклад, з помилковими словами або словами, випадково пропущеними). Як правило, обробка такого вхідного матеріалу граційна з рукописними правилами - або, створення систем рукописних правил, які роблять м'які рішення - надзвичайно складна задача, схильна до помилок і вимагає багато часу.

Системи засновані на автоматичному вивченні правил можливо зробити більш точними, надавши більше вхідних даних. Проте, системи, засновані на рукописних правилах, можуть бути більш точними лише за рахунок збільшення складності правил, що є більш завданням. Зокрема, існують обмеження на складність систем, заснованих на ручних правилах, за якими системи стають все більш некерованими. Створення більшої кількості даних для введення в системи машинного навчання просто вимагає відповідного збільшення кількості відпрацьованих робочих годин, як правило, без значного збільшення складності процесу анотації. Щодо сучасних засобів для розпізнавання природної мови, то натеper виділяють два основні способи вирішення даної задачі. Перший полягає у використанні машинного навчання, як натренованій системі розпізнавання, що містить власну базу знань та керується її вмістом. Другий полягає у застосуванні

нейронних мереж для задачі розпізнавання та класифікації [14]. Нейронні мережі – це досить потужний засіб для вирішення не лише задач розпізнавання чи класифікації, а й для визначення сенсу інформації та формуванню на його основі, певного набору власної інформації, якщо така існує. Подібним чином, людина використовує власну пам'ять. Для задачі розпізнавання природної мови, найбільш уживаними є рекурентна нейронна мережа та її різновиди. Основною перевагою рекурентної нейронної мережі є зворотне поширення імпульсу та стійкість до розміру вхідних даних, що дозволяє опрацьовувати величезні обсяги текстової інформації, без використання надлишкових ресурсів. Формування сенсу інформації є складним процесом та включає в себе декілька етапів. Для початку інформація поділяється на атомарні частини, відповідно текст поділяється на речення та слова, що відділяються знаками пунктуації, що теж, зазвичай враховуються. Наступним кроком є подання одиниці поділеної інформації (слова) на вхід нейронної мережі, причому при кожному наступному поданому слові, разом із ним подається ланцюг попередніх слів, що були складовими одного речення. Даний метод дозволяє знаходити сенс не лише усього повідомлення у вигляді його тематики чи предметної області, а й кожної з лексичних частин (речень).

Проблематика даного підходу полягає у зв'язності лексичних конструкцій між собою та утворення змістовних взаємозв'язків між розпізнаними наборами слів (речень). Для вирішення даної проблеми застосовують додаткові правила, що знижують можливість спотворення результуючого сенсу завдяки порогі чутливості щодо змісту окремих складових повідомлення (вхідної інформації).

Одним із допоміжних засобів, що використовуються в даному підході є попередня обробка текстової інформації за допомогою семантичного аналізу, що визначає ключові слова та надає змогу визначити тематику тексту.

Більшість сучасних рішень автоматичного аналізу тексту застосовують методи та алгоритми, що орієнтовані на морфологічний, фонетичний та синтаксичний рівень аналізу тексту [15]. Кожен з даних рівнів залежить один від одного та в переважній більшості застосовує результати більш нижчого рівня.

Найвищим рівнем аналізу тексту є його логіко-семантичний рівень, що визначає тему, ключові слова та дає змогу експерту оцінити важливість інформації за обраними критеріями [15].

Основою даного аналізу є застосування наборів правил для виокремлення ключових слів за допомогою пунктуаційних знаків, стоп-слів. Особливостями семантичного аналізу тексту є розмір тексту та предметна сфера застосування.

1.2 Аналіз сучасних технологій створення інтелектуальних чат-ботів

Зазвичай, чат-ботів поділяють за призначенням на предметні області, в яких повинен розумітися чат-бот. Даний програмний продукт імітує розмову з людиною в Інтернеті, саме тому даний сервіс найкраще зарекомендував себе саме в соціальних мережах (Facebook Messenger, Telegram тощо). Також варто зазначити, що чат-бот вміщує увесь необхідний потенціал для реалізації штучного інтелекту та його складових. Серед вибору платформ для застосування даний тип може містити крос-платформні рішення на базі персональних та мобільних платформ. Головними перевагами даного типу взаємодії відзначено швидкодію використання, відсутність особистих переваг чи наявність емоцій при контактуванні. Також створені діалоги фіксуються та обробляються на предмет виявлення частот використання окремих запитів та вдосконалення вказаних систем штучної взаємодії шляхом збільшення словникового запасу системи чи вдосконаленню алгоритму розбору слів [16].

Ще один вид даної системи штучної взаємодії базується на встановленні покрокового виконання типу: «Меню вибору опції» – ... – «Отримання результату» [17]. Даний тип встановлює чітке обмеження на введення даних користувачем, але підвищує відсоток успішного використання із-за відсутності помилки введення текстових чи аудіо-даних. У зв'язку з поширеністю даної технології, натеper створені не лише базові прикладні рішення, а й цілі платформи для створення власних інтелектуальних систем штучної взаємодії з можливістю вибору мови, набору алгоритмів обробки інформації, систем фіксування діалогів. Більшість з даних платформ містять власні ресурси для

обробки інформації, що зменшує навантаження на потужності серверів-клієнтів, що використовують дані платформи. Але це призводить в свою чергу до зниження рівня конфіденційності інформації, адже вказана користувачем інформація передається третій стороні (в даному випадку платформі обробки отриманих даних), а це підвищує ризик ураження даних користувача. Прикладом даного випадку є використання чат-ботів для покупок, де користувачу надана можливість введення даних для оплати у вікні чат-бота.

Серед усіх відомих реалізацій, найбільш популярними та успішними є 26 платформ, що мають певні переваги та деякі недоліки, що призводить до вагомої конкуренції на ринку програмного забезпечення [18]. Найбільшою популярністю користуються платформи з наявністю безкоштовного контенту, що дають можливість використання власних рішень за наявності пакетів прикладних мов програмування. Це гарантує надійність створюваних чат-ботів та контроль за обміном даними між користувачем та системою. Для порівняння було використано 3 сучасні платформи від різних розробників, які увібрали у себе майже всі доступні натепер можливості для розвитку: “Microsoft Bot Framework”, “IBM Watson Conversation Service” та “Wit.ai” [19]. (таблиця 1.1).

Таблиця 1.1. Порівняння платформ для створення чат-ботів

Назва платформи	Microsoft Bot Framework	IBM Watson Conversation Service	Wit.ai
Особливості	<ol style="list-style-type: none"> 1. Розуміння намірів користувачів 2. Розпізнання й аналіз голосових повідомлень користувача 3. Використання пошукового агента в мережі Інтернет для відкритого пошуку необхідної інформації 	<ol style="list-style-type: none"> 1. Побудована на нейронній мережі 2. Має три головні компоненти, слова-наміри, об'єкти та діалоги 	<ol style="list-style-type: none"> 1. Використовує слова-наміри, об'єкти, події 2. Використання мови NLP (Natural Language Process)

Програмна реалізація	<ol style="list-style-type: none"> 1. Пакет бібліотек для платформ .NET та Node.js 2. Bot Connector 3. Azure Bot Service 	<ol style="list-style-type: none"> 1. Пакет бібліотек Node SDK 2. Пакет бібліотек Java SDK 3. Пакет бібліотек Python SDK 	<ol style="list-style-type: none"> 1. Пакет бібліотек Node SDK 2. Пакет бібліотек Ruby SDK 3. Пакет бібліотек Python SDK
Ліцензія та доступність	<ol style="list-style-type: none"> 1. Open source 2. Microsoft license 3. Система контролю версій Github 	<ol style="list-style-type: none"> 1. Безкоштовний 2. Стандартний 3. Преміум 	<ol style="list-style-type: none"> 1. Безкоштовна платформа
Доступні мови для використання	<ol style="list-style-type: none"> 1. Автоматичний переклад більш ніж 30 мов, серед яких є і українська 	<ol style="list-style-type: none"> 1. Англійська 2. Японська 	<ol style="list-style-type: none"> 1. Розуміє 50 мов 2. Можливість перекладу обмежена
Канали обробки даних	<ol style="list-style-type: none"> 1. Текст 2. Зображення 3. Відео 4. Соціальні мережі 	<ol style="list-style-type: none"> 1. Текст 2. Зображення 3. Мова 	<ol style="list-style-type: none"> 1. Мова 2. Текст
Сфера використання	<ol style="list-style-type: none"> 1. Високоінтелектуальні боти 2. Пошук інформації 3. Сфера послуг 4. Додатки операційної системи Windows 10 	<ol style="list-style-type: none"> 1. Медицина 2. Фінансова індустрія 3. Сфера послуг 	<ol style="list-style-type: none"> 1. Будь-які додатки для розмов чи текстових повідомлень

1.3 Аналіз об'єкту проектування

1.3.1 Постановка задачі створення предметно-орієнтованих чат-ботів

Сучасні рішення із підтримкою аналізу текстової інформації є досить обмеженими у використанні через різноманіття предметних областей та мов. Також швидкість обробки деяких аналогів є недостатньою для комфортного користування.

Вхідними даними для інформаційної технології створення інтелектуального чат-бота є предметно-орієнтований набір текстів на англійській мові з п'яти предметних областей: кулінарія, туристична подорож, цілодобова служба підтримки, продуктовий магазин та медицина. Довжина

тексту знаходиться у діапазоні від 50 до 600 слів, що відповідає середньо-статистичній вибірці із 5-7 до 25-30 зв'язних між собою речень. Набір даних повинен мати щонайменше 10000 текстів для кожної з предметних областей для прийняттого рівня розпізнавання за допомогою нейронної мережі.

Вихідними даними є модуль семантичного аналізу тексту, тренувана нейронна мережа, що аналізуватиме вхідні повідомлення у вигляді попереднього набору даних та формуванні на їх основі відповіді, що базуватиметься на базі даних, а також модуль спілкування.

Для коректної роботи інформаційної технології створення інтелектуального чат-бота система повинна відповідати наступним вимогам:

- Встановлена бібліотека TensorFlow;
- операційна система сімейства Windows 7\8\10;
- встановлена програмна платформа .NET Core 3.1.

1.3.2 Особливості створення предметно-орієнтованих чат-ботів

Можливість використання елементів штучного інтелекту призвело до стрімкого розвитку алгоритмів обробки інформації та природної мови, як її підвиду. Обробка вхідних даних в інформаційній технології реалізується зазвичай у вигляді спеціально розроблених алгоритмів, технологій машинного навчання, нейронних мереж. Забезпечуючи підтримку користувача під час сеансу роботи із інтелектуальним чат-ботом із підтримкою розпізнавання текстової інформації може мати певні обмеження щодо використання, що виникають при інтеграції в інтелектуальну інформаційну систему (ІС). Для цього інформаційна технологія потребує надійну та висококваліфіковану обробку вхідних даних. Забезпечення даної обробки даних полягає у наданні інструментів інтелектуального аналізу даних, алгоритмів обробки великих масивів даних (Big Data), а також моніторингу процесу обробки для покращення аналізу та зменшення похибки розпізнавання. Застосовуючи усі вищеперераховані засоби та функції, більшість сучасних рішень із підтримкою

розпізнавання природної мови мають досить високий функціональний потенціал, розпізнають до 99% відсотків вхідної інформації в обраних предметних областях. Найбільш вдалим рішенням у даному напрямку – чат-бот, що використовується у сферах підтримки користувача та системах прийняття рішень. Якщо розглядати чат-бота як цілком автономну інтелектуальну інформаційну систему, постає проблема застосування для широкого кола задач та сумісність із стороннім програмним забезпеченням. Ще одним із вагомих аспектів не лише чат-ботів, а й ПС в цілому є підтримка актуальності сховища використовуваних даних. У даному випадку чат-боти поділяються на три типи: з можливістю навчання на отриманих вхідних даних – даний тип передбачає, що при першому запуску, чат-бот мав доступ до актуальних даних у сховищі, а потім поступово автоматично замінював низько ціннісні дані на отримані з інформації з вищим показником цінності; з можливістю автоматичного оновлення сховища даних на зразок великого словника чи постійно активного джерела актуальних даних – даний тип забезпечує себе актуальними даними за рахунок регулярного оновлення сховища даних у регламентованому режимі за обраним розкладом; з необхідністю механічного оновлення сховища даних розробником ПС або користувачем, якщо така опція представлена в ПС.

При виборі типу чат-бота, що буде створюватись завдяки інформаційній технології для більш глобальної ПС, основна увага приділяється предметній області в якій повинен розумітися чат-бот. Відповідно до цього ступінь розуміння та складність імплементації сховища даних прямо-пропорційно залежить від термінологічної бази предметної області. При поєднанні декількох предметних областей або виборі міжгалузевої предметної області, ступінь розуміння значно понижується та підвищується складність підтримки актуальності сховища даних чат-бота. Застосування декількох сховищ одночасно або паралельно є досить перспективним рішенням даної проблеми, адже забезпечує поділ термінологічної бази на менші сукупності, що легше супроводжувати та надавати на вхідні запити користувача інформаційної технології створення інтелектуального чат-бота. Але незважаючи на переваги,

ключовим недоліком даного способу є підвищення складності обробки та видачі інформації користувачу із-за можливих конфліктів між сховищами даних. При виборі чат-бота значною мірою впливу на його характеристики є незалежність від інших ІС та можливість автоматичного оновлення сховища актуальними даними. Базуючись на даних особливостях, обробка даних такої інформаційної технології потребує детального аналізу та використання надійних інструментів та засобів обробки даних.

Розглянувши інтелектуального чат-бота як продукт інформаційної технології із суто-технічного боку, слід зазначити, що ключовими критеріями, що повинні бути досягнуті в результаті проведеного дослідження є:

1. Розробка прототипу чат-бота із втіленням визначених ідей
2. Автономність чат-бота та створення можливості інтеграції в більш глобальну ІС
3. Надання можливості підключення декількох сховищ даних, які можуть бути використані одночасно
4. Створення ефективної обробки даних, за рахунок використання інструментів та засобів інтелектуального аналізу даних, нейронних мереж, алгоритмів обробки даних та інших методів.

Щодо функціональних можливостей інтелектуального чат-бота, то вони можуть включати в себе підтримку користувача ІС, можливості сторонніх сервісів, що використовуються чат-ботом, а також інтелектуальний аналіз дій користувача. Підтримка користувача полягає у обміні пакетами інформації між користувачем та ІС у вигляді текстових, графічних, аудіо та відео-повідомлень. Кожен з даних типів пакетів інформації повинен містити власний апарат аналізу вхідних даних ІС та бути незалежним від аналізу інших типів даних у створеній ІС. Відповідно до мінімальних вимог функціонування інтелектуального чат-бота, система повинна адекватно та своєчасно розпізнавати тип інформаційного пакету, що надходить від користувача. Найбільш примітивним рівнем в даному випадку є чат-бот із використанням команд. В даному випадку, обробка базується на виконанні закладених у його сховище даних команд без

використання складних алгоритмів та технологій. Більш складним вважається ПС із можливістю розпізнавання одного типу інформації. До даної категорії, належать ПС чат-боти із можливістю розпізнавання тексту, зображень чи аудіо-файлів. Чат-боти, що належать до даної категорії є найбільш поширеними натеper та можуть створюватись на спеціальних платформах у мережі Інтернет чи за допомогою програмування. Досить незначною є категорія чат-ботів, які поєднують у собі можливості розпізнавання декількох типів інформації та надавати користувачу ПС відповідь, також поєднуючи декілька типів інформації. Чат-боти даної категорії є найбільш складними у розробці та супроводі ПС, причому їхнє створення можливе лише висококваліфікованими розробниками програмного забезпечення. Розпізнавання інформації у інтелектуального чат-бота, поділяється на декілька етапів:

1. Визначення типу інформації
2. Розподіл отриманої від користувача інформації на окремі блоки, що можуть оброблюватись системою
3. Виконання семантичного аналізу текстової інформації
4. Виконання певних операцій перетворення даних для знаходження відповідностей та відомих чи подібних ПС зв'язків між блоками вхідної інформації, використання нейронної мережі
5. Вибір найбільш відповідної інформації у сховищах даних чат-бота
6. Перерахунок цінності сховищ даних

Оцінюючи кожен з даних етапів за складністю реалізації та важливістю коректного виконання, до складних варто віднести виконання перетворень над даними та вибір найкращого варіанту поєднання даних сховища інтелектуального чат-бота. Саме дані етапи, найбільш часто підлягають інтелектуалізації та використанню елементів штучного інтелекту. Основними на теперішній час можливостями для виконання даних етапів є використання технологій машинного навчання, комп'ютерного бачення, нейронних мереж, нейронних мереж із використанням глибокого навчання. Більшість з даних технологій та напрямків безпосередньо належать до технологій штучного

інтелекту, що гарантують усі переваги інтелектуальної обробки даних та забезпечують достатній рівень обробки вхідної інформації.

Ще одним із важливих критеріїв роботи чат-бота є інформаційне середовище у якому він працює. За даним критерієм, вони поділяються на відкриті, що передбачає використання декількома користувачами одночасно, та закриті, що можуть бути використані лише одним користувачем незалежно від тривалості сеансу.

Можливість одночасної роботи інтелектуального чат-бота одразу із декількома користувачами одночасно є найбільш складним етапом реалізації інформаційної технології. У ньому зазвичай використовуються технології паралельних обчислень та розподілених систем. Формат сеансу у даного типу чат-ботів відсутній. Фіксування тривалості роботи користувача із чат-ботом зазвичай аналізується тривалістю перерв між поданням вхідної інформації користувачем. Контролюючи процеси обміну інформацією, формуються звіти та метадані. Аналізуючи дані звіти можливо зробити висновок про рівень обробки інформації, невизначені ситуації, коли найбільш відповідна інформації у сховищі даних інформаційної технології не перевищує допустимий поріг та інші форми роботи чат-бота, створеного завдяки інформаційній технології. Невизначені ситуації виникають у випадку недостатнього розміру сховища чи його низькій інформаційній цінності, а також у випадку, коли тип вхідної інформації несумісний із можливими для розпізнавання обраною інформаційною технологією аналогом. Але найбільш частою проблемою виникнення невизначених ситуацій є суттєва відмінність вхідної інформації від термінологічної бази обраної предметної області чи областей. Спостерігаючи за цими випадками, можливо зробити висновок про необхідність імплементації додаткових функцій.

Проаналізувавши чат-бота як продукт інформаційної технології із функціональної точки зору, ключовими критеріями, що повинні бути виконані наприкінці дослідження, було обрано:

1. Проектування, розробка та імплементації моделі чат-бота, що повинна бути реалізована в інформаційній технології.
2. Виконання адекватного розпізнавання текстової інформації для декількох предметних областей.
3. Створення чи використання сторонніх інструментів та засобів для контролю та аналізу роботи інформаційної технології створення інтелектуального чат-бота.
4. Автоматичне формування звітності щодо роботи інформаційної технології.
5. Вирішення проблеми невизначених ситуацій.

1.3.3 Характеристика та аналіз систем-аналогів

Натепер існує достатня кількість чат-ботів та платформ для їх створення, що призводить до конкуренції. Серед найбільш відомих реалізацій, які обирались в якості аналогів варто зазначити: чат-бот, створений на базі фреймворку Microsoft Bot Framework із застосуванням сервісу Google DialogFlow [20] та його модифікації із застосуванням додаткових бібліотек для обробки не лише текстової, але й аудіо чи відео інформації. Даний тип чат-бота є найбільш універсальним через можливість модифікації та наявності містких інтерфейсів для реалізації власних опцій та налаштувань чат-бота. Ключовими характеристиками, які притаманні даному чат-боту, є веб-орієнтованість та якість обробки даних. Для інтелектуального аналізу текстової інформації даний тип чат-бота використовує нейронні мережі та технології інтелектуального аналізу даних. Незважаючи на відкритість Microsoft Bot Framework, Google DialogFlow не надає можливості ідентифікації типу нейронної мережі, але для даної роботи важливою є лише швидкодія обробки та залежність впливу об'єму текстової інформації на час-обробки.

Базові характеристики даного типу чат-боту у порівнянні із чат-ботом, створеним на базі інформаційної технології наведені в таблиці 1.2.

