

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ПОГОДЖЕНО
Декан факультету
інформаційних технологій
(назва факультету (ННІ))
Болбот І.М.
(підпис) (ПІБ)

“ ” _____ 20_ р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри
комп'ютерних наук
(назва кафедри)
Голуб Б.Л.
(підпис) (ПІБ)

“ ” _____ 20_ р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему Система аналізу запитів доступу до інформації на складних веб-ресурсах

Спеціальність 122 Комп'ютерні науки
(код і найменування)

Освітня програма Інформаційні управляючі системи і технології
(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

К.Т.Н., доцент
(науковий ступінь та вчене звання) (підпис)

Голуб Б.Л.
(ПІБ)

Керівник магістерської кваліфікаційної роботи

К.Т.Н., доцент
(науковий ступінь та вчене звання) (підпис)

Вайганг Г.О.
(ПІБ)

Виконав _____
(підпис)

Колесник Д.Ю.
(ПІБ студента)

КИЇВ – 2025

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

доцент, к.т.н.

(науковий ступінь, вчене звання)

(підпис)

Голуб Б.Л.

(ПІБ)

“ 10 ” жовтня 2025 року

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Колесник Дарині Юрійвні

(прізвище, ім'я, по батькові)

Спеціальність 122 “Комп'ютерні науки”

(код і найменування)

Освітня програма Інформаційні управляючі системи і технології

(назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Тема магістерської кваліфікаційної роботи Система аналізу запитів доступу до інформації на складних веб-ресурсах

затверджена наказом проректора НУБіП України від “ 10 ” жовтня 2025 р. № 2290 “С”
Термін подання завершеної роботи на кафедру 1 грудня 2025 р.

Вихідні дані до магістерської кваліфікаційної роботи: сучасні тенденції розвитку веб-ресурсів із високою інформаційною насиченістю; наукові публікації та практичні розробки у сфері Data Mining, Machine Learning та інтелектуальних систем підтримки користувачів; нормативно правові акти та вимоги до систем обробки та аналізу користувацьких запитів; дані з відкритих датасетів, база даних ChatbotDB, яка містить інформацію про запити користувачів, типи відповідей, оцінку ефективності та середній час відповіді.

Перелік питань, що підлягають дослідженню:

1. Аналіз предметної області та наявних підходів до пошуку інформації у складних веб-ресурсах
2. Моделювання та проектування інтелектуальної системи аналізу запитів користувачів
3. Архітектура та програмна реалізація інтелектуальної системи аналізу запитів
4. Оцінювання результатів роботи системи та аналіз ефективності її функціонування

Перелік графічного матеріалу (за потреби) презентація, постер, схема архітектури системи, діаграма сховища даних, РСА-візуалізація кластерів, графік пошуку оптимального К (Elbow method)

Дата видачі завдання “ 10 ” жовтня 2025 р.

Керівник магістерської кваліфікаційної роботи _____

(підпис)

Вайганг Г. О.

(прізвище та ініціали)

Завдання прийняв до виконання _____

Колесник Д.Ю.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП	7
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА НАЯВНИХ ПІДХОДІВ ДО ПОШУКУ ІНФОРМАЦІЇ У СКЛАДНИХ ВЕБ-РЕСУРСАХ	12
1.1 Загальні принципи пошуку та аналізу інформації на складних веб-ресурсах	12
1.2 Аналіз вимог до системи аналізу запитів користувачів	14
1.3 Структура інформаційних джерел для інтелектуального аналізу	18
1.4 Огляд сучасних інструментів і систем підтримки користувачів	22
1.5. Проблеми та обмеження класичних методів пошуку інформації	26
1.6 Постановка завдання для розробки системи підтримки користувачів на основі мовних моделей	30
Висновки до розділу 1	34
2. МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АНАЛІЗУ ЗАПИТІВ КОРИСТУВАЧІВ	35
2.1 Моделювання предметної області	35
2.2 Моделювання динаміки процесів у системі	38
2.3 Діаграми послідовності взаємодії компонентів системи	41
2.4. Концептуальне та логічне проєктування сховища даних інтелектуальної системи	48
Висновки до розділу 2	52
3. АРХІТЕКТУРА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АНАЛІЗУ ЗАПИТІВ	54
3.1 Архітектура системи інтелектуального аналізу	54
3.2 Розробка бази даних для зберігання інформації про запити користувачів	57
3.3 Розробка сховища даних для аналітичної обробки інформації	60
3.4 Алгоритмізація та програмування програмних модулів	63
Висновки до розділу 3	67

4. ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ РОБОТИ СИСТЕМИ ТА АНАЛІЗ ЕФЕКТИВНОСТІ ЇЇ ФУНКЦІОНУВАННЯ	69
4.1. Методи та реалізація модулів семантичного аналізу користувацьких запитів	69
4.2. Оцінювання ефективності моделей у системі семантичного аналізу запитів	72
4.3. Методика тестування та експериментальні результати	86
4.4. Оцінка ефективності роботи системи та рекомендації щодо впровадження	89
Висновки до розділу 4	91
ВИСНОВКИ	92
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	94
ДОДАТКИ	97

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- ML (Machine Learning) – машинне навчання; підхід до побудови моделей, що навчаються на даних.
- NLP (Natural Language Processing) – обробка природної мови; методи аналізу, розуміння та генерації текстів.
- IR (Information Retrieval) – інформаційний пошук; методи пошуку релевантної інформації у великих масивах даних.
- LLM (Large Language Model) – велика мовна модель на архітектурі Transformer.
- BM25 – лексичний алгоритм ранжування документів.
- TF-IDF (Term Frequency – Inverse Document Frequency) – статистична міра вагомості термінів у тексті.
- Embedding – векторне представлення тексту у багатовимірному просторі.
- Semantic Search – семантичний пошук за змістовою подібністю текстів.
- Intent – намір користувача, визначений на основі інтерпретації запиту.
- K-Means – алгоритм кластеризації методом k-середніх.
- PCA (Principal Component Analysis) – метод головних компонент для зменшення розмірності даних.
- Silhouette Score – метрика оцінки якості кластеризації.
- 1R (One Rule) – простий алгоритм класифікації на основі одного правила.
- Naive Bayes – наївний баєсівський класифікатор.
- Apriori – алгоритм побудови асоціативних правил.
- ETL (Extract, Transform, Load) – процедура вилучення, перетворення та завантаження даних.
- DWH (Data Warehouse) – сховище даних; централізована система зберігання та аналітики.
- SQL (Structured Query Language) – мова структурованих запитів до БД.
- SSIS (SQL Server Integration Services) – серверний інструмент створення ETL-процесів.
- SSAS (SQL Server Analysis Services) – сервіс багатовимірного аналізу даних.

API (Application Programming Interface) – прикладний програмний інтерфейс.

JSON (JavaScript Object Notation) – формат обміну структурованими даними.

UI (User Interface) – інтерфейс користувача.

UX (User Experience) – користувацький досвід взаємодії з системою.

OCR (Optical Character Recognition) – оптичне розпізнавання символів.

HTTP(S) (HyperText Transfer Protocol Secure) – протокол передавання гіпертексту (з підтримкою шифрування).

DB (Database) – база даних.

NER (Named Entity Recognition) – розпізнавання іменованих сутностей.

Tokenizer – компонент NLP для поділу тексту на токени.

Vector Index / ANN (Approximate Nearest Neighbours) – методи наближеного пошуку найближчих векторів.

GPU (Graphics Processing Unit) – графічний процесор.

REST API (Representational State Transfer API) – архітектурний стиль веб-сервісів.

Latency – затримка у часі відповіді системи.

ВСТУП

Актуальність теми. У сучасних умовах цифрової трансформації обсяг інформації, що накопичується у веб-ресурсах державних, освітніх та корпоративних структур, зростає непропорційно швидко. Дані таких систем характеризуються значною структурною різноманітністю, багатоформатністю, нерівномірною якістю метаданих і складністю навігації, що істотно ускладнює їх ефективне використання [1]. Традиційні інформаційно-пошукові системи, засновані на лексичному зіставленні термінів (зокрема TF-IDF і BM25), забезпечують прийнятну ефективність лише для формально сформульованих запитів. Проте більшість користувацьких звернень у практиці взаємодії з веб-ресурсами становлять короткі та неструктуровані запити природною мовою, що містять синонімію, полісемію та предметно-специфічні конструкції [2].