Таблиця 1.2 – Порівняльний аналіз основних характеристик чат-ботів

	Чат-бот, створений на базі інформаційної технології	Чат-бот із використанням Microsoft Bot Framework та Google DialogFlow
Кількість предметних областей	до 4	1
Наявність зовнішнього контролю за чат-ботом	існує	відсутній
Використання проміжних відповідей користувача	інтегровано у якості опції налаштування	відсутнє
Формування статистики	існує	існує
Можливість зміни предметної області чат-бота	існує	відсутнє
Інтеграція з соціальними мережами	Skype, Telegram, Facebook, Twitter	понад 16

1.4 Висновок

Під час проведення аналізу предметної області аналізу текстової інформації визначено засоби для реалізації поставленої задачі, технології штучного інтелекту у вигляді нейронних мереж та аналоги інформаційних технологій, що мають підтримку розпізнавання природної мови, а також удосконалений метод семантичного аналізу тексту. Серед даних інформаційних, було проаналізовано моделі та види чат-ботів, що містять необхідний потенціал для реалізації обробки природної мови у вигляді програмного модуля.

Було розглянуто сучасні технології для підвищення якості аналізу текстової інформації. В результаті досліджень було відзначено переваги нейронних мереж перед технологією машинного навчання. Оскільки обробка текстової інформації полягає не лише у розпізнаванні, а й у наданні прийнятної відповіді за мінімальний проміжок часу, було обрано двох-ступеневу обробку за допомогою семантичного аналізу в якості попередньої обробки та рекурентну нейронну мережу із застосуванням методу довготривалої короткочасної пам'яті для інтелектуального аналізу текстової інформації.

В ході аналізу об'єкту проектування визначено основні вимоги до інформаційної технології створення інтелектуального чат-бота. Був проведений аналіз систем аналогів у вигляді платформ для створення чат-ботів: Microsoft Bot Framework, IBM Watson Conversation Service, Wit.ai, визначено їх можливості, переваги та недоліки.

Отже в даному розділі магістерської кваліфікаційної роботи визначено основні проблеми які виникають при обробці текстової інформації, технології за допомогою яких їх можна усунути та вимоги до інформаційної технології інтелектуального чат-бота.

2 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНИХ ЧАТ-БОТІВ

2.1 Аналіз особливостей вхідної інформації та обґрунтування вибору засобів інформаційної технології

Охарактеризувавши інформаційну технологію створення інтелектуального чат-бота за технічними та функціональними якостями розглянуто найбільш притаманні натеper типи та моделі чат-ботів. Кожен з них має власні переваги та недоліки для різних поставлених задач. Гарантуючи наявність усіх компонентів інтелектуального чат-бота, якісна характеристика його складових (сховища даних, інструменти аналізу сеансів і т. д.) відіграє першочергове значення. Саме тому, вибір складових був сформований на основі порівняння найкращих рішень, що представлені натеper на ринку програмного забезпечення. Оскільки одним із критеріїв, що має виконувати інформаційна технологія створення інтелектуального чат-бота є аналіз вхідної інформації декількох типів, категорія створюваного чат-бота належить до найбільш складної, що може проектуватись та розроблюватись лише із використанням програмних бібліотек та мов програмування високого рівня. Серед відомих натеper типів комп'ютерної інформації існує інформація у вигляді символів (азбука Морзе, шифри), тексту (повідомлення, статті), графічних об'єктів (зображення), звуку (аудіо-файли, мелодії), відео (фільми). Для розпізнавання символів достатньо використання відповідних алгоритмів для дешифрування чи аналізу і не потребує значною мірою використання елементів штучного інтелекту. Впровадження розпізнавання даного типу інформації є недоцільним та призводить до підвищення загальної складності реалізації та проектування ІС чат-бота. Текстова інформація є найбільш поширеною в якості предмету інформаційної взаємодії між ІС та користувачами. Текстова інформація має найбільшу кількість інструментів та засобів щодо аналізу та визначення рівня інтелектуального змісту. Вона включає в себе увесь спектр необхідних для

користувача опцій та функцій, за допомогою яких відбувається керування даними та компонентами обраних ПС та ПС чат-бота. Для успішного розпізнавання тексту найкращими інструментами та технологіями себе зарекомендували технології машинного навчання та нейронні мережі. Порівнюючи дані технології варто зазначити, що обидві використовують елементи штучного інтелекту та мають приблизно однакову швидкість, але відмінним є те, що нейронні мережі при коректному обранні кількості нейронів, епох, розміру тренувальної вибірки здатні розпізнавати інформацію до 99,9% на відміну від машинного навчання, де цей показник має досить широкий діапазон можливих значень та безпосередньо залежить від розміру сховища даних інтелектуального чат-бота [8]. Інформація графічного типу, переважно зображення, також є однією з найбільш популярних у інформаційному середовищі та має достатньо засобів для обробки та розпізнавання, що необхідно для інтелектуального чат-бота. Найбільш поширеними технологіями для обробки даного типу інформації, як і для текстового типу є машинне навчання та нейронні мережі, а також технологія комп'ютерного бачення. Комп'ютерне бачення досить перспективний напрямок, що використовується для класифікації та розпізнавання зображень. Інформація у вигляді звукозаписів та аудіо-файлів є менш використовуваною, але також досить перспективною у використанні ПС чат-бота та її розпізнавання. Для цього також досить широко використовуються алгоритми розбору звукозаписів на частини, машинне навчання, нейронні мережі. Відеоінформація є найбільш об'ємною та може містити одразу декілька типів інформації (зображення і текст, звукозапис та зображення), що призводить до значного підвищення складності до задач розпізнавання та класифікації, що вирішуються нейронними мережами та нейронними мережами із використанням технології глибинного навчання.

Серед розглянутих типів, розпізнавання текстової та графічної інформації є найбільш спорідненими функціями, що застосовуються в сучасних ПС чат-ботів. Виходячи з цього, було обрано розпізнавання текстової інформації як основного типу інформації. Розпізнавання зображень є додатковою задачею із-

за складності поєднання компонентів для розпізнавання різних типів інформації. Для обробки текстових даних найбільш поширеними є засоби інтелектуального аналізу даних, згорткові нейронні мережі, рекурентні нейронні мережі. Завдяки подібності елементів тексту, рівень аналізу слів та їхніх конструкцій майже повністю повинен відповідати рівню аналізу тексту людиною, адже кожен атомарний елемент текстової інформації представляє собою слово чи знак пунктуації. Але в ряд причин, пов'язаних із лексичними та граматичними особливостями текстової інформації рівень інтелектуального аналізу текстової інформації на даний момент не відповідає бажаному рівню, що змушує продовжувати пошуки алгоритмів, методів та засобів для покращення розбору природної мови. В нашому випадку інформаційна технологія створення інтелектуального чат-бота повинна бути багатомовною, тобто обмежень текстової інформації за мовою не передбачено, через можливості зміни сховища даних.

2.2 Аналіз методів розпізнавання та аналізу текстової інформації для декількох предметних областей

Обробка текстової інформації поділяється на декілька типів, починаючи від семантичного, і завершуючи інтелектуальним аналізом. Відповідно комплексний аналіз надає змогу отримати цілісне розуміння щодо вхідної інформації від користувача. Для цього розглянемо ланцюг обробки тексту, що буде реалізовано в інформаційній технології створення інтелектуального чат-бота.

1. Розбиття тексту на компоненти (речення)
2. Семантичний аналіз тексту (формування семантичного ядра із ключових слів)
3. Аналіз попередніх результатів із семантичним ядром
4. Інтелектуальний аналіз семантичного ядра

5. Аналіз результатів інтелектуального аналізу та застосування сторонніх сервісів.
6. Об'єднання результатів усіх етапів та формування відповіді

Перші два етапи є взаємопов'язаними та можуть бути об'єднані в один модуль. Семантичний аналіз тексту полягає у формуванні набору ключових слів із частотою їх використання. Для цього необхідно виділити три групи слів, які будуть виключені із списку слів для семантичного аналізу: стоп-слова, допоміжні слова та низько-частотні слова.

Істотно зауважити, що починаючи з першого етапу та до останнього, кількість інформації, придатної для обробки зменшується, завдяки чому було проаналізовано можливість удосконалення семантичного аналізу тексту та проведення порівняльного аналізу звичайного семантичного аналізу із удосконаленим, що увібрав у себе набір правил для фільтрування низькочастотних слів та стоп-слів, що зазвичай складають до понад 40% від загального набору ключових слів, а також часову характеристику залежності об'єму вхідного тексту до розміру семантичного ядра (набору ключових слів) [3].

Семантичний аналіз є нечутливим до предметної області та формує лише структурований набір слів, що ідентифікують тематику тексту.

Для порівняльного аналізу було використано декілька статей [3], а результати оформлено у вигляді таблиці 2.1.

Таблиця 2.1 – Порівняльна характеристика звичайного та удосконаленого методу семантичного аналізу тексту.

Стаття	1	2	3	4	5
Частота появи ключових слів (звичайний)	75.89%	80.09%	81.6%	71.2%	74.58%
Частота появи ключових слів (удосконалений)	45.89%	57.47%	54.73%	36%	30.31%
Час виконання аналізу (звичайний), с	0.26	0.28	0.46	0.25	0.32
Час виконання аналізу (удосконалений), с	0.31	0.35	0.71	0.62	0.54

Кількість ключових слів / загальний обсяг тексту (звичайний):	370	389	383	342	378
Кількість ключових слів / загальний обсяг тексту (удосконалений):	218	273	260	171	144

Отримані результати свідчать про значну надлишковість обсягу семантичного ядра вхідної текстової інформації, що в подальшому спричинить подовження інтелектуального аналізу тексту та може призвести до неоднозначного результату через наявність схожих синтаксичних конструкцій.

Отже, використання удосконаленого методу семантичного аналізу тексту є виправданим та обґрунтованим у зв'язку із зменшенням розміру семантичного ядра, а відповідно вхідних даних для інтелектуального аналізу тексту, що в свою чергу пришвидшує його обробку.

Для подальшої обробки семантичне ядро може бути використане як індикатор зміни тематики тексту у порівнянні із результатами попередніх обробок для поточного сеансу користувача чат-бота. Для цього вже існуючі відповіді та семантичні ядра попередніх запитів користувача порівнюються з поточним. Причому у разі різниці між наповненням семантичних ядер на рівні 50% та вище вже варто відкинути попередні результати та використовувати лише поточне семантичне ядро. Якщо ж поточне семантичне ядро містить понад 50% від раніше отриманого вхідного тексту, то попередні результати лише пришвидшують подальший інтелектуальний аналіз семантичного ядра у зв'язку із аналізом лише нової частини, що не була раніше зафіксована. Таким чином досягається максимальна ефективність інтелектуальної обробки тексту, а навантаження на інтелектуальний аналіз мінімізується.

Після формування вхідного набору ключових слів, інтелектуальний аналіз полягає у визначенні тематики тексту, предметної області та набору ідентифікаторів, що дозволять в подальшому здійснити пошук у термінологічних базах даних для формування відповіді на запит користувача. В

загальному даний етап є найбільш складним та може бути умовно поділений на дві послідовні частини: визначення тематики та формування відповіді.

Перша частина є окремою задачею розпізнавання та класифікації тексту, що потребує використання інтелектуальних технологій обробки текстових даних.

Для вирішення даної задачі, розглянемо основні рішення та визначимо найкраще для даного випадку. Серед інтелектуальних технологій обробки тексту для задачі розпізнавання та класифікації тексту виділяють нейронні мережі, методи автоматичного аналізу тексту, алгоритми машинного навчання та дерева рішень. Щодо автоматичного аналізу тексту варто зазначити, що він базується на наборі правил, що потребують постійного оновлення при зміні предметних областей, відповідно дане рішення не є доцільним для застосування у інформаційній технології. Алгоритми машинного навчання для задачі розпізнавання та класифікації тексту потребують наявності значної вибірки даних та експерта, який вказуватиме на якість класифікації.

Дерева рішень є ще одним способом виконання класифікації. Дерево рішень – це класифікатор, який представлений як ієрархічне розбиття простору даних. Структура дерева містить 2 типи вузлів: листовий вузол (містить значення цільового атрибута, тобто позитивна або негативна мітка в задачі бінарної класифікації) та вузол вирішення (містить умову для одного з атрибутів розділу пробілів). Розбиття простору даних здійснюється рекурсією. Складністю їх використання в даному випадку є відсутність стійкості до розміру вхідних даних [21].

Нейронні мережі в даному випадку є найкращим засобом для задачі розпізнавання та класифікації тексту, оскільки не потребує експерта для визначення тематики, а ґрунтується на розмірі вибірки та її вмісту. Серед існуючих натеper типів нейронних мереж та їх видів, для задачі розпізнавання та класифікації тексту застосовують рекурентні та згорткові нейронні мережі. Порівняльний аналіз різних типів нейронних мереж наведено у таблиці 2.2.

Таблиця 2.2 – Порівняльний аналіз нейронних мереж для задачі класифікації тексту

Назва*	DNN	RNN	RNN + LSTM	CNN
Середній час обробки**, с	0,4538	0,4804	0,3992	0,8257
Можливість зворотного поширення	відсутня	існує	існує	відсутня
Стійкість до часових прогалин	не стійка	не стійка	стійка	не стійка
Залежність від розміру вхідних даних	залежна	незалежна	незалежна	залежна

*, середній час обробки тексту довжиною до 150-200 слів;

**, DNN – глибинна нейронна мережа, RNN – рекурентна нейронна мережа, RNN+LSTM – рекурентна нейронна мережа із використанням алгоритму довгої короткочасної пам'яті, CNN – згортова нейронна мережа.

Щодо методів навчання нейронних мереж, то існують два методи: навчання без вчителя та з вчителем.

Навчання без вчителя використовує не позначені раніше набори даних, щоб виявити структуру та знайти схожі патерни з вхідних даних. Використовується для задачі кластеризації.

Навчання з вчителем передбачає формування набору даних, що використовується для тренування нейронної мережі із попередньо-визначеною вибіркою характеристик.

Рекурентна нейронна мережа (РНМ) за період часу від її створення до сьогодення, зазнала великої кількості модифікувань та методів, що призвело до цілого набору варіантів нейронної мережі для кожного з поставлених завдань. Найбільш відомими РНМ є мережі Елмана, Джордана, мережа з відлунням стану, мережа із використанням методу довгої короткочасної пам'яті LSTM, реверсивна РНМ, РНМ неперервного часу, ієрархічна РНМ, РНМ другого

порядку, РНМ кількох масштабів часу, нейронні машини Тюрінга. Серед представлених модифікацій РНМ, найбільшу увагу було зосереджено на використанні алгоритму довгої короткочасної пам'яті, що найкраще себе зарекомендувала саме у задачах розпізнавання текстової інформації. Розроблено та опубліковано цей алгоритм було у 1997 році Хохрайтером та Шмідгубером [22]. Даний алгоритм дозволяє уникнути проблеми зникання градієнту та запобігає зниканню та виникненню зворотно поширюваних похибок. Це відбувається за рахунок зворотного поширення крізь необмежене число шарів розгорнутої у просторі РНМ. Базуючись на зворотному поширенні крізь довільне число шарів, РНМ із використанням алгоритму довгої короткочасної пам'яті стійка до часових прогалин та не чутлива до довжини вхідної текстової інформації. Завдяки можливостям такої РНМ, найбільш широке її застосування знаходиться у сфері розпізнавання текстової інформації, де дана модель РНМ почала перевершувати традиційні моделі розпізнавання. Прикладами її дієвості є використання пошуковим гігантом Baidu з 2014 року, використання в Google Android та Google Voice Search [23]. Завдяки потужності даної РНМ, її успішно застосовують для розпізнавання та моделювання мов та багатомовної обробки.

Розглядаючи модель РНМ із використанням алгоритму LSTM як модифікацію звичайної РНМ, в якості вузлів виступають LSTM елементи, що представляють собою найпростішу РНМ із можливістю зворотного поширення. Методом порівняння результатів, отриманих експериментального для вхідної вибірки текстової інформації, було визначено РНМ із набором характеристик: 64 нейрони вхідного шару, 32 вихідних нейронів, 5 прихованих шарів по 1024 нейрони у кожному. Функція активації нейрона – сигмоїдальна, та кількість епох для тренування – 2000.

За результатами проведеного аналізу було обрано рекурентну нейронну мережу із використанням алгоритму довгої короткочасної пам'яті (LSTM). Основними перевагами її використання є незалежність від розміру вхідних даних, а також можливість зворотного поширення, що дозволяє формувати логічні ланцюги слів та визначати тематику тексту за найменший час.

Друга частина є окремою задачею кластеризації текстових даних для визначення відповіді, що також потребує використання інтелектуальних технологій обробки тексту.

Порівнюючи розвиток обробки тексту із розвитком штучного інтелекту варто підкреслити спорідненість напрямків та безпосереднє застосування технологій штучного інтелекту. Вибір інформаційно-ціннісних елементів із потоку інформації формується за рахунок пошуку серед інформації конструкцій, що вже визначені сховищем даних чат-бота як конструкції, що належать до термінологічної бази обраної предметної області чи областей. Описаний пошук здійснюється за рахунок використання інтелектуальних засобів аналізу інформації. Для задачі кластеризації семантичного ядра було розглянуто два найбільш поширені типи нейронної мережі. Перший тип – згорткові нейронні мережі, особливість яких полягає у мінімальній попередній обробці даних перед використанням. Недоліком даного типу є складність навчання у порівнянні із рекурентною нейронною мережею. Другий тип – рекурентні нейронні мережі, що також належать до класу штучних нейронних мереж та до задач яких входить класифікація текстової інформації. На відміну від згорткових нейронних мереж прямого поширення, з'єднання між вузлами у рекурентних нейронних мереж утворює орієнтований цикл [8]. Це створює внутрішній стан мережі, що дозволяє їй проявити динамічну поведінку у часі, на базі якої формується внутрішня пам'ять. Завдяки ефекту внутрішньої пам'яті, рекурентна нейронна мережа є стійкою до динамічної зміни вхідного потоку інформації та для обробки довільних послідовностей входів. Третій тип – мережа Кохонена, яка містить два шари, та завдяки вхідний та вихідний, що мають назву карти мережі [24]. Під час формування кластерів, усі дані є апріорно невідомими, але містять набір міток, що визначають клас, до якого вони належать, в даному випадку це предметна область та її терми. Незважаючи на простоту реалізації, її застосування є виправданим у зв'язку із швидкістю обробки даних. Дана мережа використовує навчання без вчителя. Ключовою особливістю вибору мережі Кохонена є швидкість навчання, адже на кожен запит користувача необхідно брати за основу

базові тренувані ваги та вдосконалювати вагову матрицю, що в результаті надає можливість заново обирати відповідь з термінологічних баз даних, у зв'язку з цим, швидкість навчання рекурентної та згорткової нейронних мереж значно довший, що не задовольняє умову підвищення швидкодії обробки текстової інформації для декількох предметних областей. Причому для кожної предметної області існує своя власна мережа Кохонена. Результатом її використання є отримання вектору індексованих значень, що будуть використані для отримання відповідних термінів з термінологічних баз даних, пов'язаних з даною предметною областю.

Завдяки цьому було вирішено використовувати мережу Кохонена для задачі кластеризації. Вихідний шар мережі Кохонена розташовується зазвичай у двовимірному просторі (рис. 2.1.а, б).

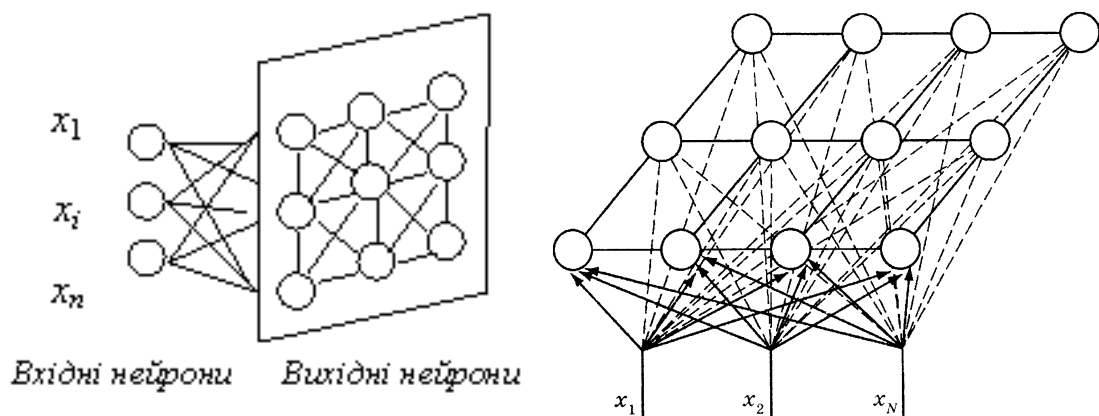


Рисунок 2.1 а, б – Двовірна структура мережі Кохонена

Мережа Кохонена для навчання використовує метод послідовних наближень. Починаючи з випадковим чином обраного вихідного розташування центрів, алгоритм поступово покращується для кластеризації навчальних даних [24], В якості навчальних даних використовуються терміни термінологічних баз даних.

Вхідним набором даних є вектор слів, що послідовно надаються мережі. Після проходження усіх тестових векторів та перерахунку вагової матриці, визначаються кластери, що являють вектор індексів, по яким є змога отримати

відповідь на запит користувача. В даному випадку, ваги організуються у відповідності з тим, що сусідні вузли-нейрони чуттєві до схожих вхідних сигналів.

Для реалізації алгоритму необхідно визначити міру сусідства нейронів (окіл нейронів-переможців). На рисунку 2.2 представлені зони топологічного сусідства нейронів на карті ознак у різні моменти часу. $NE_j(t)$ - множина нейронів, що вважаються сусідами нейрона j у момент часу t . Зони сусідства зменшуються з часом.

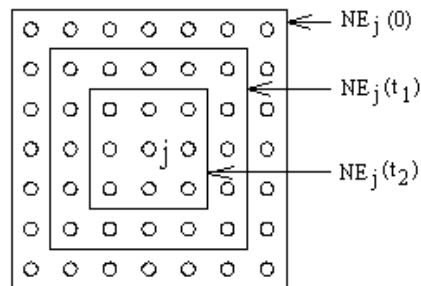


Рис. 2.2 - Зони топологічного сусідства на карті ознак у різні моменти часу

Потенційними функціями сусідства є:

а) дзвоноподібна функція Гауса, яка обчислюється за формулою 2.1

$$f_{gauss1}(d, \sigma) = e^{-\frac{d^2}{2\sigma^2}}, \quad (2.1)$$

де σ^2 - дисперсія відхилення;

б) функція «мексиканський капелюх», що обчислюється за формулою

2.2

$$f_{gauss2}(d, \sigma) = \left(1 - \left(\frac{d}{\sigma}\right)^2\right) \cdot e^{-\left(\frac{d}{\sigma}\right)^2}, \quad (2.2)$$

в) косинусоїдна функція, яка обчислюється за формулою 2.3

$$f \begin{cases} \cos\left(\frac{d\pi}{2\sigma}\right) \\ 0, d \geq \sigma;_{cos} \end{cases} \quad (2.3)$$

г) конусоподібна функція, яка обчислюється за формулою 2.4

$$f_{cone}(d, \sigma) = \begin{cases} 1 - \frac{d}{\sigma}, d < \sigma; \\ 0, d \geq \sigma; \end{cases} \quad (2.4)$$

д) циліндрична функція, яка обчислюється за формулою 2.5

$$f_{cylinder}(d, \sigma) = \begin{cases} 1, & d < \sigma; \\ 0, & d \geq \sigma; \end{cases} \quad (2.5)$$

е) функція «французький капелюх», яка обчислюється за формулою 2.6

$$h(p) = \begin{cases} 1, & |p| \leq \alpha \\ -\frac{1}{3}, & \alpha < |p| \leq 3\alpha \\ 0, & |p| > 3\alpha \end{cases} \quad (2.6)$$

Для функціонування даної частини аналізу текстових даних проведено дослідження ефективності роботи з використанням різних функцій сусідства.

Для даного дослідження було створено окрему термінологічну базу даних, що включала в себе терміни з предметної області туристичної агенції.

При перевірці ефективності функцій-сусідства визначалась якість навчання, що визначається за формулою 2.7:

$$Q = \frac{X}{X+Y} \quad (2.7)$$

де, X – це кількість задовільних відповідей на вхідний вектор, Y – кількість незадовільних відповідей на вхідний вектор.

1) Функція Гауса (формула 2.8);

$$F(p, t) = \exp\left(-\frac{p^2}{2\sigma^2(t)}\right) \quad (2.8)$$

2) Косинусоїдна функція (формула 2.9);

$$F_{cone}(p, t) = \begin{cases} 1 - \frac{p}{t}, & p < t; \\ 0, & p \geq t; \end{cases} \quad (2.9)$$

2) Функція «Мексиканський капелюх» (формула 2.10);

$$F(p, t) = \exp\left(-\frac{p^2}{\sigma^2(t)}\right)\left(1 - \frac{2}{\sigma^2(t)}p^2\right) \quad (2.10)$$

3) Функція «Французький капелюх» (формула 2.11);

$$h(p) = \begin{cases} 1, & |p| \leq \alpha \\ -\frac{1}{3}, & \alpha < |p| \leq 3\alpha \\ 0, & |p| > 3\alpha \end{cases} \quad (2.11)$$

Для вибору найефективнішої з представлених функцій сусідства, було проведено порівняння їх точності на основі вибірки у 500 вхідних векторів, де

300 векторів надається навчальній вибірці, а решта 200 – тестовій вибірці; $\varepsilon=10^{-4}$. Результати порівняння наведені у таблиці 2.3.

Таблиця 2.3 – Функції сусідства та досягнена достовірність діагностики на тестовій вибірці

Функція сусідства	Кількість правильно розпізнаних вхідних векторів	Кількість неправильно розпізнаних вхідних векторів	Якість навчання
Функція Гауса	155	45	83%
Косинусоїдна функція	182	18	91%
Функція «Мексиканський капелюх»	169	31	84,5%
Функція «Французький капелюх»	171	29	85,5%

Отже, для подальшої реалізації обрана функція косинусоїдна функція, що надає найбільшу точність кластеризації.

Алгоритм функціонування мережі Кохонена:

1. Ініціалізація мережі. Ваговим коефіцієнтам мережі надаються малі випадкові значення.

2. Подання на вхід мережі нового вхідного вектору.

3. Визначення відстані до всіх нейронів мережі:

Відстані d_j від вхідного сигналу до кожного нейрона j визначаються за формулою 2.12.

$$d_j = \sum_{i=1}^N (x_i(t) \cdot w_{ij}(t))^2 \quad (2.12)$$

де x_i - i -ий елемент вхідного сигналу в момент часу t , $w_{ij}(t)$ - вага зв'язку від i -го елемента вхідного сигналу до нейрона j у момент часу t . У зв'язку з тим, що один нейрон надає змогу ідентифікувати лише один термін в термінологічних база даних, тому відстань до кожного нейрона обраховується та зберігається окремо. В результаті обробки усіх значень вектору, буде отримано масив у

вигляді відстаней до усіх нейронів, з якого буде в подальшому обиратись найменша відстань кожного нейрона до вектору вхідних значень.

4. Налаштування ваг нейрона j^* і його сусідів (рис. 2.3):

Робиться налаштування ваг для нейрона j^* і всіх нейронів з його околу. Нові значення ваг обчислюються за формулою 2.13.

$$W = w_{ij}(t + 1) = w_{ij}(t) + r(t)(x_i(t) - w_{ij}(t)) \quad (2.13)$$

де $r(t)$ - швидкість навчання, що зменшується з часом (додатне число, менше одиниці).

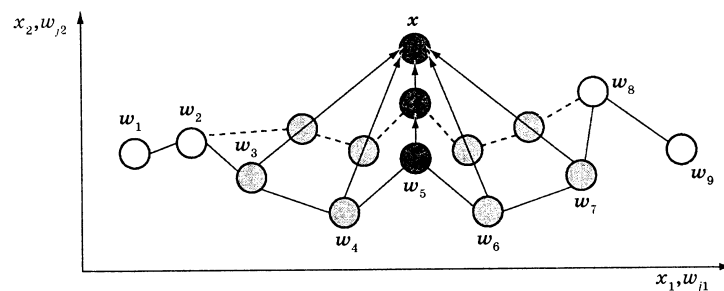


Рисунок 2.3 – Зміна ваг мапи Кохонена

В алгоритмі використовується коефіцієнт швидкості навчання, який поступово зменшується, для більш точної корекції на новій епосі.

Властивість топологічної впорядкованості досягається в алгоритмі за допомогою використання поняття околу. Окіл (радіус сусідства) - це декілька нейронів, що оточують нейрон-переможець [24]. Відповідно до швидкості навчання, розмір околу поступово зменшується, так, що спочатку до нього належить досить велике число нейронів (можливо вся карта), на самих останніх етапах окіл стає нульовим і складається лише з нейрона-переможця. Для вхідного набору даних у вигляді вектору текстових значень, визначається нейрон-переможець для кожного із значень вектору, а вихідний шар містить вектор значень, що ідентифікуватиме набір термінів, що буде використаний при подальшому пошуку в термінологічних базах даних предметних областей. В алгоритмі навчання корекція застосовується не тільки до нейрона-переможця, але і до всіх нейронів з його поточного околу. В результаті такої зміни околу, початкові доволі великі ділянки мережі мігрують в бік навчальних прикладів. З

кжною новою епохою швидкість навчання і розмір околів зменшується, тим самим всередині ділянок карти виявляються більш тонкі розходження, що зрештою призводить до точнішого налаштування кожного нейрона.

При використанні декількох термінологічних баз, необов'язково споріднених предметних областей, виникає необхідність введення порогу чутливості. Адже при використанні декількох сховищ даних одночасно для вибору найкращого результату, кожне сховище надаватиме власну найкращу відповідь із відповідним коефіцієнтом розпізнання. У цьому випадку, відповідь сховища даних, що не пов'язано із предметної областю сеансу, повинна бути відхилена, як недоречна. Саме для цього і вводиться даний поріг, що має на меті відхиляти подібні відповіді та уникати таких ситуації взагалі.

2.3 Аналіз та обґрунтування вибору зовнішніх інформаційних технологій при реалізації інтелектуальних чат-ботів

При визначенні основних проблем та недоліків, що постають перед сучасними рішеннями ІС чат-ботів відзначено проблему, коли користувач використовує мовленнєві терміни та засоби, що не належать до термінологічної бази сховища ІС чат-бота або навіть до споріднених із нею термінологічних баз предметних областей. В даному випадку, аналітичні та інтелектуальні можливості чат-бота мінімальні, а результат більшою мірою є невдалий або несумісний із можливими варіантами для відповіді. У зв'язку із цим запропоновано використовувати сторонні джерела із відкритими даними, які містять більшість з можливих термінологічних баз усіх предметних областей. Найкращим засобом, що відповідає поставленій задачі є пошукова система, а саме її сервіси, що надають можливості доступу та пошуку в її сховищах. Таким чином, було проаналізовано декілька найбільш відомих пошукових систем із наявністю сервісів, що можуть бути використані ІС чат-бота [25-27] (таблиця 2.1).

Таблиця 2.1 – Порівняльний аналіз пошукових систем

	Google Search	BING	Yahoo
Сервіс/користувальницький інтерфейс	Google Custom Search/ Google Service API	BING API, Azure Search API	Yahoo BOSS API
Наявність безоплатного використання	100 запитів на день	3 запити за секунду, 3000 в місяць	Відсутня
Кількість користувачів	> 1 мільярда	Невідомо	Невідомо
Загальна кількість запитів щодня	9 мільярдів	Невідомо	Невідомо
Кількість проіндексованих сторінок	40 мільярдів	Невідомо	Невідомо
Підтримка багатомовного пошуку	Підтримується	Підтримується	Підтримується

Серед представлених пошукових систем було обрано Google Search як найбільш потужну із найбільшою базою індексованих сторінок у мережі Інтернет та відповідно сервіс Google Custom Search.

При одночасній роботі декількох створених інтелектуальних чат-ботів у разі виникнення помилок в роботі необхідно створення журналу історії роботи, формування звітності. Для даної задачі існують готові програмні модулі з можливою інтеграцією. Одним з таких рішень є NLogger [28]. Серед основних переваг у порівнянні з іншими модулями є швидке налаштування та можливість багато-ступеневої звітності для задач інформування, попередження, та тимчасового припинення роботи чат-бота у разі потреби чи для подальшого оновлення.

2.4 Етапи створення інформаційної технології та структурна модель інтелектуального чат-бота

Створення інформаційної технології поділяється на декілька етапів:

1. Визначення компонентів, необхідних для реалізації інтелектуального чат-бота.
2. Створення алгоритму взаємодії між компонентами інформаційної технології створення інтелектуального чат-бота.
3. Створення деталізованих алгоритмів роботи кожного з компонентів.
4. Комплексне поєднання усіх компонентів та формування першого інтелектуального чат-бота із підтримкою декількох предметних областей.
5. Створення технічної документації щодо подальшого використання та можливої модифікації із додаванням нових функцій чи інтегрування в інші інформаційні системи.

Детально розглянувши основні принципи побудови інформаційних технологій, варто зазначити, що ключовими характеристиками успішних інформаційних технологій є прозорість їх структури, а саме взаємодії між усіма компонентами інформаційної технології. Щодо інформаційної технології створення інтелектуального чат-бота дана структурна схема повинна бути максимально чіткою та містити виключно необхідні компоненти.

Для повноцінної роботи інформаційної технології необхідні такі компоненти:

- Репозиторій сховища даних
- Модуль глобального керування
- Модуль попередньої обробки
- Модуль інтелектуального аналізу
- Модуль операційного контролю
- Модуль введення/виведення інформації

Репозиторій сховища даних – необхідний для операцій із термінологічними сховищами даних. Здійснює формування запитів до сховищ даних та контроль за їх виконанням. При реалізації даного компоненту необхідно врахувати структуру типів даних, що зберігатимуться в термінологічних сховищах даних, для цього, даний репозиторій необхідно створити в якості стороннього пакету, із можливістю використання версій та можливості оновлення у разі зміни структур та типів даних в термінологічних сховищах даних без зміни в роботі інтелектуального чат-бота.

Даний модуль складається з чотирьох сервісів та двох API-сервісів: внутрішнього та зовнішнього. Такий розділ присутній у всіх модулів, окрім модулю введення (виведення) інформації. Перевагами даного підходу є відокремлення внутрішньої структури чат-бота, що відповідає за його роботу, та зовнішньої для редагування даних, зміни налаштувань, передачі даних для формування статистики.

DbManagerService – сервіс призначений для керування підключеннями до баз даних, також містить команди для копіювання чи редагування баз даних, що використовуються створеними чат-ботами. Містить 11 команд.

LoadStorageCommand – відповідає за завантаження даних до обраної термінологічної бази даних, а також створення усіх таблиць і взаємозв'язків між ними, якщо бази даних не існує.

SaveChangesCommand – зберігає користувальницькі зміни, під час сеансу із користувачем чат-бота виконує функцію збереження текстової інформації користувача у окремій базі даних.

ConnectToDbCommand – відповідає за підключення до бази даних за допомогою строки підключення та надає доступ до репозиторію. В основному використовується разом з **LoadStorageCommand**.

ResetConnectionsCommand – призначена для скидання прив'язок чат-бота до термінологічних баз даних. Використовується при зміні предметних областей, в яких повинен розумітись чат-бот та запускається лише з панелі глобального керування.

`CreateBackupCommand` – призначена для створення копій основних баз даних, що включають у себе усі бази даних пов’язані з діяльністю глобального модулю інформаційної технології, а також баз даних термінологічних областей для запобігання втрати даних.

`UpdateDbSettingsCommand` – призначена для оновлення налаштувань та налаштування індексації баз даних термінологічних областей.

`ManageGlobalDbConnectionCommand` – призначена для підключення до бази даних глобального модулю керування інформаційною технологією.

`CheckManagerServiceCommand` – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

`SendDbManagerReportCommand` – призначена для формування і надсилання звіту щодо діяльності сервісу, включаючи записи, пов’язані із використанням підключень, загального трафіку та журналу операцій входів у бази даних.

`SetupServicesRelationsCommand` – призначена для реплікації даних у інші бази даних, а також містить інформацію про усі підключені по інформаційної технології сервіси, запускається виключно при потребі встановлення нових сервісів, працює з усіма версіями інших сервісів.

DbRepositoryService – сервіс призначений для репозиторію баз даних. Включає методи для створення, редагування, збереження та отримання даних, включаючи не лише дані термінологічних баз даних, а й баз даних для модуля глобального керування. Також містить опис усіх моделей. Містить 21 команду.

`AddSettingsCommand` – призначена для оновлення налаштувань сервісів. Працює з усіма версіями інших модулів.

`AddCommand` – призначена для додавання нових екземплярів класів в сховища даних.

`UpdateCommand` – призначена для оновлення існуючих записів бази даних.

`RemoveCommand` – призначена для видалення даних з обраної бази даних.

`CompareCommand` – призначена для отримання об’єкту-різниці між даними в обраній базі даних.

`GetByIdCommand` – призначена для отримання екземпляру класу з бази даних за допомогою унікального поля-ідентифікатора.

`GetRangeCommand` – призначена для отримання набору даних з обраної бази даних. Містить фільтрацію для отримання потрібних даних.

`RemoveAllCommand` – призначення для видалення усіх записів обраної таблиці у обраній базі даних.

`AggregateCommand` – призначена для агрегації даних та отримання необхідних даних у визначеному структурованому вигляді.

`GetLightVersionCommand` – призначена для отримання даних без завантаження екземплярів даних, що прив'язані між таблицями, цілісний об'єкт з усіма пов'язаними даними доступно отримати використовуючи команду `AddCommand`.

`UpdateAllCommand` – призначена для оновлення полів екземплярів таблиць із використанням фільтру для фільтрації та оновлення лише необхідних даних.

`ReplaceCommand` – призначена для заміни даних в екземплярах таблиць обраної бази даних.

`MergeDataCommand` – призначена для комбінації даних з декількох термінологічних баз даних.

`CustomFilterUpdateCommand` – призначена для оновлення даних в усіх термінологічних базах даних за вибраними атрибутами обраних класів.

`AddChatbotCommand` – призначена для додавання створеного чат-бота.

`UpdateChatbotCommand` – призначена для оновлення налаштувань чат-бота.

`GetChatbotCommand` – призначена для отримання обраного чат-бота з усіма його зв'язками, доступна лише з панелі глобального керування.

`RemoveChatbotCommand` – призначена для видалення обраного чат-бота.

`GetAllChatbotsCommand` – призначена для отримання списку усіх чат-ботів без їх залежних складових.

`CheckDbRepositoryServiceCommand` – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

DbChatbotService – сервіс призначений для збереження даних щодо історії використання чат-ботів, а також їх технічна інформація. Містить 5 команд.

GetRelatedInfoCommand – призначена для завантаження залежних компонентів чат-бота, підключень до сервісів.

GetReportByChatbotIdCommand – призначена для формування звіту щодо роботи чат-бота, включаючи запис історії використання, використання сторонніх сервісів.

AddToHistoryCommand – призначена для запису завершеного діалогу та його технічної інформації.

MakeSelectionCommand – призначена для повторного проходження обраного сеансу користувача.

RemoveHistoryCommand – призначена для очищення історії сеансів.

DbDashboardService – сервіс призначений для збереження даних модуля глобального керування, звітів щодо роботи чат-ботів, налаштувань. Містить 15 команд.

LoadDataCommand – призначена для завантаження даних необхідних для панелі керування інформаційної технології.

AuthorizeCommand – призначена для авторизації користувача панелі керування інформаційної технології.

LoginCommand – призначена для аутентифікації користувача панелі керування інформаційної технології.

NotifyAllChatbots – призначена для введення текстового повідомлення в усі поточні сеанси користувача.

CreateChatbotCommand – призначена для створення чат-бота.

RemoveChatbotCommand – призначена для видалення чат-бота.

CheckAllServicesCommand – технічна команда, призначена для перевірки усіх підключень та працездатності усіх сервісів, запускає команди **Check<ServiceName>Command**.

`ShowStatisticsCommand` – призначена для збору статистики використання сервісів.

`SetupGoogleServiceCommand` – призначена для налаштування сервісу пошукової мережі Google.

`GetAllDbConnectionsCommand` – призначена для отримання усіх доступних баз даних, що використовує інформаційна технологія в поточний момент часу.

`SaveDbSchemeTemplateCommand` – призначена для збереження схеми даних, що термінологічною бази даних.

`UpdateUserDataCommand` – оновлення даних щодо користувачів чат-бота.

`LoadUsersCommand` – завантаження даних щодо користувачів чат-ботів.

`RemoveUserCommand` – видалення користувача та усю історію його використання чат-ботів.

`CompareDataCommand` – призначена для порівняння даних на панелі керування інформаційної технології.