Виявлена проблема особливо відчутна у великих освітніх і державних веб-ресурсах, де інформація фрагментована між численними розділами, локальними нормативними документами, регламентами, навчально-методичними матеріалами і внутрішніми службовими інструкціями [3]. За відсутності семантичного аналізу змісту такі ресурси демонструють низький рівень релевантності пошукової видачі, що призводить до втрати часу, повторних звернень та зростання навантаження на служби підтримки.

Суттєвий прогрес у галузі обробки природної мови (NLP) і глибинного навчання створює підґрунтя для переходу від лексичного до семантичного пошуку. Моделі на базі архітектури Transformer, включно з великими мовними моделями (LLM), забезпечують можливість формувати векторні подання текстів, інтерпретувати смислові зв'язки, визначати намір користувача та узгоджувати семантику запиту з контентом [4]. Застосування embedding-моделей, алгоритмів класифікації та кластеризації, а також методів Data Mining відкриває можливості для створення інтелектуальних систем аналізу запитів, здатних підтримувати користувачів у складних предметних доменах [5].

У контексті функціонування університетських та державних веб-ресурсів такі системи є важливим інструментом підвищення якості доступу до інформації. Інтелектуальна система аналізу запитів користувачів дозволяє поєднати семантичний пошук, аналіз контенту та методи машинного навчання з метою формування точних і релевантних відповідей у динамічному інформаційному середовищі [6]. Отже, розроблення інтегрованої системи, здатної аналізувати структуру контенту та інтерпретувати користувацькі наміри, є актуальним науково-прикладним завданням.

Об'єкт дослідження – процеси пошуку, обробки та представлення інформації на складних веб-ресурсах з використанням інтелектуальних методів аналізу запитів і контенту.

Предмет дослідження – методи та технології аналізу запитів користувачів із застосуванням мовних моделей і алгоритмів обробки природної мови, а також моделі організації сховища даних і інтеграції контент-аналізу для забезпечення ефективного доступу до інформаційних ресурсів.

Мета дослідження полягає у розробленні інтелектуальної системи аналізу запитів користувачів для забезпечення ефективного доступу до інформації на складних веб-ресурсах на основі технологій обробки природної мови, семантичного пошуку та аналізу контенту.

Для досягнення поставленої мети необхідно розв'язати такі основні **завдання**:

- 1) виконати огляд сучасних методів обробки природної мови, семантичного пошуку та контент-аналізу в задачах доступу до інформації на складних веб-ресурсах;
- 2) проаналізувати існуючі архітектури інтелектуальних систем пошуку й підтримки користувачів та визначити вимоги до цільової системи;
- 3) сформулювати модель предметної області та розробити логічну структуру бази даних і сховища даних для зберігання інформації про запити, контент і результати обробки;

- 4) спроектувати архітектуру інтелектуальної системи аналізу запитів із виділенням модулів семантичного аналізу, класифікації, пошуку та аналітичної обробки даних;
- 5) реалізувати програмні модулі системи з використанням сучасних технологій NLP, машинного навчання та інструментів Data Mining;
- 6) провести експериментальні дослідження, оцінити ефективність роботи системи за ключовими показниками (точність, повнота, релевантність, час відповіді) на тестових вибірках запитів;
- 7) розробити методичні рекомендації щодо впровадження створеної системи в інформаційну інфраструктуру освітніх та інших складних веб-ресурсів.

Методи дослідження. У роботі застосовано методи системного аналізу для формування вимог і побудови архітектури системи; методи обробки природної мови (токенізація, лематизація, визначення частин мови, синтаксичний та семантичний аналіз) для інтерпретації текстових запитів і документів; технології векторного подання текстів (embedding-моделювання) для представлення запитів і контенту у багатовимірному просторі ознак; методи машинного навчання та Data Mining для класифікації запитів, групування тематично споріднених звернень і виявлення закономірностей у використанні веб-ресурсу; алгоритми семантичного пошуку для порівняння векторних представлень запитів і документів; статистичні методи аналізу для оцінювання якості роботи системи за кількісними метриками.