DbOutsideAPIService – сервіс призначений для зовнішнього контролю та надання результату виконання запитів (доступно лише з сервісу `GlobalOutsideAPIService`) Містить 6 методів.

DbInsideAPIService – сервіс призначений для перенаправлення запитів від інших модулів у команди для подальшого виконання Містить 6 методів.

Модуль глобального керування – необхідний для глобального керування створеними інтелектуальними чат-ботами. А також для їх автоматичного оновлення чи зміни налаштувань, надання доступу до термінологічних сховищ даних предметних областей. Даний модуль повинен бути реалізований у вигляді веб-інтерфейсу із можливістю авторизації та обов'язковою політикою ролей. Серед додаткових опцій повинна бути можливість перевірки активності будь-якого із створених чат-ботів, можливість створення тестування у вигляді завантаження тексту, а також можливість одночасної роботи з декількома створеними чат-ботами.

Даний модуль складається з п'яти сервісів та двох API-сервісів. Кожен із сервісів містить запис історії використання для подальшого перегляду.

GlobalCoreService – сервіс містить бізнес-логіку керування чат-ботами.

Містить 5 команд.

SetActiveChatbotCommand – призначена для активації чат-бота.

DisableChatbotCommand – призначена для деактивації чат-бота.

ConnectToChatbotCommand – призначена для підключення та отримання доступу до керування чат-ботом, зміною його налаштувань.

NextActionCommand – призначена для обробки дій панелі користувачів.

CheckGlobalCoreServiceCommand – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

GlobalDataService – сервіс містить класи-моделі, що описують структуру чат-бота. А також методи для редагування інформації чат-ботів. Містить 5 команд.

UpdateChatbotDataCommand – призначена для запису даних від чат-бота.

FindDataByFieldsCommand – призначена для отримання даних з термінологічних баз даних за обраними полями.

ReplaceCommand – призначена для оновлення даних в термінологічній базі даних.

EditSchemeTemplate – призначена для редагування структури таблиць термінологічної бази даних.

CheckGlobalDataServiceCommand – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

GlobalLogService – сервіс призначений для запису та формування звітів щодо історії використання сервісів. Містить 4 команди.

InfoLogCommand – призначена для запису інформації щодо сеансів користувачів, а також діяльності сервісів, що використовуються під час сеансу користувача.

WarningLogCommand – призначена для запису попереджень, різних версій сервісів, включаючи ситуації, коли запити не виконувались.

ErrorLogCommand – призначена для запису історії помилок, що виникали під час роботи інформаційної технології.

`CheckGlobalLogCommand` – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

GlobalControlService – сервіс призначений для прямого керування чат-ботом. Включає в себе методи, що використовують сторонні сервіси, включаючи `GoogleCustomSearch`, для пошуку інформації у пошуковій мережі Google. Містить 8 команд.

`AnalyzeCommand` – призначена для аналізу повідомлення від користувача.

`SendMessageCommand` – призначена для надсилання текстового повідомлення користувачу.

`SendResultMessageCommand` – призначена для надсилання текстового повідомлення у відповідь на повідомлення від користувача.

`SendInformationCommand` – призначена для надсилання технічної інформації щодо історії діалогу під час сеансу користувача.

`FindByGlobalSearchCommand` – призначена для пошуку відповіді на повідомлення користувача із використанням пошукового сервісу.

`FindByLocalStorageCommand` – призначена для пошуку відповіді на повідомлення користувача із використанням локального сховища даних.

`FindByConnectedDbsCommand` – призначена для пошуку відповіді на повідомлення користувача із використанням підключених термінологічних баз даних.

`CheckGlobalControlServiceCommand` – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

GlobalDashboardService – сервіс призначений для обробки запитів від панелі керування. Містить 16 методів.

GlobalOutsideAPIService – сервіс призначений для зовнішнього контролю та надання результату виконання запитів. Містить 10 методів.

GlobalInsideAPIService – сервіс призначений для перенаправлення запитів від інших модулів у команди для подальшого виконання. Містить 8 методів.

Модуль попередньої обробки – є компонентом власне інтелектуального чат-бота та містить в собі удосконалений метод логіко-семантичного аналізу тексту. Вихідними даними даного модуля є набір ключових слів, а також звітна інформація щодо характеристики вхідної інформації.

Даний модуль складається із двох сервісів та двох API-сервісів: внутрішній та зовнішній. Вхідними даними для сервісів є команди.

PreviewService – включає в себе семантичний аналіз текстової інформації. Містить 4 команди.

AnalyzeCommand – призначена для семантичного аналізу текстової інформації.

FormDataCommand – допоміжна команда для перетворення формату текстового повідомлення на дані, що є вхідними даними для **AnalyzeCommand** команди.

NextCommand – призначена для передачі результатів на інші сервіси та збереження даних результатів у локальному сховищі чат-бота.

CheckPreviewServiceCommand – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

PreviewDataService – включає в себе опис моделей даних, містить розширення для базових операцій з текстовими даними.

PreviewInsideAPIService – сервіс призначений для перенаправлення запитів від інших модулів у команди для подальшого виконання. Містить 2 методи.

PreviewOutsideAPIService – сервіс призначений для зовнішнього контролю та надання результату виконання запитів. Містить 4 методи.

Модуль інтелектуального аналізу – складається із тренованої нейронної мережі, а також засобів для обробки результатів та трансформування у логічні висновки. Вхідними даними є вихідні дані модулю попередньої обробки. Вихідними даними є набір інструкцій та команд для репозиторію сховища даних та модулю операційного контролю. Основний модуль інтелектуального чат-бота.

Даний модуль використовує чотири сервіси два API-сервіси: внутрішній та зовнішній. Вхідними даними для сервісів є команди.

AISetupService – сервіс призначений для автоматичного та мануального налаштування параметрів нейронної мережі. Містить 4 команди.

`SetupTensorFlowCommand` – призначена для початкового налаштування TensorFlow для інформаційної технології, встановлення необхідних пакетів.

`GetSettingsCommand` – призначення для завантаження налаштувань, що використовуються в TensorFlow фреймворку та зберігаються в базі даних глобального модулю керування.

`CheckTensorflowAPICompabilityCommand` – призначена для перевірки сумісності поточної реалізації із останньою версією бібліотек фреймворку TensorFlow.

`CheckAISetupService` – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

AITrainService – сервіс призначений для тестування нейронної мережі на наборах даних. Містить 4 команди.

`TrainCommand` – призначена для тренування нейронної мережі. Містить Python скрипти для запуску процесу навчання нейронної мережі. А також використовує методи для автоматичного налаштування параметрів нейронної мережі у разі їх відсутності у базі даних глобального модулю керування.

`TestCommand` –призначена для тестування натренованої нейронної мережі, а саме містить Python скрипти для тестування вагової матриці нейронної мережі.

`UpdateWeightMatrixCommand` – призначена для оновлення ваг нейронної мережі. Містить Python скрипти.

`CheckAITrainServiceCommand` – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

AIProcessService – сервіс призначений для виконання обробки даних. Основний модуль, що містить бізнес-логіку обробки даних нейронною мережею. Містить 4 команди.

`ProcessDataCommand` – призначена для обробки вхідної інформації у вигляді блоків текстової інформації.

`AnalyzeTextCommand` – відповідає за знаходження відповіді у термінологічній базі даних.

`AnalyzeTextThemeCommand` – відповідає за визначення тематики вхідної текстової інформації, а також відповідно і термінологічних баз даних, в яких буде доцільний подальший пошук інформації для формування відповіді користувачу.

`CheckAIProcessServiceCommand` – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

AIInfrastructureService – сервіс призначений для збереження тренованих вагових матриць нейронної мережі для кожного із чат-ботів (кожен чат-бот містить власну копію вагової матриці, що може в процесі використання змінюватись). Містить 4 команди.

`SaveWeightMatrixCommand` – зберігає вагову матрицю нейронної мережі у базі даних.

`UpdateNetworkSettingsCommand` – призначена для оновлення налаштувань нейронної мережі для подальшого перенавчання.

`CreateNewNetworkCommand` – призначена для створення нової вагової матриці нейронної мережі.

`CheckAIInfrastructureServiceCommand` – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

AIInsideAPIService – сервіс призначений для перенаправлення запитів від інших модулів у команди для подальшого виконання. Містить 4 методи.

AIOutsideAPIService – сервіс призначений для зовнішнього контролю та надання результату виконання запитів. Містить 2 методи.

Модуль операційного контролю – здійснює порівняння отриманих даних з модулю інтелектуального аналізу для використовуваних інтелектуальним чат-ботом предметних областей. Формує запити до пошукової системи у разі потреби та здійснює оперативний контроль за поточним функціонуванням чат-

бота. Найвищий рівень абстракції інтелектуального чат-бота. Повинен містити технічну інформацію щодо виконаної роботи. При необхідності передає керування інтелектуальним чат-ботом на модуль глобального керування.

Даний модуль складається з чотирьох сервісів та двох API-сервісів.

BotControlService – даний сервіс містить команди для обробки запитів від користувача, включаючи текстові дані, що не були використані нейронною мережею. Містить 5 команд.

AnalyzeCommand – призначена для аналізу повідомлення від користувача.

SendResultMessageCommand – призначена для надсилання текстового повідомлення у відповідь на повідомлення від користувача.

SendInformationCommand – призначена для надсилання технічної інформації щодо історії діалогу під час сеансу користувача.

FindResultCommand – призначена для пошуку відповіді на повідомлення користувача із використанням усіх доступних засобів (пошукова мережа, термінологічні бази даних, локальне сховище).

CheckBotControlServiceCommand – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

BotCoreService – даний сервіс містить бізнес-логіку обробки даних від користувача. Містить 4 команди.

NextActionCommand – призначена для обробки дій користувача.

StartSessionCommand – призначена для початку нового сеансу користувача. Включає надсилання користувачу початкового повідомлення.

FinishSessionCommand – призначена для завершення поточного сеансу користувача. Виконується у разі завершення реплік чи по завершенню часу.

CheckBotCoreServiceCommand – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

BotOIService – даний сервіс відповідає за передачу даних на модуль інтелектуального аналізу та отримання даних від модуля попередньої обробки. Містить 5 команд.

`ReceiveTextCommand` – призначена для отримання текстового повідомлення користувача та його збереження.

`ReceiveDataCommand` – призначена для отримання результатів семантичного аналізу тексту.

`SendToAnalyzeCommand` – призначена для подальшої обробки модулем інтелектуального аналізу.

`SendDataToPanelCommand` – призначена для надсилання інформації в історію діалогу, що можна переглянути на панелі глобального керування для подальшого збереження в базі даних.

`CheckBotOIServiceCommand` – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

BotConnectorService – даний сервіс відповідає за канал зв'язку та містить набір інтерфейсів для використання чат-бота у різних соціальних мережах. Містить 6 команд.

`ConnectToFacebookCommand` – призначена для підключення чат-бота до соціальної мережі Facebook.

`ConnectToTelegramCommand` – призначена для підключення чат-бота до соціальної мережі Telegram.

`ConnectToTwitterCommand` – призначена для підключення чат-бота до соціальної мережі Twitter.

`ConnectToViberCommand` – призначена для підключення чат-бота до соціальної мережі Viber.

`CheckBotConnectorServiceCommand` – технічна команда, призначена для перевірки усіх підключень та працездатності усіх команд даного сервісу.

Модуль введення/виведення інформації – складається із інтерфейсу, що приймає вхідну інформацію від користувача та надає вихідну текстову інформацію у разі потреби користувачу. Даний модуль пов'язаний виключно з операційним модулем для інтелектуального чат-бота. Даний модуль складається з одного сервісу та одного API-сервісу.

IOService – сервіс призначений для відображення інтерфейсу користувача, панелі керування, а також веб-інтерфейсу чат-ботів (використовуються за замовчуванням). Містить 6 методів.

IOAPIService – сервіс призначений для передачі вхідної інформації від користувача до інших сервісів, пов’язаних модулів. Містить 12 методів.

Сумарно за працездатність інформаційної технології безпосередньо відповідають 20 сервісів.

Схема взаємодії між компонентами інформаційної технології подана на рисунку 2.4.

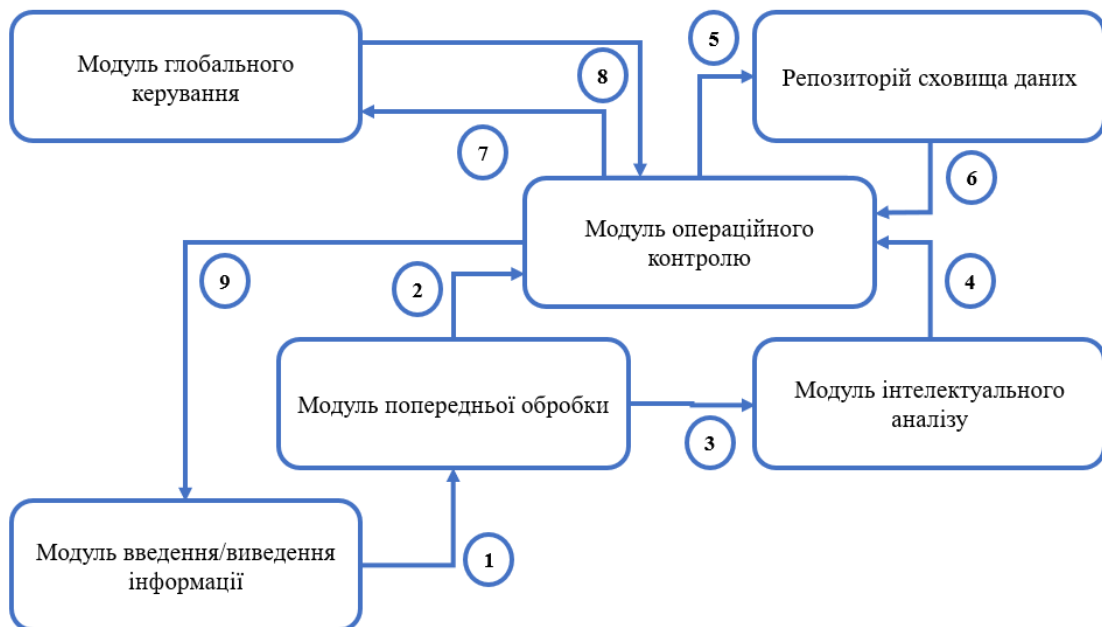


Рисунок 2.4 – Схема взаємодії компонентів інформаційної технології створення інтелектуальних чат-ботів

Процес виконання аналізу текстової інформації інтелектуальним чат-ботом, а саме покроковість виконання аналізу подана у вигляді нумерації переходів між компонентами інформаційної технології. Структурна модель інтелектуального чат-бота представлена на схемі (рисунок 2.5).

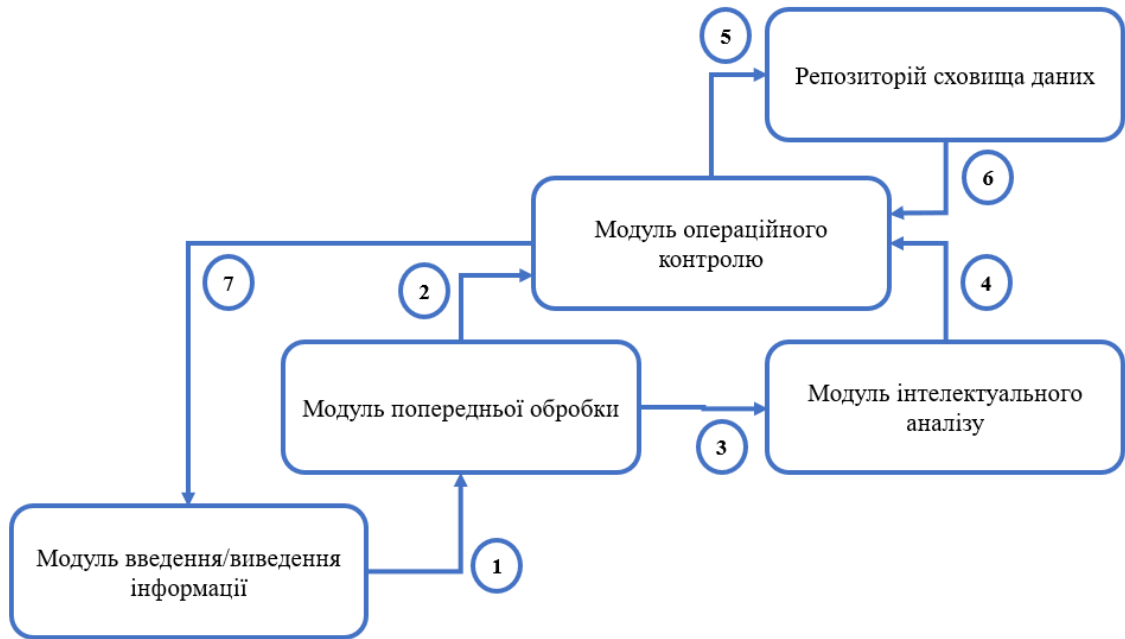


Рисунок 2.5 – Структурна схема інтелектуального чат-бота

Загальна архітектура інформаційної технології міститиме клієнт-серверну архітектуру із використанням API-сервісів та інтерфейсів модулів. Дану схему зображено на рисунку 2.6.

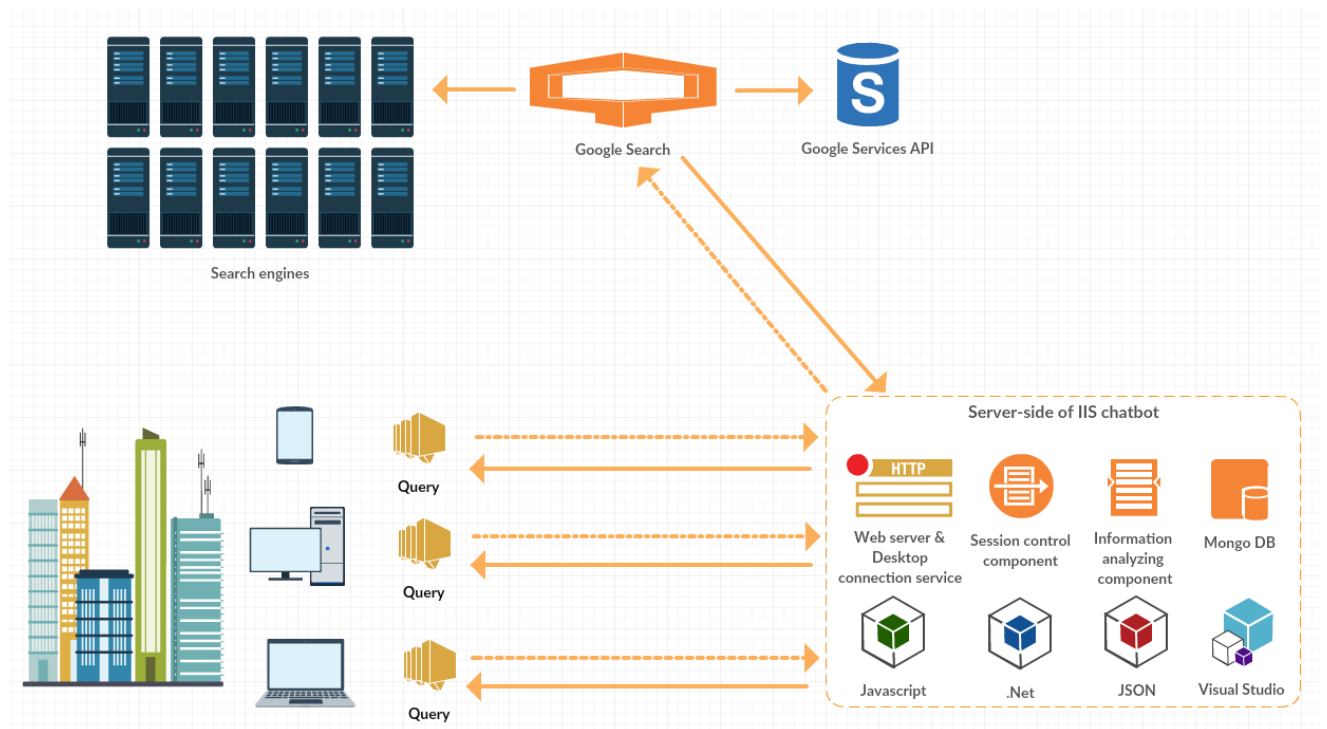


Рисунок 2.6 – Архітурна модель інформаційної технології

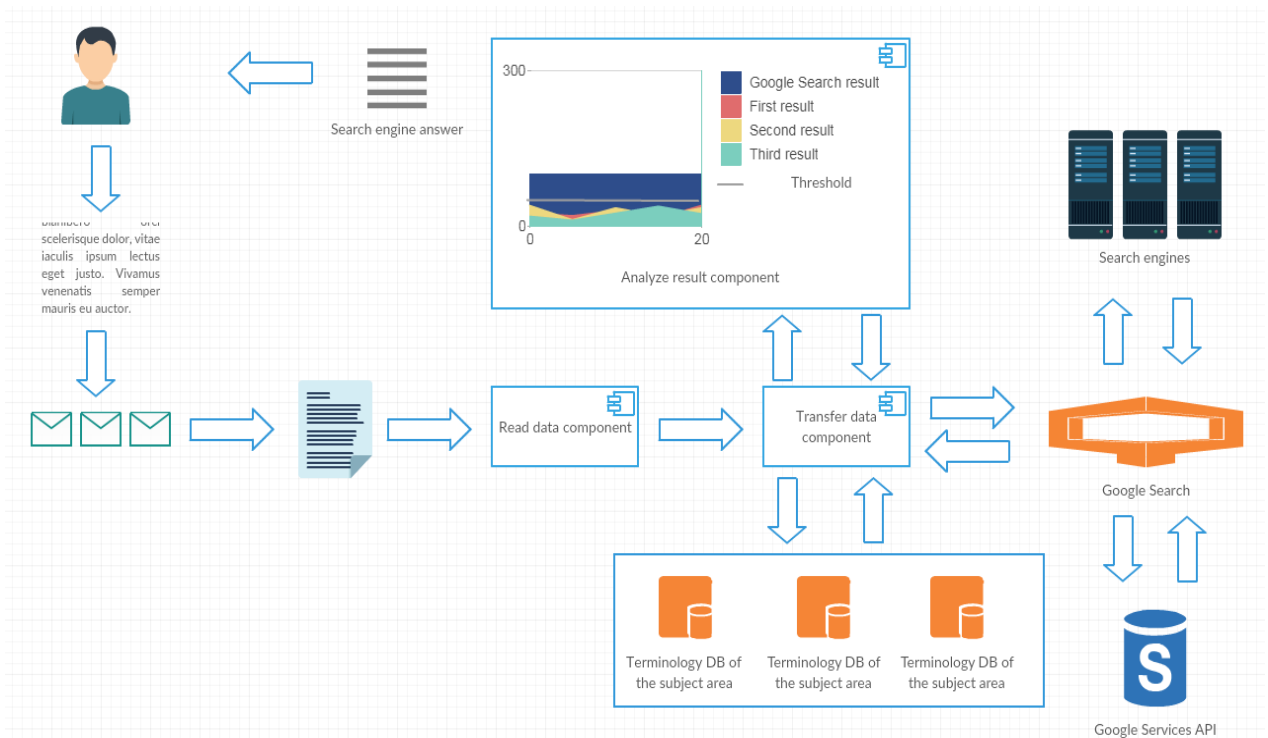


Рисунок 2.7 – Процес аналізу текстового повідомлення користувача

2.5 Висновок

В даному розділі виконано аналіз предметної області інтелектуальної технології створення інтелектуального чат-бота. Розроблено структурну схему інформаційної технології. Визначено та проаналізовано усі етапи обробки текстової інформації. Запропоновано та обґрунтовано використання удосконаленого методу семантичного аналізу тексту, використання штучної рекурентної нейронної мережі для задачі розпізнавання текстової інформації. Серед розглянутих алгоритмів та методів аналізу текстової інформації були відзначені алгоритми: технології машинного навчання та нейронні мережі у особливості рекурентні із-за наявності зворотного поширення сигналу, що притаманне для обробки текстової інформації та задачі класифікації. Розроблено структуру інформаційної технології інтелектуального чат-бота. Інформаційна технологія складається з шести компонентів: репозиторій сховища даних, модуль глобального керування, модуль попередньої обробки, модуль інтелектуального аналізу, модуль операційного контролю, модуль

введення/виведення інформації. В якості технології обробки текстової інформації для задачі класифікації тексту було обрано рекурентну нейронну мережу із використанням алгоритму довготривалої короткочасної пам'яті завдяки стійкості до часових прогалин, незалежності від розміру вхідних даних, а також усуненню проблеми зникнення градієнту під час навчання нейронної мережі. Для задачі кластеризації тексту було обрано нейронну мережу Кохонена завдяки можливості найшвидшого перенавчання для підвищення рівня кластеризації.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНИХ ЧАТ-БОТІВ

3.1 Обґрунтування вибору програмно-апаратної платформи

Використання нейронної мережі потребує наявність місткого сховища даних для тренувальної вибірки та термінологічної бази обраної предметної області чи областей. Обираючи сховище даних для програмного модуля інтелектуального чат-бота основними критеріями виступають такі параметри, як:

- Здатність до масштабованості
- Можливість розширення
- Стійкість до втрати даних
- Можливість резервування копій сховища даних

Серед найбільш популярних сховищ даних, що цілком задовольняють поставлені критерії до порівняння було обрано MS SQL, Mongo DB, Firebird DB (таблиця 1) [29-31].

Таблиця 3.1 – Порівняльна характеристика баз даних

	MS SQL	Mongo DB	Firebird DB
Мова запитів	T-SQL	Mongo Query Lang	T-SQL & OLAP & OLTP
Ядро бази даних	C & C++ & C#	C++	C++
Тип бази даних	Реляційна база даних	Об'єктно-орієнтована база даних	Кросс-платформна реляційна база даних
Постійне удосконалення роботи	Оновлення версій, додавання нових функцій	Постійне удосконалення бази даних	Вдосконалення бази даних спільнотою розробників

Продовження таблиці 3.1

Наявність можливості безкоштовного використання	Безкоштовний контент, який містить величезну кількість можливостей для роботи з базою даних	Безкоштовний контент із наявністю комерційних версій	Безкоштовний контент
Максимальний розмір сховища	524,272 TB	необмежено	необмежено
Наявність індексів	наявна	наявна	наявна
Можливість візуалізації даних	SQL Server Management Studio (SSMS) засіб для візуалізації та керування даними в MS SQL	Mongo Atlas & Robomongo & Robo 3T	відсутня
Використання розподілених компонентів бази даних	наявна	наявна	відсутня
Можливість асинхронного виконання запитів	наявна	наявна	наявна
Можливість імпорту/експорту даних	наявна	наявна	наявна
Наявність драйверів для високорівневих мов програмування	ADO .NET & Entity Framework & Entity Framework Core	Mongo Drivers	Firebird Drivers

Після аналізу запропонованих сховищ даних, в першу чергу вирішено використовувати Mongo DB із-за ряду технічних та функціональних особливостей, головними з яких є: об'єктно-орієнтований підхід, можливість обробки слабо-структурованих даних, наявність потужної технології асинхронного виконання запитів, транзакцій. Наявність драйвера для конвертації даних у JSON-формат, що може використовуватись для передачі даних у мережі

Інтернет по мережевому протоколу. Додатково для сервісів модулю операційного контролю чат-бота та модуля попередньої обробки даних буде використано MS SQL у зв'язку із строгою типізацією та необхідністю індексування, а також приведення відношень допоміжної бази даних у відповідності із третьою нормалізованою формою. Для підключення до баз даних необхідно використовувати файли конфігурації проекту. Для технічних даних, констант, найменування полів та ідентифікаційних даних створених чат-ботів буде створено окрему базу даних.

Для подальшого супроводу чи зміни структури баз даних, необхідно створення міграцій та локальних копій.

3.2 Обґрунтування вибору мови програмування

Обравши технологію обробки вхідних даних та сховище даних для термінологічної бази обраної предметної області чи областей, потрібно обрати платформу для реалізації ІС чат-бота. Серед платформ-аналогів, більшість платформ має модульну структуру, причому кожен з яких може мати власну інфраструктуру і може бути створений за допомогою різних мов програмування. У зв'язку з цим, під час аналізу сучасних мов програмування, необхідно враховувати не лише наявність інтеграції потрібних технологій, а й можливість використання декількох мов програмування в одному модулі. Оскільки вибраний тип чат-бота потребує повної програмної реалізації, то вибір платформи трансформується на вибір мови програмування для кожного з модулів, що дозволить реалізувати усі заплановані функції та характеристики усіх програмних модулів для створення інтелектуального чат-бота. Середовищем програмування обрано Microsoft Visual Studio 2019 [32], яка підтримує більшість мов програмування високого рівня. Порівнюючи найбільш потужні мови програмування, було розглянуто C# 7.2, Java SE 11 та EcmaScript 6 (таблиця 2) [33, 34]:

Таблиця 3.2 – Порівняльна характеристика мов програмування

	C# 7.2	Java SE 11	Ecmascript 6
Стандарт	2000, ECMA, ISO	Java Language Specification	1997, ECMA
Підтримка багато потокової обробки	підтримується	підтримується	підтримується, починаючи з NodeJS 8
Обробка помилок	підтримується	підтримується	підтримується
Кросплатформність	підтримується	підтримується	підтримується
Парадигми програмування	ООП, структурна та конкурентна парадигми	ООП, конкурентна парадигма	ООП, організоване на наслідуванні прототипів
Наявність бібліотек для роботи з нейронними мережами	AForge, Alexa, TensorFlow	Відсутні потужні бібліотеки	TensorFlow, Brain.js, Synaptic, Neataptic, Conventjs, Webdnn
Можливість роботи з комітками пам'яті	Unsafe code, Garbage Collector	Підтримка управління пам'яттю	не підтримується
Сфера застосування	Application, CRM, RAD, business, client-side, general, mobile development, server-side, web	Application, business, client-side, general, mobile development, server-side, web	Client-side, server-side, web
Середовище виконання	CLR, JIT compiler	Java Virtual Machine (JVM)	NodeJS, Web-pages
Середовище розробки	Microsoft Visual Studio, Rider IDE, Visual Studio Code	Eclipse, Android Studio, IntelliJ JetBrains	WebStorm, Microsoft Visual Studio, Visual Studio Code

Серед представлених мов програмування високого рівня, вибір був зроблений на користь C# 7.3 із-за зручності роботи з нейронними мережами, широкими можливостями у плані сфері застосування, дружелюбного та інтуїтивно зрозумілого середовища розробки Microsoft Visual Studio. Робота з багато потоковими проектами та підтримкою асинхронності виконання стали ключовими факторами вибору даної мови програмування.

Для веб-інтерфейсу модуля введення (виведення) інформації було обрано EcmaScript 6 у зв'язку із підтримкою більшості браузерів та великій кількості інструментів для виведення даних.

Для роботи з нейронною мережею було обрано мову програмування Python. Перевага даної мови програмування у порівнянні з іншими є наявність великої кількості бібліотек для обробки великих масивів даних, а також наявність сторонніх бібліотек, що значно спрощують пришвидшують процес навчання нейронних мереж завдяки оптимізованим алгоритмам. Додатково для візуальної частини модуля введення (виведення інформації) було обрано мову програмування Typescript, що використовується фреймворком React. Основною перевагою даної мови є строга типізація.

Серед технологій та сторонніх бібліотек, що будуть використані під час технічної реалізації варто відзначити:

React – бібліотека, фреймворк для створення інтерфейсу користувача. Дозволяє створювати декларативні користувальницькі інтерфейси, в тому числі і веб-орієнтовані. Застосовується в глобальному модулі керування, модулі введення (виведення) інформації, модулі операційного контролю чат-бота. Основна ідея полягає у використанні компонентів із власним станом, що може використовувати як власні засоби, описані в тілі компонента, так і вхідні дані, відповідно змінюючи інтерфейс. Уся допоміжна інфраструктура компонентів написана на Javascript, у зв'язку з чим, можлива передача даних у форматі JSON. Додатково існує чимала кількість сторонніх бібліотек із інтеграцією даного

фреймворку. Завдяки цьому значна кількість сучасних веб-інтерфейсів використовують саме даний фреймворк [35].

.NET Core 3.1 – модульна платформа для розробки програмного забезпечення з відкритим кодом. .NET Core заснована на .NET Framework. Платформа .NET Core відрізняється завдяки модульній структурі, можливості роботи на різних платформах. Дана платформа має велику кількість технологій та доступ до хмарних технологій завдяки Microsoft Azure Services. Кожен компонент оновлюється через менеджер NuGet. Кожен модуль має можливість працювати з різними версіями інших модулів і не залежить від оновлення всієї платформи. Завдяки наявності підтримки сторонніх бібліотек, бізнес-логіка, основна частина сервісів написана на даній платформі та має задокументовану інфраструктуру [36].

Swagger – бібліотека для створення автоматичного опису усіх доступних методів контролерів, а за необхідності й сервісів, що використовуються у інформаційній технології. Перевагами її використання серед інших бібліотек є прозорість налаштування, можливість автоматичного оновлення документації під час оновлення модулів інформаційної технології, а також можливість генерування інтерфейсів об'єктів для передачі на клієнтську частину веб-додатку [37].

Pandas – програмна бібліотека, написана для мови програмування Python для маніпулювання даними та їхнього аналізу. Однією з ключових переваг у порівнянні з іншими подібними бібліотеками є наявність власного апарату структур даних та операцій для маніпулювання чисельними таблицями та часовими рядами. Також дана бібліотека містить інструменти для зчитування, обробки, запису даних між структурами даних у пам'яті та різними форматами файлів, інструменти для підтримки пропущених даних, форматування для отримання зведених даних. Дана бібліотека також підтримує ієрархічне індексування масивів для роботи з наборами багатовимірними даними в структурах з нижчою вимірністю [38].

MediatR – поведінковий шаблон дизайну, який допомагає зменшити хаотичні залежності між об'єктами. Головною метою даного шаблону є заборона прямого обміну між об'єктами і замість цього використання посередника [39].

Послуги чи класи часто залежать від інших класів, і досить швидко призводить до підвищення складності програмного продукту. Шаблон посередника служить організатором і викликає всі необхідні служби та сервіси, узгоджуючи їх залежності. Жодна служба не залежить від іншої, лише від посередника. Завдяки використанню платформи .NET Core, було використано реалізацію даного шаблону в бібліотеці MediatR для C# [40].

HtmlAgilityPack – бібліотека для платформи .NET Core, що дозволяє обробляти запити із використанням мови розмітки HTML. Завдяки потужним функціональним можливостям є широко використовуваною у сучасних програмних продуктах із використанням платформи .NET Core [41].

Entity Framework Core – фреймворк для бази даних, один з найбільш популярних та функціональних із підтримкою усіх сучасних засобів та функцій для керування реляційною базою даних MS SQL [42].

Xunit – фреймворк для юніт-тестування серверної частини модулів інформаційної технології створення інтелектуального чат-бота, що написані на платформі .NET Core [43].

Microsoft Framework Bot – фреймворк для реалізації API-сервісів чат-бота, використання гнучкої інфраструктури для налаштування базових компонентів чат-бота, вибору конфігурації, наявності можливості формування нотацій на основі паралельних сеансів користувачів [44].

Google DialogFlow – фреймворк для застосування ключових слів у діалогах, знаходження стоп-слів, із можливістю визначення приблизної тематики за набором ключових слів [20].

MongoDB – об'єктно-орієнтовна база даних, основний формат даних BSON (Binary JSON). Наявність бібліотеки для обробки даних, підтримка агрегації даних і MapReduce [45].

У зв'язку із використанням значної кількості додаткових бібліотек та сервісів необхідно періодично виконувати інтеграційне тестування та перевіряти працездатність сервісів. При необхідності кожен із зовнішніх сервісів кожного із модулів інформаційної технології містить щонайменше одну команду для перевірки працездатності сервісу. У разі потреби можливо формувати звіт із результатів виконання даних команд. Наразі дана опція не реалізована.

3.3 Розробка схеми алгоритму взаємодії основних компонентів інформаційної технології

Кожен модуль представлений у вигляді бібліотеки або компонента, що має власну версію та у разі оновлення, повинен бути сумісний з іншими компонентами. Таким чином, п'ять модулів представлені у вигляді п'яти проєктів, об'єднаних в одне ціле. Залежності між даними модулями представлена на рисунку 2.4. А власне інформаційна технологія, включає в себе також модуль глобального керування, що відповідає за контроль над усіма створеними інтелектуальними чат-ботами та містить увесь необхідний функціонал для налаштування, запуску чи припинення роботи, а також налаштування конфігурації та надання доступу до баз даних предметних областей. Оскільки усі бази даних термінологічних областей мають однакову структуру (кількість таблиць, а також їх моделі з усіма атрибутами та їх властивостями), то кожен створений за допомогою інформаційної технології чат-бот, матиме щонайменше можливість пошуку в інших термінологічних базах даних, якщо на це буде відповідний дозвіл від оператора на веб-сторінці панелі керування чат-ботами. Під час взаємодії між чат-ботом та глобальною панеллю керування, відбувається надсилання запитів на сервер з подальшим перенаправленням на панель керування, що представлена у вигляді веб-сайту із API-сервісом, що в свою чергу має доступ до усіх компонентів чат-бота. У разі потреби зміни налаштувань, глобальна панель керування за допомогою команд, надсилає запит на сервер, де функціонує необхідний чат-бот. При необхідності є

можливість, збереження історії запитів у кожного модуля та сервісів інформаційної технології. У разі необхідності розширення функціональних можливостей за рахунок використання сторонніх бібліотек чи створення нових сервісів, необхідно реалізувати механізм збереження запитів, механізм блокування випадкових протиріч між запитами. У разі копіювання та одночасної роботи двох та більше копій модуля глобального керування, кожен запит повинен містити унікальний ідентифікатор для відслідковування виконаних операцій.

Модуль попередньої обробки забезпечує перетворення вхідної інформації на стандартизовану. Для цього ще при введенні тексту користувачем, враховується час написання та довжина введеного тексту. Після введення повідомлення, починається семантичний аналіз тексту для виявлення набору ключових слів та отримання масивів даних, що складаються з наборів слів, об'єднаних між собою за тематикою та ключові слова, що надають змогу попередньо визначити тематику повідомлення. Під час виконання семантичного аналізу, технічна інформація (час обробки, кількість слів) надсилаються у вигляді запиту на сервер модулю інформаційного контролю, який фіксує отриману інформацію та у разі потреби надсилає її на сервер глобальної панелі керування інформаційної технології. Дана інформація може бути використана для формування звітності чи історії користування обраним чат-ботом. Під час тестування дана статистична інформація надає змогу покращення загальної обробки текстових даних. Усі основні схеми взаємодії модулів інформаційної технології створення інтелектуального чат-бота представлено на рисунках 3.1-3.3.