Наукова новизна одержаних результатів полягає в такому:

- уточнено та формалізовано підхід до інтеграції мовних моделей і контент-аналізу в єдиній архітектурі інтелектуальної системи аналізу запитів користувачів складних веб-ресурсів;
- розроблено архітектуру системи семантичного пошуку, що поєднує класичні пошукові індекси з нейромережевими векторними поданнями текстів і компонентами інтелектуальної обробки запитів;

- удосконалено методику визначення семантичної подібності між запитами та контентом за рахунок поєднання embedding-моделей і аналітичних показників, що дозволяє підвищити релевантність пошукової видачі;
- запропоновано підхід до адаптації мовних моделей до предметних доменів освітніх та державних веб-ресурсів шляхом урахування структурних особливостей контенту й специфіки користувацьких запитів;
- сформовано модель бази даних і сховища даних для накопичення та аналітичної обробки інформації про запити користувачів і результати взаємодії з веб-ресурсом, що забезпечує можливість подальшого розширення функцій системи.

Практичне значення одержаних результатів полягає у створенні програмної реалізації прототипу інтелектуальної системи аналізу запитів, яка може застосовуватися для оптимізації доступу до інформації на складних веб-ресурсах, підвищення релевантності результатів, зменшення навантаження на служби підтримки та вдосконалення цифрової взаємодії користувачів із інформаційними системами.

Апробація результатів. Основні положення та результати дослідження доповідалися і обговорювалися на **XVI Міжнародній науково-практичній конференції молодих учених «Інформаційні технології: економіка, техніка, освіта»**, де представлено тези «Система аналізу запитів доступу до інформації на складних веб-ресурсах з аналізом контенту» (Д. Ю. Колесник, Г. О. Вайганг); на **VII Всеукраїнській науково-практичній інтернет-конференції студентів і аспірантів «Теоретичні та прикладні аспекти розробки комп'ютерних систем '2025»**, де опубліковано тези «Модель бази даних для інтелектуальної системи підтримки користувачів складних веб-ресурсів» (Д. Ю. Колесник); а також на **VI Міжнародній науково-практичній конференції молодих учених, аспірантів і студентів «Сучасні інформаційні технології та системи в управлінні»**, де подано результати щодо формування архітектури інтелектуальної системи пошуку на складних веб-ресурсах (Г. О. Вайганг, Д. Ю.

Колесник). Окремі елементи моделі сховища даних і компонентів системи було представлено на профільних науково-практичних заходах НУБіП України.

Структура роботи. Магістерська кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. У вступі обґрунтовано актуальність теми, сформульовано мету, завдання, об'єкт, предмет, окреслено методи дослідження, наукову новизну, практичне значення та наведено відомості про апробацію результатів. У першому розділі подано аналіз предметної області та сучасних підходів до пошуку інформації на складних веб-ресурсах. У другому розділі розроблено модель предметної області та логічну структуру бази даних і сховища даних. Третій розділ присвячено побудові архітектури інтелектуальної системи, алгоритмізації та програмній реалізації її основних модулів. У четвертому розділі наведено результати тестування прототипу системи, оцінювання її ефективності та сформульовано рекомендації щодо практичного впровадження. Загальний обсяг роботи становить 96 сторінок, містить 37 рисунків, 20 таблиці та 32 джерела.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА НАЯВНИХ ПІДХОДІВ ДО ПОШУКУ ІНФОРМАЦІЇ У СКЛАДНИХ ВЕБ-РЕСУРСАХ

1.1 Загальні принципи пошуку та аналізу інформації на складних веб-ресурсах

Складні веб-ресурси, такі як державні портали чи університетські системи, функціонують як багаторівневі цифрові екосистеми з великою різноманітністю форматів даних та нерівномірною якістю метаданих. Це ускладнює процес пошуку й суттєво відрізняє його від роботи зі звичайними сайтами. Ефективна система пошуку має одночасно працювати зі структурованими й неструктурованими даними, інтерпретувати зміст документів та залишатися стійкою до неоднозначності запитів користувачів, які дедалі частіше формулюються природною мовою [10].

Складність обробки таких запитів зумовлена їхньою короткістю, полісемією, використанням вузькоспеціалізованих термінів і різноманітністю форматів контенту, де потрібна інформація може бути прихованою у вкладених документах. Для підвищення точності сучасні системи використовують поєднання статистичних методів, алгоритмів машинного навчання та трансформерних моделей, що дозволяють коректно розпізнавати намір користувача та враховувати контекст попередніх дій [9].