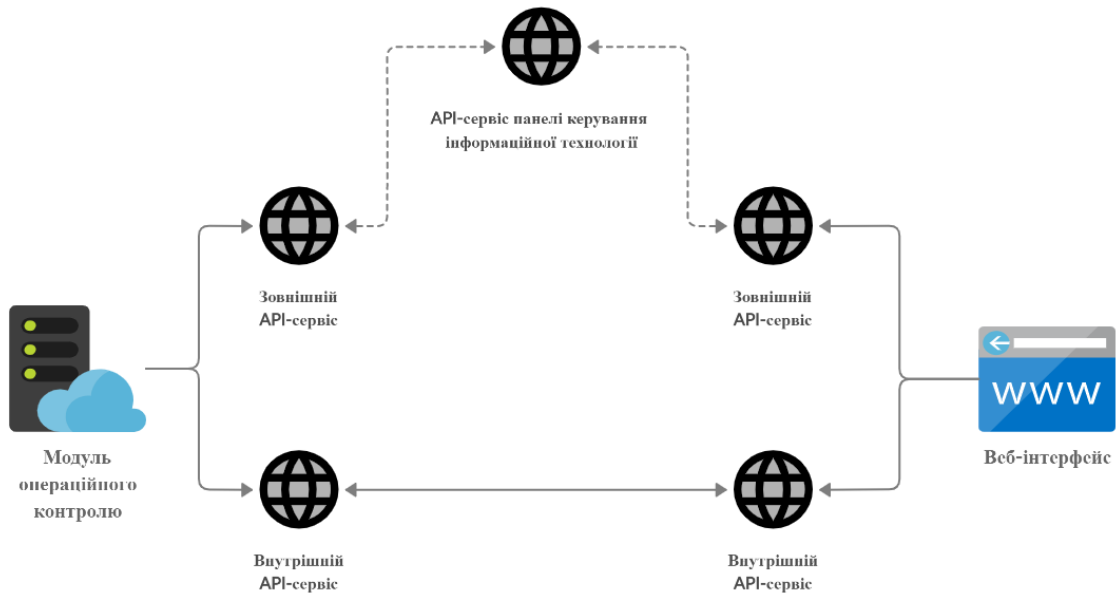


Рисунок 3.1 – Схема взаємодії між модулями введення (виведення) інформації та модуля операційного контролю

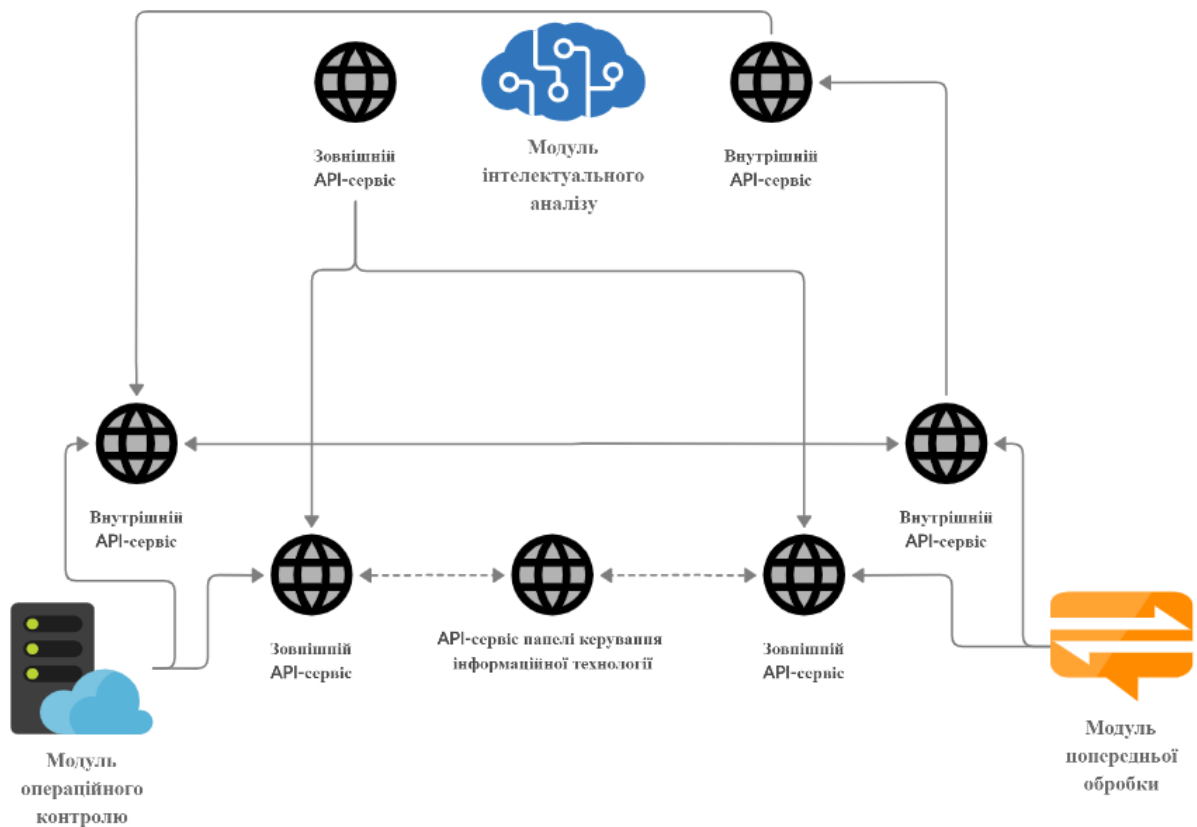


Рисунок 3.2 –Схема взаємодії між модулями операційного контролю, модулю попередньої обробки та модулям інтелектуального аналізу

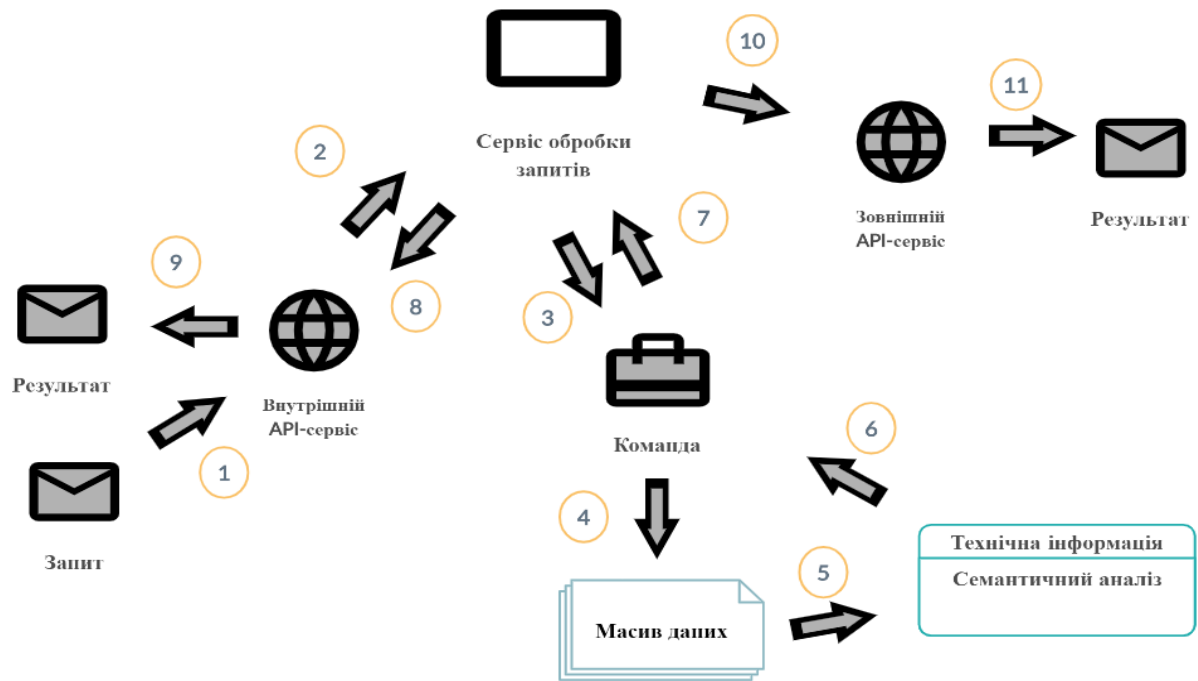


Рисунок 3.3 – Схема обробки інформації в модулях інформаційної технології на прикладі модулю попередньої обробки даних

3.4 Тестування інтелектуального чат-бота в обраних предметних областях та аналіз результатів

Після програмної реалізації усіх сервісів, їх складових, обробників команд та інтерфейсу користувача панелі керування та панелі використання чат-боту, було створено три чат-боти. Для кожного з чат-ботів було сформовано власну вагову матрицю нейронної мережі, що формувалась на тестових наборах даних по 20000 текстів-діалогів на 7-12 реплік у кожному. Мова діалогів – англійська. Кожен з чат-ботів після усіх налаштувань було протестовано на однаковому наборі текстів-діалогів, з пропущеними репліками. Після виконання тестування було проведено порівняльний аналіз отриманих результатів, а також технічної інформації щодо часу обробки текстових повідомлень, проміжних результатів семантичного аналізу текстів та інше. Більш детально процес тестування зображений на рисунках 3.4-3.9.

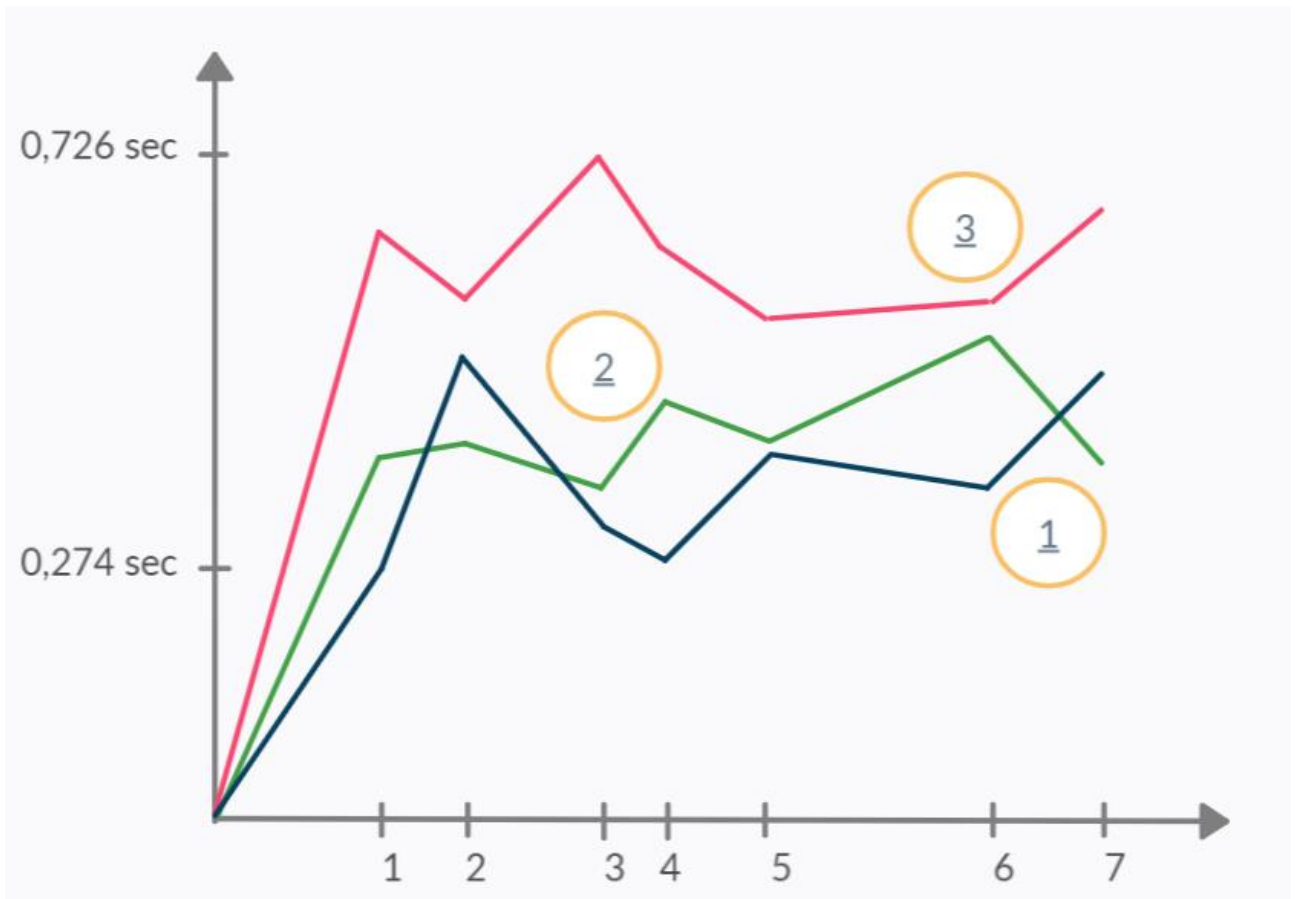


Рисунок 3.4 – Графік швидкодії обробки вхідного повідомлення чат-ботами

Синій графік (кружечок 1) використовує одну предметну область, зелений графік (кружечок 2) використовує дві предметні області, рожевий графік (кружечок 3) використовує одразу три предметні області. Відповідно до цього доцільно зробити висновок щодо зростання часу обробки семантичного ядра у разі збільшення кількості предметних областей. Після багаторазового тестування на різних текстах, відхилення були в межах 5%-7%. При додаванні третьої предметної області у вигляді термінологічної бази даних, загальний час кластеризації вектору вхідних даних підвищувався на 50%-60%.

Аналізуючи отримані результати, рівень розпізнавання текстової інформації прямопропорційно залежить від розміру сховища даних. При можливості розширення сховища, рівень розпізнавання інформації буде зростати. При порівнянні роботи чат-бота із використанням однієї, чотирьох та восьми термінологічних баз даних, рівень розпізнавання інформації досить суттєво відрізняється (таблиця 3.3).

Таблиця 3.3 – Результати автоматичного тестування

Кількість термінологічних баз даних	1	4	8
Кількість тестів, речень	5000	5000	5000
Семантичне ядро (без використання покращення), слів	23	24	26
Семантичне ядро (з використанням покращення), слів	17	17	18
Точність визначення тематики речення (без використання покращення)	87,396%	89,824%	92,108%
Точність визначення тематики речення (з використанням покращення)	90,399%	93,472%	94,813%

Точність визначення тематики речення при зростанні кількості термінологічних баз даних збільшується, за рахунок збільшення загального об'єму інформації. Вхідний вектор для кластеризації із використанням удосконаленого методу семантичного аналізу тексту підвищує точність визначення тематики тексту в середньому на 2-3% із-за зменшення виникнення протиріч між результатами для значень вхідного вектора слів, що складають семантичне ядро тексту. В даному випадку, загальна тематика містить виключно адекватні дані без наявності дублюючих чи надлишкових значень.

<
МОНІТОРИНГ
ПІДКЛЮЧЕННЯ
СТАТИСТИКА
НАЛАШТУВАННЯ
ПРЯМЕ
>

Сеанс: 3f2badd3-fcb1-498d-bcf7-8a0c91e5e542 --- Режим перегляду

Історія сеансу

Give me current timetable
Time:20:04:03

Please, select kind of transport and period of time.
Process: 0.649 sec Time: 20:04:08

Trains, for tommorow to London
Time:20:05:45

We found three accepted records for tommorow.
Process: 0.476 sec Time: 20:06:01

Give me two tickes, if possible at the morning
Time:20:07:12

Ok, wait a minute. We will reserve your places.
Process: 0.703 sec Time: 20:07:15

Oh, what about additional service, tea, type of car?
Time:20:08:12

What is you preference?
Process: 0.557 sec Time: 20:08:15

Please, two teas, second class car, and not the last place in the car
Time:20:09:55

We will find one accepted record at 9:45 am. Do you want something else?
Process: 0.847 sec Time: 20:09:59

No, thanks.
Time:20:10:06

Wait for checking and reserving your seats
Process: 0.659 sec Time: 20:10:09

Train N#36195, car 3, places 34,35, ADD: tea-2
Process: 0.528 sec Time: 20:10:21

Thanks!
Time:20:10:42

We will happy to help you!
Process: 0.680 sec Time: 20:10:46

Надіслати на пошту

Повторити діалог

Переглянути деталі

Набір термінів
timetable; transport; period of time; train; London; trip; ticket; morning; place; train station; tea; car; seat;

Деталі:
Визначена предметна область: [Туристична агенція](#). Тематика: [ticket buying](#), [train journey](#), [Great Britain](#).

Рисунок 3.5 – Вікно огляду сеансів користувача на панелі керування

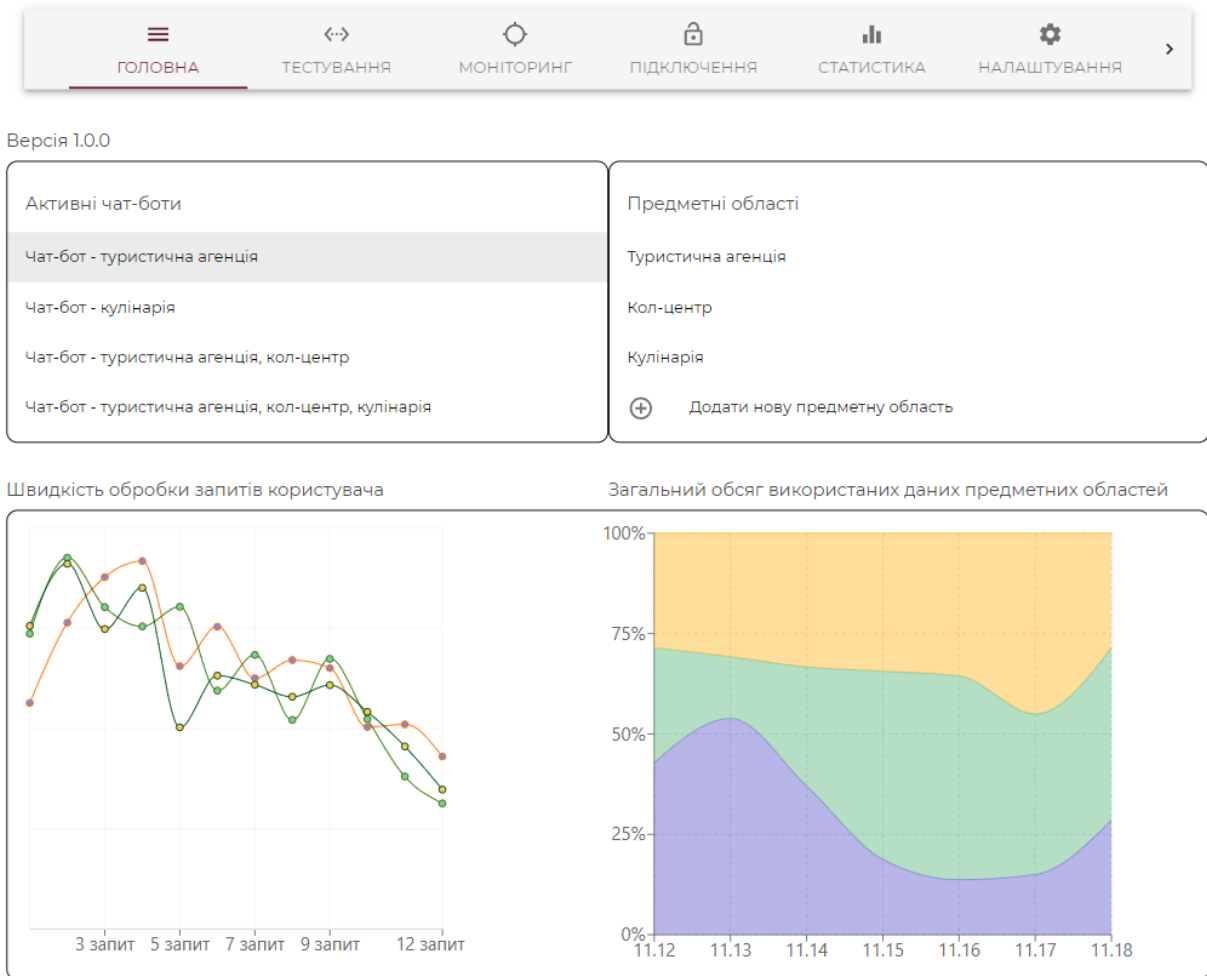
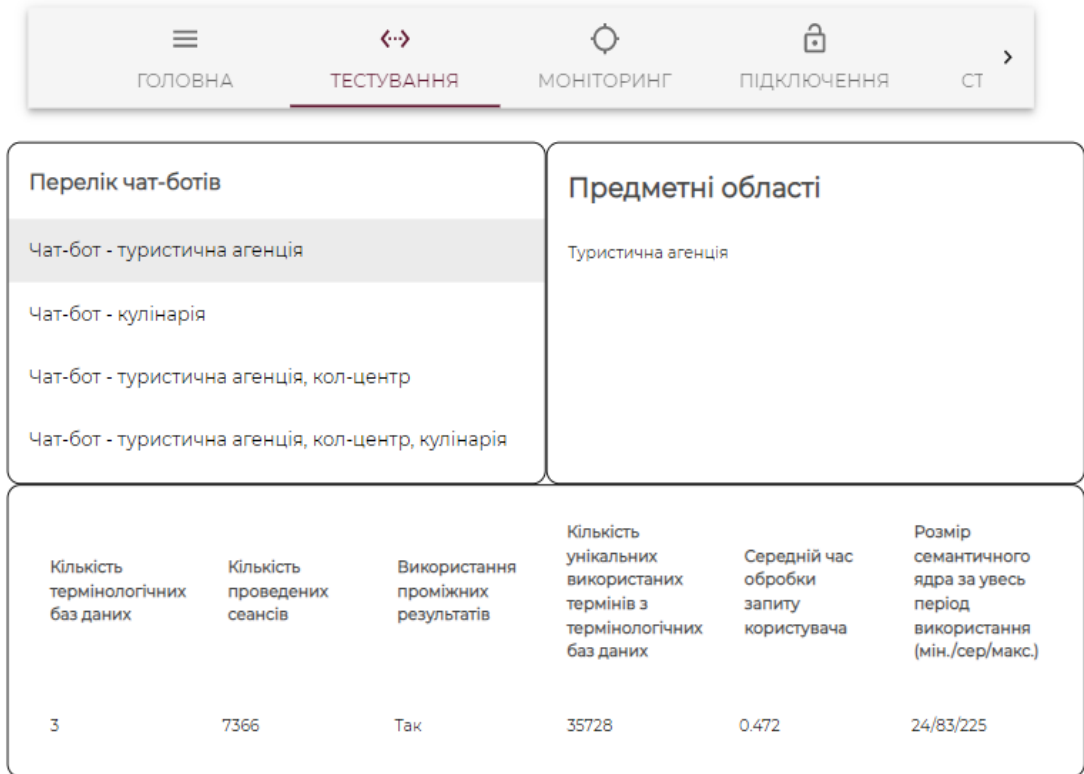


Рисунок 3.6 – Глобальна панель керування

Головна сторінка глобальної панелі керування містить перелік чат-ботів з посиланнями на сторінку їх налаштування. Також містить список доступних предметних областей та можливість додавання нової предметної області. Серед візуальних даних представлені два графіки: перший містить інформацію щодо швидкості обробки N-ого запиту користувача усіма чат-ботами, та містить три значення: максимальний час, мінімальний час та середньостатистичний час обробки N-ого запиту. Другий графік містить інформацію щодо обсягу використаних даних предметних областей за день та їх відсоткового співвідношення до загального обсягу використаних даних термінологічних баз даних.



Кількість звертань до пошукової мережі за запит користувача

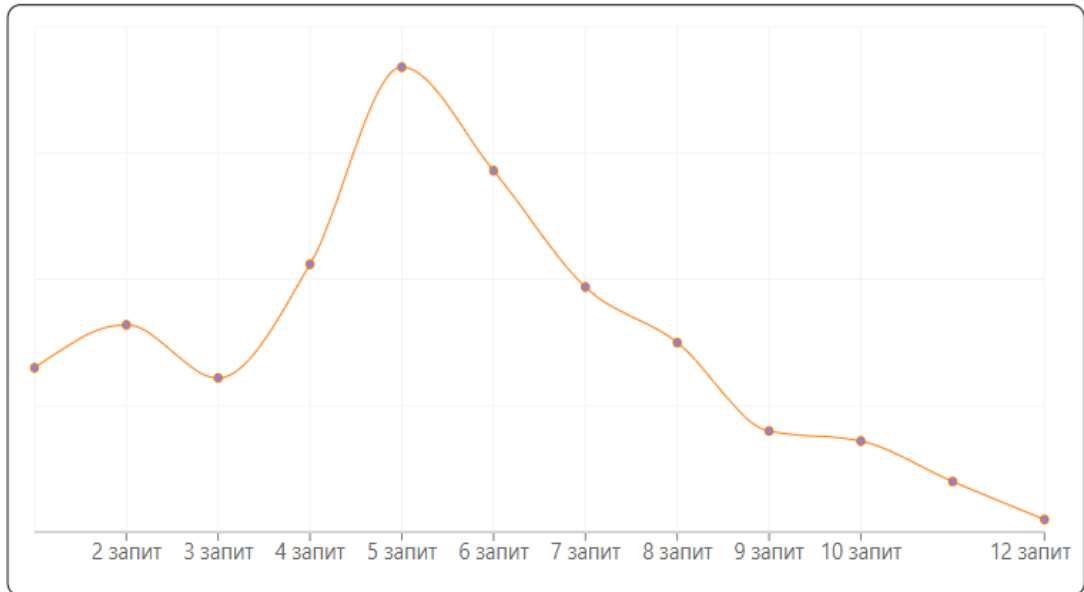


Рисунок 3.7 – Результати тестування першого чат-бота

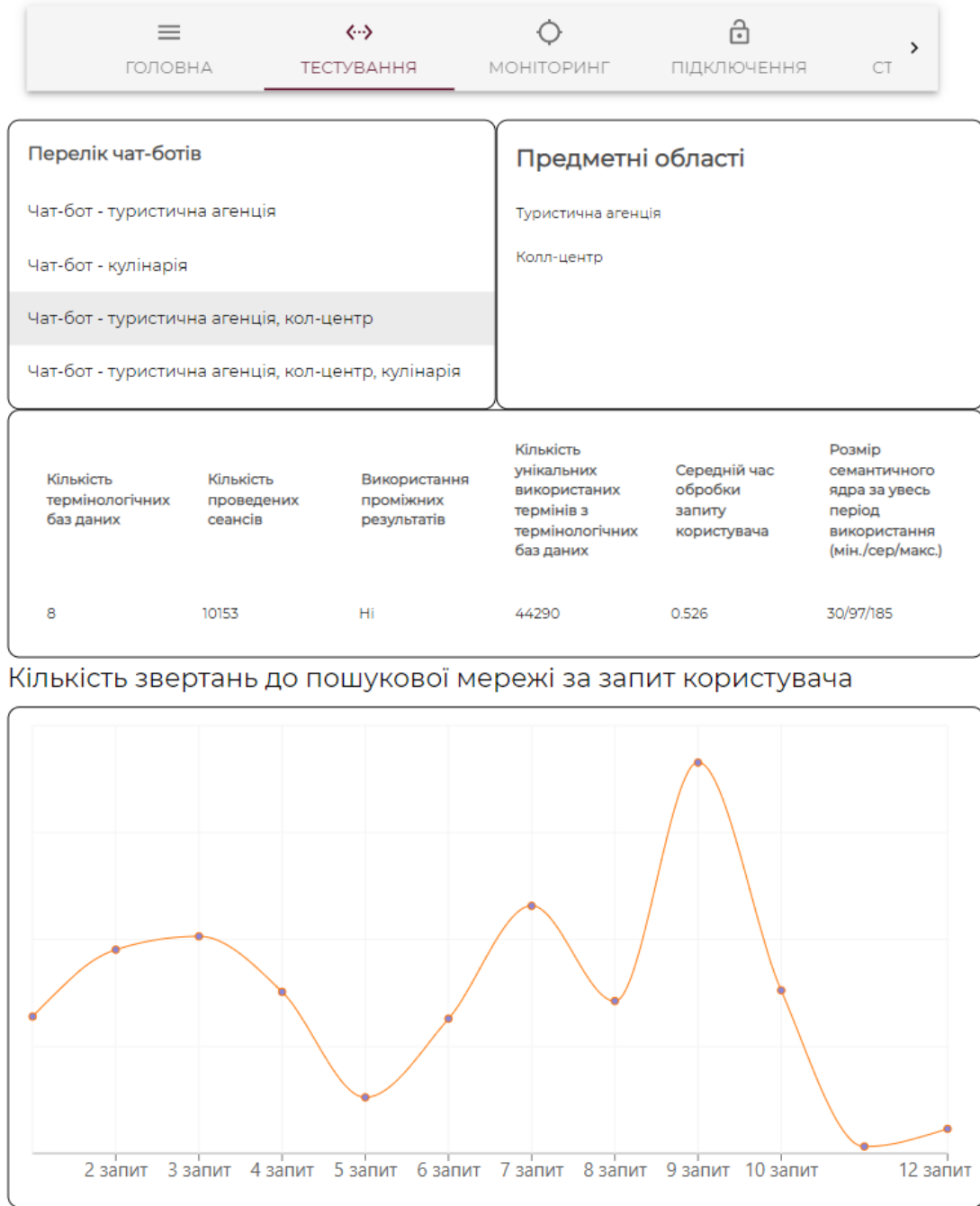


Рисунок 3.8 – Результати тестування чат-бота для двох предметних областей

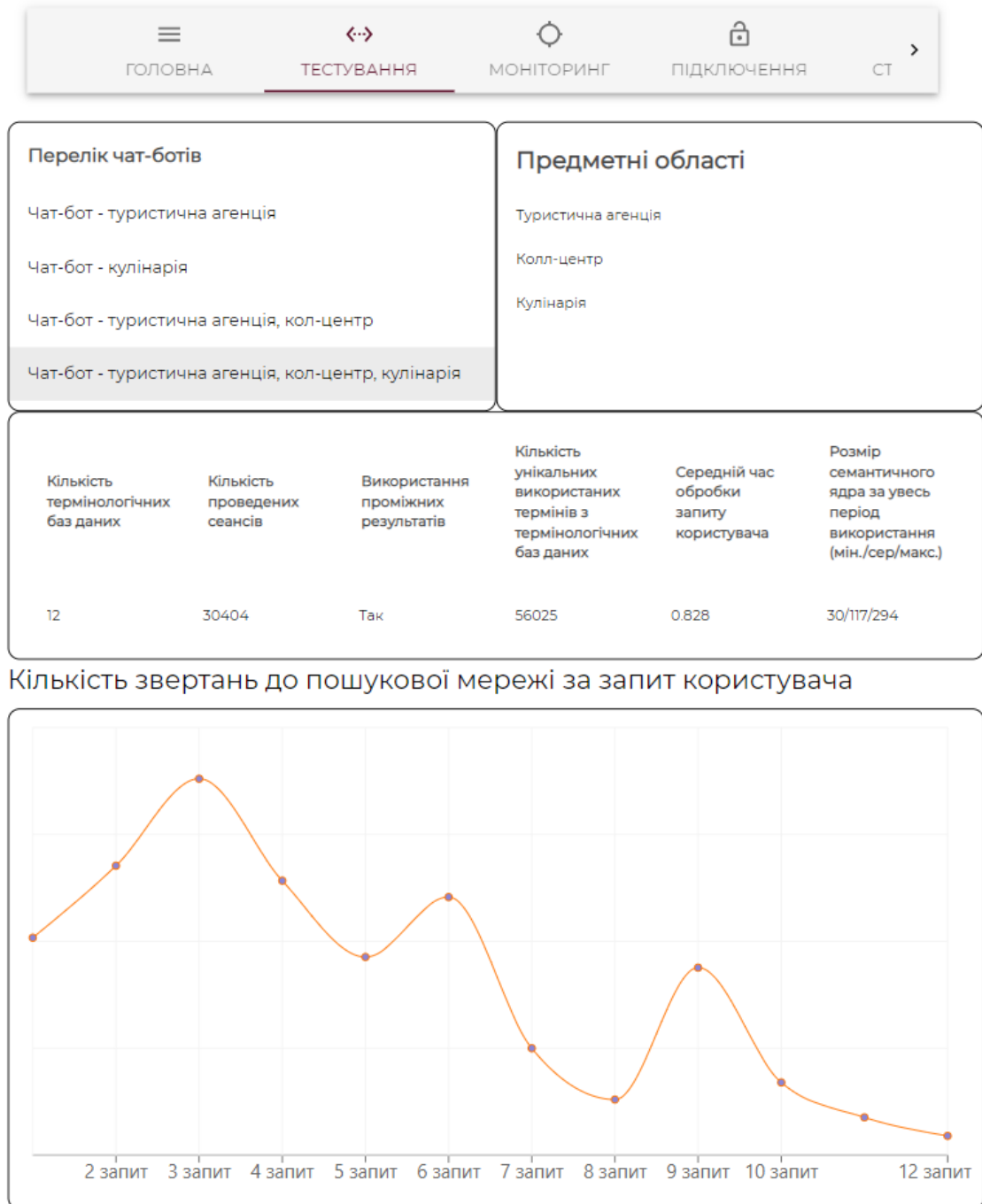


Рисунок 3.9 – Результати тестування чат-бота для трьох предметних областей

Під час тестування було виявлено приховану залежність між частотою звертань до пошукової мережі у разі відсутності даних з достатнім рівнем відповідності та використанням проміжних результатів аналізу запитів користувача. Дана залежність прослідковується на рисунках 3.7-3.9. У двох із них, графік функції має одну чітко виражену максимальну точку екстремуму до

середини сеансу, а далі спадає. В свою чергу при відкиданні проміжних результатів (рисунок 3.8 – відповідне налаштування вимкнене) чітко виражена точка максимуму відсутня.

3.5 Висновки

В третьому розділі обґрунтовано вибір програмно-апаратної платформи із застосуванням порівняльного аналізу трьох мов програмування для серверної частини та бізнес-логіки. Визначено технічну архітектуру інформаційної технології. При створенні графіків основна увага полягала у використанні інкапсуляції модулів та надання доступу через сервіси, які в свою чергу виконують перенаправлення запиту на необхідну команду. Серед представлених варіацій було обрано командну архітектуру, із записом вхідних даних та результату виконання запиту. Кожен запит повинен бути підкріплений виконаною командою або декількома командами. Під час технічної реалізації було виконано юніт-тестування серверної частини модулів за допомогою фреймворка XUnit.

Модульна структура інформаційної технології створення інтелектуального чат-бота забезпечила масштабованість та можливість до покращення кожного з модулів окремо один від одного. Для забезпечення цілісності та працездатності інформаційної технології введено версії кожного з модулів, починаючи з версії 1.0.0.

Після технічної реалізації було створено декілька чат-ботів та перевірено їх діяльність на тестових наборах даних.

Отримані результати свідчать про вплив набору даних та налаштувань параметрів нейронної мережі для кожного з чат-ботів, що спостерігається на рисунках 3.7-3.9.

Щодо формування звітності за час користування чат-ботами, то уся інформація доступна для перегляду за допомогою панелі керування чат-ботами.

Під час тестування виявлено, що оптимальним для використання є

поєднання двох предметних областей чат-ботом, при якому швидкодія розпізнавання є найкращою по відношенню до кількості предметних областей. Наявність розподілених термінологічних баз даних для предметної області значно зменшує загальний час обробки, але при зростанні їх кількості, результуюче відношення може втратити до 10% від початкового семантичного ядра тексту.

Залежність від розміру семантичного ядра під час тестування виявилась прямо-пропорційною кількості термінологічних баз даних, а також застосуванню проміжних результатів аналізу попередніх текстів в налаштуваннях чат-ботів.

Основний час обробки текстової інформації як і очікувано був витрачений на задачу кластеризації та визначення результуючої відповіді та складав до 50% від загального часу обробки.

Статистичні дані, отримані після автоматичного тестування чат-ботів на тестових вибірках, надали змогу детальніше розглянути залежності між даними та швидкодією кожного з етапів аналізу текстової інформації. В подальшому варто розширити статистику використання термінологічних баз даних, а також надання одразу декількох відповідей для кращого розуміння вибору чат-бота.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Визначення комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть к.т.н., доц. каф. КН Озеранський В.С., к.т.н., проф. каф. КН Месюра В.І. та к.т.н, доц. каф. КН Ваховська Л.М.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою [46].

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.1.

Таблиця 4.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	Озеранський В.С.	Месюра В.І.	Ваховська Л.М.
	Бали, виставлені експертами:		
1	4	3	4
2	4	4	4
3	3	4	3
4	4	4	3
5	4	4	4
6	4	3	4
7	4	4	4
8	4	3	4
9	3	4	3
10	4	3	3
11	3	3	4
12	3	4	4
Сума балів	44	43	44
Середньоарифметична сума балів СБ	$СБ = \frac{\sum_1^2 СБ_i}{2} = 44$		

Отже, з отриманих даних таблиці 4.1 варто зауважити, що дана розробка має високий рівень комерційного потенціалу.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько-технологічної роботи

Для розробки нового програмного продукту необхідно здійснити такі витрати.

Основна заробітна плата для розробки програмного засобу чи програми визначається за формулою (4.1):

$$Z_{\text{осн}} = \frac{M}{T_p} * t, \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника;

T_p – кількість робочих днів у місяці, T_p складає 22 робочих днів;

t – кількість днів праці розробника для завершення усіх робіт, t складає 60 днів.

Розрахунки заробітних плат для керівника та розробника програмного забезпечення наведено в таблиці 4.2.

Таблиця 4.2 – Розрахунок основної заробітної плати

Працівник	Оклад M , грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	6000	272,72	10	2727,2
Інженер-програміст	5500	250	60	15000
Всього:				17727,2

Після розрахунку основної заробітної плати, необхідно розрахувати додаткову:

$$З_{\text{дод}} = 0,1 * 17727,2 = 1772,72 \text{ (грн.)}$$

Нарахування на заробітну плату операторів НЗП розраховується як 22% від сумарного значення основної та додаткової заробітної плати. Дане нарахування здійснюється за формулою 4.2.

$$Н_{\text{зп}} = (З_{\text{осн}} + З_{\text{дод}}) * \frac{\beta}{100} \quad (4.2)$$

$$Н_{\text{зп}} = (17727,2 + 1772,72) * \frac{22}{100} = 4289,98 \text{ (грн.)}$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за формулою 4.3:

$$A = \frac{V * N_a}{100} * \frac{T}{12}, \text{ де} \quad (4.3)$$

V – балансова вартість обладнання, грн;

N_a – річна норма амортизаційних відрахувань (у відсотках, для програмного забезпечення 25%);

T – Термін використання ($T = 4$ місяці).

Таблиця 4.3 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
Персональний комп'ютер	15000	25	4	1250
Ноутбук	7500			625
Всього:				1875

Додатково також необхідно розрахувати витрати на комплектуючі. Для розрахунку даного типу витрат, використаємо формулу 4.4.

$$K = \sum_1^n H_i * C_i * K_i, \quad (4.4)$$

де n – кількість комплектуючих;

H_i – кількість комплектуючих i -ого виду;

C_i – ціна комплектуючих на момент придбання i -ого виду, грн;

K_i – коефіцієнт транспортних витрат (прийmemo $K_i = 1,1$).

Таблиця 4.4 – Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування комплектуючого	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флеш-карта	шт.	200	1	200
Упаковка паперу	уп.	140	1	140
Канцелярське приладдя (ручки, олівці)	уп.	25	1	25
Всього:				365

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V * \Pi * \Phi * K_{\Pi}, \quad (4.5)$$

де V – вартість 1 кВт-години електроенергії ($V = 1,7$ грн/кВт);

Π – установлена потужність комп'ютера та ноутбука ($\Pi = 1,2$ кВт);

Φ – фактична кількість годин роботи комп'ютера та ноутбука ($\Phi = 210$ год.);

K_{π} – коефіцієнт використання потужності ($K_{\pi} < 1, K_{\pi} = 0,8$).

$$V_e = 1,7 * 1,2 * 210 * 0,8 = 342,72 \text{ (грн.)}$$

Розрахуємо інші витрати $V_{ін}$. В загальному випадку, інші витрати $V_{ін}$ приймаються як (100...300)% від суми основної заробітної плати розробників та робітників, які виконували дану роботу, тобто:

$$V_{ін} = (1..3) * (Z_{осн} + Z_{дод}). \quad (4.6)$$

Отже, інші витрати становлять:

$$V_{ін} = 2 * (17727,2 + 1772,72) = 38999,84 \text{ (грн.)}$$

Загальне значення витрат, що складає суму усіх попередніх статей витрат на виконання даної частини роботи складає:

$$B = Z_{осн} + Z_{дод} + H_{зп} + A + K + V_e + V_{ін} \quad (4.7)$$

$$B = 17727,2 + 1772,72 + 4289,98 + 1875 + 365 + 342,72 + 38999,84 = 65372,46 \text{ (грн.)}$$

Розрахуємо загальну вартість наукової роботи $V_{заг}$ за формулою 4.8:

$$V_{заг} = \frac{B}{\alpha}, \quad (4.8)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відносних одиниць і дорівнює одиниці.

$$V_{заг} = \frac{65372,46}{1} = 65372,46$$

Прогнозування загальних витрат $ЗВ$ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{V_{заг}}{\beta}, \quad (4.9)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання запланованої роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{65372,46}{0,85} = 76908,77 \text{ (грн.)}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо отримання прибутку від реалізації результатів розробки. Зростання чистого прибутку варто оцінювати, враховуючи теперішню вартість грошей у національній валюті. Це забезпечить підприємству чи компанії надходження додаткових коштів, які можуть бути направлені в подальшому на покращення показників фінансової діяльності.

У випадку зростання чистого прибутку компанії $\Delta \Pi_i$ для періоду (в роках), протягом якого очікується отримання позитивного результату від впровадження розробки, розраховується за формулою 4.10:

$$\Delta \Pi_i = \sum_1^n (\Delta \Pi_{\text{я}} * N + \Pi_{\text{я}} * \Delta N)_i, \quad (4.10)$$

де $\Delta \Pi_{\text{я}}$ – покращення основного якісного показника від провадження результатів розробки у даному році;

N – основний кількісний показник, що визначає діяльність компанії у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$ – основний якісний показник, що визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від провадження розробки.