Актуальною є гібридна модель інформаційного пошуку, яка поєднує лексичні методи (BM25) для швидкого первинного відбору документів із семантичним пошуком на основі векторних подань, що забезпечує знаходження змістово близької інформації незалежно від збігу ключових слів. Дослідження підтверджують, що саме комбінація sparse- та dense-підходів дає оптимальне співвідношення швидкодії й точності [7]. Використання сучасних трансформерних моделей (BERT, GPT) значно розширило можливості розуміння тексту, забезпечивши глибоку контекстуалізацію та здатність інтерпретувати складні або неоднозначні запити.

Використання великих мовних моделей та embedding-архітектур вимагає значних обчислювальних ресурсів і доменної адаптації, оскільки універсальні моделі часто не враховують локальну термінологію чи специфічні правила структурування даних [9]. Ефективність аналізу контенту залежить від коректного вилучення інформації з HTML, PDF, таблиць та JSON, нормалізації даних, виділення сутностей і кластеризації документів, а також побудови графів знань, які відображають семантичні зв'язки між матеріалами ресурсу [18]. У разі низької якості джерел, відсутності OCR або дефіциту метаданих точність пошуку різко зменшується, що підтверджено сучасними дослідженнями структури інформаційних потоків [21].

Встановлено, що якість обробки складних запитів значною мірою залежить від урахування історії взаємодії користувача, адже контекст попередніх дій підвищує точність визначення наміру та дозволяє системі уточнювати запит у режимі багатокрокового діалогу [10]. Персоналізація та інтерактивні моделі взаємодії, що базуються на LLM, забезпечують суттєве покращення результатів, особливо в середовищах із розгалуженою структурою контенту [9]. Водночас упровадження таких технологій стикається з проблемами продуктивності, складністю підтримки актуального індексу, помилками в даних та вимогами до захисту персональної інформації.

Сучасні IR-системи застосовують оптимізаційні механізми, такі як стиснення embeddings, швидкі алгоритми пошуку найближчих сусідів і кешування, що дає змогу зменшити обчислювальні витрати без втрати якості результатів [7]. У підсумку, ефективний пошук на великих веб-ресурсах потребує гібридного поєднання лексичних методів для швидкого відбору документів і семантичних моделей для змістового аналізу, а також контекстно чутливого визначення наміру користувача, що є ключовою передумовою релевантної видачі інформації.

Практична реалізація такої системи вимагає обов'язкової інтеграції контент-аналізу (включно з OCR, парсингом і виділенням сутностей), щоб повністю охопити різномірний вміст ресурсу. Крім того, необхідно адаптувати

мовні моделі до конкретного домену та постійно оптимізувати процеси індексування, враховуючи обмеження обчислювальних ресурсів для підтримки необхідної швидкості та продуктивності.

Запровадження інтелектуальних модулів пошуку на реальних веб-ресурсах пов'язане також з низкою нефункціональних обмежень. До таких належать обмежена пропускна здатність інфраструктури, необхідність підтримувати малий час відповіді за великої кількості одночасних звернень, вимоги до захисту персональних даних та інформаційної безпеки, а також потреба у прозорості та пояснюваності результатів пошуку [6; 9]. У зв'язку з цим у практиці реалізуються компромісні архітектурні рішення: поєднання швидкого лексичного пошуку з вибіркоким застосуванням семантичних моделей до підмножини кандидатів, використання асинхронної обробки, поетапного оновлення індексів та механізмів контролю якості даних.

Отже, загальні принципи побудови систем пошуку та аналізу інформації на складних веб-ресурсах базуються на поєднанні гібридного підходу до інформаційного пошуку, розвиненого контент-аналізу та врахування контексту користувачької взаємодії. Ефективність таких систем визначається не лише вибором конкретних алгоритмів, а й якістю підготовки даних, глибиною доменної адаптації мовних моделей, а також здатністю архітектури масштабуватися та забезпечувати прийнятний час відповіді.

1.2 Аналіз вимог до системи аналізу запитів користувачів

Проектування інтелектуальної системи аналізу запитів доступу до інформації на складних веб-ресурсах передбачає поетапне формування вимог, які охоплюють як функціональні можливості, так і нефункціональні характеристики. Якість узагальнення цих вимог безпосередньо визначає ефективність роботи системи, точність результатів пошуку, стійкість до зростання навантаження та зручність використання для кінцевих користувачів [3; 7]. Основне призначення цільової системи полягає у забезпеченні можливості швидкого й змістовно коректного доступу до інформації на веб-ресурсах із

розгалуженою структурою, великою кількістю сторінок і документів різних форматів, де традиційний пошук за ключовими словами часто виявляється недостатнім [4].