В результаті впровадження результатів наукової розробки витрати на виготовлення інформаційної технології зменшаться на 120 грн (що призведе до аналогічного зростання чистого прибутку підприємства на 120 грн), а кількість користувачів, які будуть використовувати дану інформаційну технологію збільшиться: протягом першого року – на 800 користувачів, протягом другого року – на 600, протягом третього року – на 400 користувачів. Реалізація інформаційної технології до впровадження результатів наукової розробки складала 400 користувачів, а прибуток, що отримував розробник до впровадження результатів наукової розробки – 200 грн.

Спрогнозуємо збільшення чистого прибутку від провадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого прибутку $\Delta\Pi_i$ протягом першого року складатиме:

$$\Delta\Pi_1 = 120 * 400 + (200 + 120) * 800 = 304000 \text{ грн.}$$

Протягом другого року:

$$\Delta\Pi_2 = 120 * 400 + (200 + 120) * (800 + 600) = 496000 \text{ грн.}$$

Протягом третього року:

$$\Delta\Pi_3 = 120 * 400 + (200 + 120) * (800 + 600 + 400) = 624000 \text{ грн.}$$

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $E_{\text{абс}}$ вкладених інвестицій розраховується за формулою 4.11:

$$E_{\text{абс}} = (\text{ПП} - PV), \quad (4.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник становить 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки 2, 3, 4.

Рисунок, що характеризує рух платежів у вигляді інвестицій та додаткових прибутків набуде вигляду, зображеного на рисунку 4.1.

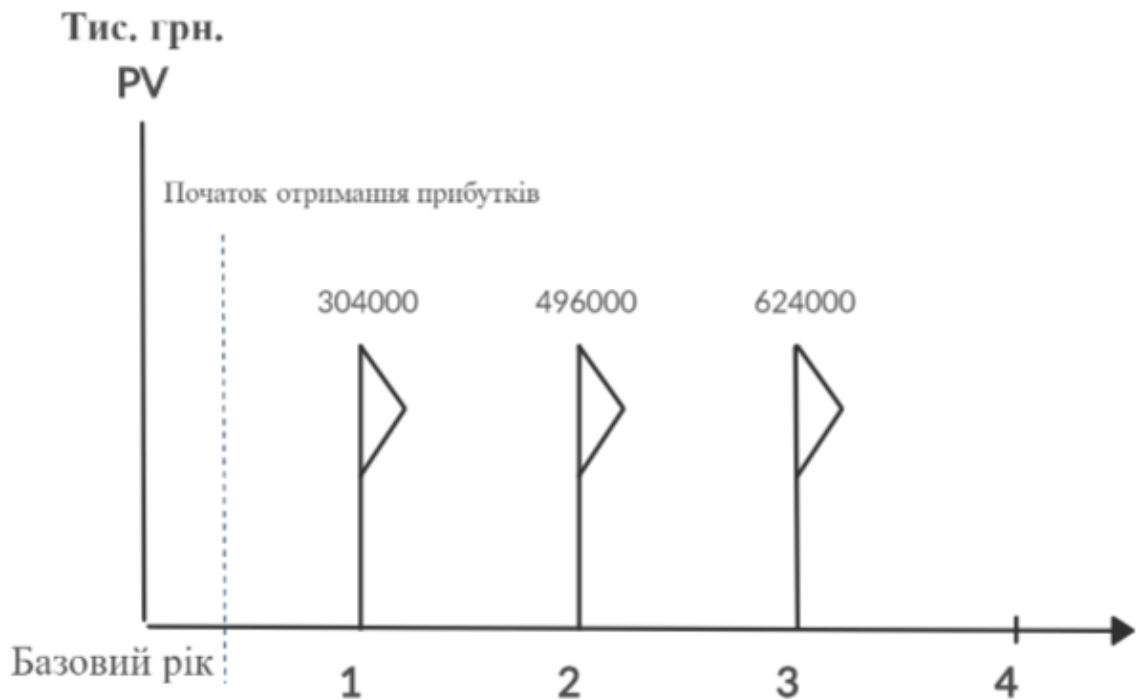


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо вартість чистих прибутків за формулою 4.12:

$$\text{ПП} = \sum_1^m \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (4.12)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

T – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник становить 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки.

Отже, розрахуємо вартість чистого прибутку:

$$\text{ПП} = \frac{76\,908,77}{(1 + 0,1)^0} + \frac{304\,000}{(1 + 0,1)^1} + \frac{496\,000}{(1 + 0,1)^2} + \frac{624\,000}{(1 + 0,1)^3} = 1\,232\,010,20 \text{ (грн.)}$$

Тоді розрахуємо $E_{абс}$:

$$E_{абс} = 1\,232\,010,20 - 76\,908,77 = 1\,155\,101,43 \text{ (грн.)}$$

Оскільки $E_{абс} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР доцільне.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_v за формулою 4.13:

$$E_v = \sqrt[T]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.13)$$

де $E_{абс}$ – абсолютна ефективність вкладених в наукову розробку інвестицій, грн;

PV – теперішня вартість інвестицій $PV = ZB$, грн;

$T_{ж}$ – життєвий цикл наукової роботи, роки.

В результаті, одержимо:

$$E_v = \sqrt[3]{1 + \frac{1\,155\,101,41}{76\,908,77}} - 1 = 1,4621 \text{ або } 146,21\%$$

Використовуючи розраховану величину відносної ефективності, порівняємо її з мінімальною (бар'єрною) ставкою дисконтування τ_{min} , яка визначає мінімальну дохідність, нижче за яку, інвестиції вкладати немає сенсу.

У загальному вигляді (бар'єрна) ставка дисконтування τ_{min} визначається за формулою 4.14:

$$\tau = d + f, \quad (4.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні $d = 0,2$;

f – показник, що характеризує ризикованість вкладень, величина $f = 0,1$.

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки $E_B = 146,21\% > \tau_{min} = 0,3 = 30\%$, то інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ розраховується за формулою 4.15:

$$T_{ок} = \frac{1}{E_B}, \quad (4.15)$$

$$T_{ок} = \frac{1}{1,4621} = 0,684 \text{ року}$$

Визначивши термін окупності даної наукової розробки, доцільно зробити висновок, що фінансування даної наукової розробки буде перспективним.

4.5 Висновок

В даному розділі було проведено економічне обґрунтування доцільності розробки програми класифікації банківських текстів, яка створена у результаті науково-технічної діяльності. Незалежними експертами було здійснено оцінювання комерційного потенціалу розробки, за результатами якого було визначено, що нова розробка має високий рівень комерційного потенціалу, оскільки кожен із експертів виставив понад 40 балів. Також було виконано прогнозування витрат на виконання розробки, де розраховано основну заробітну

плату кожного із розробників, додаткову заробітну плату всіх розробників, нарахування на заробітну плату, амортизацію обладнання, комп'ютерів та приміщень, витрати на допоміжні матеріали, витрати на силову електроенергію тощо. Загальна сума витрат на виконання означених робіт склала 76 908,77 грн. Визначено, що абсолютна ефективність вкладених інвестицій становить 1155101,41 грн, і це свідчить про те, що вкладання коштів на виконання та впровадження результатів НДДКР є доцільним та вигідним. Було розраховано відносну (щорічну) ефективність вкладених в наукову розробку інвестицій – 146,21%, її величина більша за мінімальну (бар'єрну) ставку дисконтування, отже інвестор буде зацікавлений у фінансуванні даної наукової розробки та матиме змогу отримувати досить високий прибуток, починаючи вже з першого року впровадження. Проведено розрахунок терміну окупності, який складає 0,683 року. В загальному варто зробити висновок, що фінансування розробки інформаційної технології створення інтелектуального чат-бота, яка створена у результаті науково-технічної діяльності є конкурентоспроможною та здатна надавати прибуток уже на восьмий місяць після її впровадження.

ВИСНОВКИ

В результаті здійсненого дослідження було розроблено інформаційну технологію створення інтелектуального чат-бота. Проаналізовано предметну область інтелектуального аналізу тексту та його застосування у сучасних інформаційних системах. Обґрунтовано доцільність використання технологій штучного інтелекту для задачі аналізу тексту, а саме класифікації, агрегації та кластеризації. Описано процес попереднього аналізу тексту та удосконалено метод семантичного аналізу тексту для покращення якості класифікації та визначення тематики текстової інформації.

В першому розділі проаналізовано використання чат-ботів за функціональними та технічними характеристиками, а також сформовано вимоги до створення інформаційної технології створення інтелектуального чат-бота. Завдяки необхідності створення платформи для керування чат-ботами охарактеризовано прикладні аспекти застосування чат-ботів, а також використання технологій та засобів для їх діяльності. Здійснено порівняльний аналіз чат-боту, створеного на базі розроблюваної інформаційної технології, із чат-ботом, створеним на базі фреймворку Microsoft Bot Framework.

В другому розділі виконано аналіз предметної області інтелектуальної технології створення інтелектуального чат-бота. Розроблено структурну схему інформаційної технології. Визначено та проаналізовано усі етапи обробки текстової інформації. Запропоновано та обґрунтовано використання удосконаленого методу семантичного аналізу тексту, використання штучної рекурентної нейронної мережі для задачі класифікації текстової інформації. Серед розглянутих алгоритмів та методів аналізу текстової інформації були відзначені алгоритми: технології машинного навчання та нейронні мережі у особливості рекурентні із-за наявності зворотного поширення сигналу, що притаманне для обробки текстової інформації та задачі класифікації. Розроблено структуру інформаційної технології інтелектуального чат-бота. Інформаційна технологія складається з шести компонентів: репозиторій сховища даних, модуль глобального керування, модуль попередньої обробки, модуль

інтелектуального аналізу, модуль операційного контролю, модуль введення/виведення інформації. Для задачі кластеризації тексту було обрано нейронну мережу Кохонена завдяки можливості найшвидшого перенавчання для підвищення рівня кластеризації.

В третьому розділі обґрунтовано вибір програмно-апаратної платформи із застосуванням порівняльного аналізу трьох мов програмування для серверної частини та бізнес-логіки. Визначено технічну архітектуру інформаційної технології. Щодо формування звітності за час користування чат-ботами, то уся інформація доступна для перегляду за допомогою панелі керування чат-ботами.

Під час тестування виявлено, що оптимальним для використання є поєднання двох предметних областей чат-ботом, при якому швидкодія розпізнавання є найкращою по відношенню до кількості предметних областей. Наявність розподілених термінологічних баз даних для предметної області значно зменшує загальний час обробки, але при зростанні їх кількості, результуюче відношення може втратити до 10% від початкового семантичного ядра тексту.

Залежність від розміру семантичного ядра під час тестування виявилась прямо-пропорційною кількості термінологічних баз даних, а також застосуванню проміжних результатів аналізу попередніх текстів в налаштуваннях чат-ботів.

Основний час обробки текстової інформації як і очікувано був витрачений на задачу кластеризації і визначення результуючої відповіді та складав до 50% від загального часу обробки.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. How computers change the world [Електронний ресурс]: techspirited.com / Доступ: <https://techspirited.com/how-computers-changed-world> – Назва з екрану
2. Data Processing [Електронний ресурс]: planningtank.com / Доступ: <https://planningtank.com/computer-applications/data-processing-data-processing-methods> – Назва з екрану
3. Удосконалення методу семантичного аналізу тексту / А. А. Яровий, Д. С. Кудрявцев, Л.В. Крилик // «ІНТЕРНЕТ-ОСВІТА-НАУКА-2020»: Збірник праць конференції. – Вінниця : ВНТУ, 2020. – С. 2-4.
4. Yarovii A., Kudriavtsev D., Prozor. O. / Improving the accuracy of text message recognition with an intelligent chat-bot information system [Електронний ресурс]: XV міжнародна науково-технічна конференція комп'ютерних наук CSIT-2020 / Доступ: <http://csit.lp.edu.ua/> – Назва з екрану
5. Результати конкурсу студентських наукових робіт з напрямку «Інформатика та кібернетика» [Електронний ресурс]: impuls.vntu.edu.ua
Доступ: http://impuls.vntu.edu.ua/index.php?option=com_content&view=article&id=6714%3A2020-06-16-10-59-14&catid=9%3A2014-02-09-08-50-23&Itemid=1&lang=en – Назва з екрану
6. Кудрявцев Д. С., Яровий А. А., «ЧАТ-БОТ ЯК СИСТЕМА ІНТЕЛЕКТУАЛЬНОЇ ВЗАЄМОДІЇ» в Матеріали конференції «XLVIII Науково-технічна конференція підрозділів Вінницького національного технічного університету (2019)», Вінниця, 2019. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/index/pages/view/zbirn2019> – Назва з екрану
7. Д. С. Кудрявцев, В. В. Колодний, «Комп'ютерна програма “Інформаційна технологія візуального моделювання тернарних гештальт-ранжувань на крузі переваг» (ТеГР)” Свідоцтво про реєстрацію авторського права на твір № 78345. Дата реєстрації 17.04.2018.
8. Andrii Yaroviyi, Dmytro Kudriavtsev, Serhii Varaban, Volodymyr Ozeranskyi, Liudmyla Krylyk, Andrzej Smolarz, Gayni Karnakova, "Information technology in creating intelligent chatbots," Proc. SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2019, 1117627 (6 November 2019); doi: 10.1117/12.2537415.
9. A. Volovyk, V. Kychak, D. Kudriavtsev, D. Havrilov, A. Yaroviyi and L. Krylik, "Simultaneous Estimation in Linear Dynamic Systems with the Indeterminate Structure Disturbances", 2020 IEEE 40th International Conference on

- Electronics and Nanotechnology (ELNANO), pp. 651-655, 2020.
10. В. Колодний і Д. Кудрявцев, ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВІЗУАЛЬНОГО МОДЕЛЮВАННЯ ТА ОБРОБКИ ТЕРНАРНИХ ГЕШТАЛЬТ-РАНЖУВАНЬ, ІТКІ, vol 42, № 2, с. 26-34, Жов 2018.
 11. NLP Processing [Електронний ресурс]: towardsai.net / Доступ: <https://towardsai.net/p/nlp/natural-language-processing-nlp-with-python-tutorial-for-beginners-1f54e610a1a0> – Назва з екрану
 12. Recurrent Neural Networks and-LSTM [Електронний ресурс]: Towardsdatascience / Доступ: <https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5> – Назва з екрану
 13. Chatbot is [Електронний ресурс]: expertsystem.com / Доступ: <https://www.expertsystem.com/chatbot/> – Назва з екрану
 14. Maintaining Context in Chatbot [Електронний ресурс]: Chatbots magazine / Доступ: <https://chatbotsmagazine.com/maintaining-context-in-chatbots-2016b6a5b7c6> – Назва з екрану
 15. The multiple dimensions of information quality [Електронний ресурс]: Research Gate article / Доступ: https://www.researchgate.net/publication/242929284_The_Multiple_Dimensions_of_Information_Quality – Назва з екрану
 16. Top 26 Chatbot Builders [Електронний ресурс]: cmswire.com / Доступ: <https://www.cmswire.com/information-management/top-26-chatbot-builders-for-2019/> – Назва з екрану
 17. Beginner's Guide to Computer Vision [Електронний ресурс]: medium.com / Доступ: <https://medium.com/readers-writers-digest/beginners-guide-to-computer-vision-23606224b720> – Назва з екрану
 18. Chatbot data analyzing [Електронний ресурс]: Blog appliedai / Доступ: <https://blog.appliedai.com/chatbot-analytics/> – Назва з екрану
 19. What is machine translation? [Електронний ресурс]: systransoft.com / Доступ: <http://www.systransoft.com/systran/translation-technology/what-is-machine-translation/> – Назва з екрану
 20. Google DialogFlow [Електронний ресурс]: dialogflow.cloud.google.com / Доступ: <https://cloud.google.com/dialogflow/docs> – Назва з екрану
 21. Recurrent Neural Networks [Електронний ресурс]: ResearchGate / Доступ: https://www.researchgate.net/publication/338723814_Fundamentals_of_Recurrent_Neural_Network_RNN_and_Long_Short-Term_Memory_LSTM_network – Назва з екрану
 22. Recurrent Neural Networks and LSTM [Електронний ресурс]: ResearchGate / Доступ: https://www.researchgate.net/publication/13853244_Long_Short-term_Memory – Назва з екрану

23. Neural Networks behind Google Voice [Електронний ресурс]: Google blog / Доступ: <https://ai.googleblog.com/2015/08/the-neural-networks-behind-google-voice.html> – Назва з екрану
24. Нейронна мережа Кохонена [Електронний ресурс]: Data mining / Доступ: <https://ranalytics.github.io/data-mining/105-Cohonen-Maps.html> – Назва з екрану
25. Google Custom Search API [Електронний ресурс] : Google APIs / Доступ: <https://developers.google.com/custom-search> – Назва з екрану
26. Azure BING Custom Search [Електронний ресурс]: Azure platform Microsoft / Доступ: <https://azure.microsoft.com/en-us/services/cognitive-services/bing-custom-search/> – Назва з екрану
27. Yahoo BOSS search engine [Електронний ресурс]: Yahoo services / Доступ: <https://developer.yahoo.com/boss/search/> – Назва з екрану
28. NLogger library [Електронний ресурс]: nuget.org / Доступ: <https://github.com/NLog/NLog/wiki> – Назва з екрану
29. My SQL vs Mongo DB [Електронний ресурс]: db-engines.com / Доступ: <https://db-engines.com/en/system/Microsoft+SQL+Server%3BMongoDB> – Назва з екрану
30. Firebird DB vs Mongo DB [Електронний ресурс]: db-engines.com / Доступ: <https://db-engines.com/en/system/Firebird%3BMongoDB> – Назва з екрану
31. Firebird DB vs My SQL [Електронний ресурс]: db-engines.com / Доступ: <https://db-engines.com/en/system/Firebird%3BMicrosoft+SQL+Server> – Назва з екрану
32. Microsoft Visual Studio [Електронний ресурс]: G2 / Доступ: <https://www.g2.com/products/visual-studio/reviews> – Назва з екрану
33. Comparison of programming languages [Електронний ресурс]: ThoughtCo / Доступ: <https://www.thoughtco.com/comparing-popular-programming-languages-958275> – Назва з екрану
34. C# Main features [Електронний ресурс]: sharpcorner.com / Доступ: <https://www.c-sharpcorner.com/article/C-Sharp-and-its-features> – Назва з екрану
35. React framework [Електронний ресурс]: reactjs.org / Доступ: <https://reactjs.org/docs/getting-started.html> – Назва з екрану
36. .Net Core 3.1 [Електронний ресурс]: docs.microsoft.com / Доступ: <https://docs.microsoft.com/en-us/dotnet/fundamentals> – Назва з екрану
37. Swagger [Електронний ресурс]: docs.microsoft.com / Доступ: <https://docs.microsoft.com/en-us/aspnet/core/tutorials/web-api-help-pages-using-swagger> – Назва з екрану
38. Pandas [Електронний ресурс]: Pandas офіційний сайт / Доступ: <https://pandas.pydata.org> – Назва з екрану

39. MediatR pattern [Електронний ресурс]: Medium.com / Доступ: <https://medium.com/@ducmeit/net-core-using-cqrs-pattern-with-mediatr-part-1-55557e90931b> – Назва з екрану
40. MediatR pattern for C# [Електронний ресурс]: docs.microsoft.com / Доступ: <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/microservice-application-layer-implementation-web-api> – Назва з екрану
41. HtmlAgilityPack [Електронний ресурс]: HtmlAgilityPack офіційний сайт / Доступ: <https://html-agility-pack.net> – Назва з екрану
42. Entity Framework Core [Електронний ресурс]: docs.microsoft.com / Доступ: <https://docs.microsoft.com/en-us/ef> – Назва з екрану
43. xUnit framework [Електронний ресурс]: xUnit офіційний сайт / Доступ: <https://xunit.net> – Назва з екрану
44. Microsoft Bot Framework [Електронний ресурс]: Microsoft Bot Framework офіційний сайт / Доступ: <https://dev.botframework.com> – Назва з екрану
45. MongoDB Aggregation Framework [Електронний ресурс]: docs.mongodb.com / Доступ: <https://docs.mongodb.com/manual/aggregation> – Назва з екрану
46. Методичні вказівки до виконання студентами-магістрантами наукового напрямку економічної частини магістерських кваліфікаційних робіт / Уклад В.О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.