Сучасний користувач очікує від пошукової підсистеми не лише механічного співставлення термінів, а й здатності інтерпретувати зміст запиту, виявляти його намір та повертати узгоджену з цим наміром відповідь. Це особливо актуально для державних, освітніх та корпоративних порталів, де інформація розподілена між численними регламентами, положеннями, навчально-методичними матеріалами та службовими документами [2; 6]. За таких умов система аналізу запитів виконує роль проміжної ланки між природномовними зверненнями користувачів та внутрішньою структурою веб-ресурсу, фактично формуючи інтелектуальний навігаційний шар, який поєднує модулі інтерпретації запитів, семантичного пошуку й контент-аналізу [5].

Вимоги до функціональності системи визначають спектр операцій, які вона має підтримувати на рівні користувацького інтерфейсу та внутрішніх компонентів. По-перше, система повинна коректно обробляти природномовні запити українською та, за потреби, англійською мовами, інтерпретуючи контекст звернення та намір користувача. По-друге, підсистема пошуку має працювати не лише на основі лексичного збігу, а й забезпечувати семантичне зіставлення запитів із документами за допомогою векторних подань текстів і нейронних мовних моделей типу BERT, RoBERTa або спеціалізованих embedding-моделей [4; 8]. По-третє, система повинна виконувати контент-аналіз ресурсів, включно з автоматичним вилученням тексту з HTML-сторінок, PDF та DOCX-документів, виділенням заголовків, ключових тем та іменованих сутностей, а також формуванням міждокументних зв'язків для підтримки контекстної навігації [6; 9].

Окремою групою функціональних вимог є можливість формування інтелектуальних відповідей, коли користувач отримує не лише перелік посилань, а й короткі узагальнені фрагменти, що пояснюють сутність знайденої інформації з посиланням на відповідні джерела. Для адміністраторів веб-ресурсу система

має виконувати роль аналітичного інструмента: накопичувати статистику запитів, виявляти найбільш запитувані теми, визначати «вузькі місця» у структурі контенту та параметри навантаження. Важливо, щоб передбачалася можливість поетапного донавчання моделей на основі історії звернень і зворотного зв'язку від користувачів, що дає змогу поступово підвищувати точність та релевантність видачі [10; 11]. Узагальнений перелік ключових функціональних вимог наведено в таблиці 1.1.

Таблиця 1.1

Функціональні вимоги системи аналізу запитів

№	Функціональна вимога	Короткий опис
1	Обробка природномовних запитів	Система повинна розуміти запити українською та англійською мовами, інтерпретувати контекст і намір користувача.
2	Семантичний пошук	Забезпечення пошуку не лише за словами, а за змістом за допомогою нейромережових моделей і embeddings.
3	Контент-аналіз	Автоматичне вилучення тексту, заголовків, сутностей і ключових тем із веб-сторінок, PDF, DOCX тощо.
4	Формування інтелектуальних відповідей	Надання коротких текстових відповідей або резюме з посиланням на джерела.
5	Аналітика запитів	Збір і візуалізація статистики запитів, визначення популярних тем і проблемних зон веб-ресурсу.
6	Навчання системи	Можливість донавчання моделей на основі історії запитів і зворотного зв'язку користувачів.
7	Інтеграція з API	Підтримка зовнішнього доступу до пошуку й аналітики через REST або GraphQL API.

Поряд із функціональними характеристиками важливе значення мають нефункціональні вимоги, які визначають якість роботи системи, її продуктивність, масштабованість, безпечність та зручність використання. Для системи аналізу запитів на складних веб-ресурсах критичним параметром є час відповіді: від користувацької сесії очікується отримання результатів у межах кількох секунд навіть за умов значної кількості одночасних звернень [7].

Архітектура має підтримувати горизонтальне масштабування, що дозволить нарощувати ресурси обробки та зберігання даних у міру зростання аудиторії та обсягів контенту.

Особлива увага приділяється вимогам до інформаційної безпеки: дані запитів можуть містити персональну або конфіденційну інформацію, тому необхідним є використання захищених протоколів передавання, механізмів автентифікації та авторизації, а також регламентів доступу до аналітичних модулів. Користувацький інтерфейс має бути інтуїтивно зрозумілим, адаптивним до різних типів пристроїв та підтримувати допоміжні механізми, такі як автодоповнення запитів, підсвічування ключових фрагментів у результатах і засоби уточнення пошуку [9; 12]. Важливим елементом є також пояснюваність результатів (Explainable AI), коли користувач має можливість зрозуміти, чому саме певні документи були ранжовані як найбільш релевантні. Цей аспект істотно впливає на довіру до інтелектуальних систем, особливо в контексті публічних інформаційних сервісів [12; 14].

Узагальнена характеристика нефункціональних вимог до системи аналізу запитів подана в таблиці 1.2.

Таблиця 1.2

Нефункціональні вимоги системи аналізу запитів

№	Нефункціональна вимога	Короткий опис
1	Продуктивність	Обробка запитів до 2-5 секунд; оптимізація кешування та індексації для швидкого доступу.
2	Масштабованість	Підтримка горизонтального масштабування серверів для великих обсягів даних.
3	Безпека	Захист персональних даних, шифрування запитів і двофакторна автентифікація.
4	Зручність	Простий, інтуїтивний інтерфейс із підтримкою адаптивного дизайну для різних пристроїв.
5	Доступність	Безперервна робота 24/7, підтримка вебінтерфейсу та API для інтеграцій.
6	Пояснюваність	Забезпечення можливості користувачу розуміти принципи формування результатів.

Результати досліджень поведінки користувачів у складних інформаційних системах підтверджують, що очікування аудиторії виходять за межі простого отримання коректної відповіді. Користувачі прагнуть розуміти логіку формування результатів, отримувати пояснення та мати можливість скоригувати запит на основі наданих підказок [12]. У роботах Zhu та співавт. показано, що пояснюваність і прозорість пошукового процесу істотно підвищують довіру до системи в умовах складних інформаційних середовищ [12], тоді як дослідження Xu та Al-Shamari демонструють доцільність використання моделей обробки природної мови для покращення якості взаємодії людини з пошуковими сервісами й зменшення когнітивного навантаження на користувача [14; 15]. Використання великих мовних моделей у цьому контексті дає змогу поєднати алгоритмічну точність із глибшим розумінням змісту запиту та його контексту [4].

Таким чином, вимоги до системи аналізу запитів доступу до інформації на складних веб-ресурсах формуються на перетині технологічних та поведінкових аспектів. Вони враховують не лише суто технічні параметри продуктивності та безпеки, а й очікування користувачів щодо якості, пояснюваності та зручності взаємодії. Сформульований комплекс функціональних і нефункціональних вимог створює методичну основу для подальшого моделювання архітектури цільової системи та визначення інформаційних потоків

1.3 Структура інформаційних джерел для інтелектуального аналізу

Сучасні веб-ресурси державного, освітнього та наукового призначення формують багатокомпонентне інформаційне середовище, у якому поєднуються дані різних форматів і рівнів структурованості. Для проектування системи інтелектуального аналізу запитів необхідно встановити чітку модель інформаційних джерел, оскільки саме їхня структура визначає складність методів оброблення та ефективність подальшої інтеграції у модулі пошуку, індексації та контент-аналізу. Дослідження свідчать, що якість організації

джерел інформації має суттєвий вплив на релевантність видачі та швидкість роботи алгоритмів семантичного пошуку [17; 21].

Узагальнено інформаційні джерела можна розглядати за параметром рівня структурованості. Структуровані ресурси містять наперед визначені схеми (schema-first), які забезпечують чіткість форматів та однозначність пошуку, що є характерним для класичних реляційних баз даних або журналів подій. Напівструктуровані джерела поєднують формальні елементи (теги, атрибути, API-опис) зі змінними фрагментами вільного тексту. Неструктуровані дані не мають сталої схеми, що ускладнює операції вилучення змісту та потребує NLP-оброблення, векторизації та формування зв'язків між сутностями [19; 21].

Загальна схема класифікації джерел подана на рис. 1.1, який демонструє відмінності між групами даних та відповідні рівні їхньої формалізації.

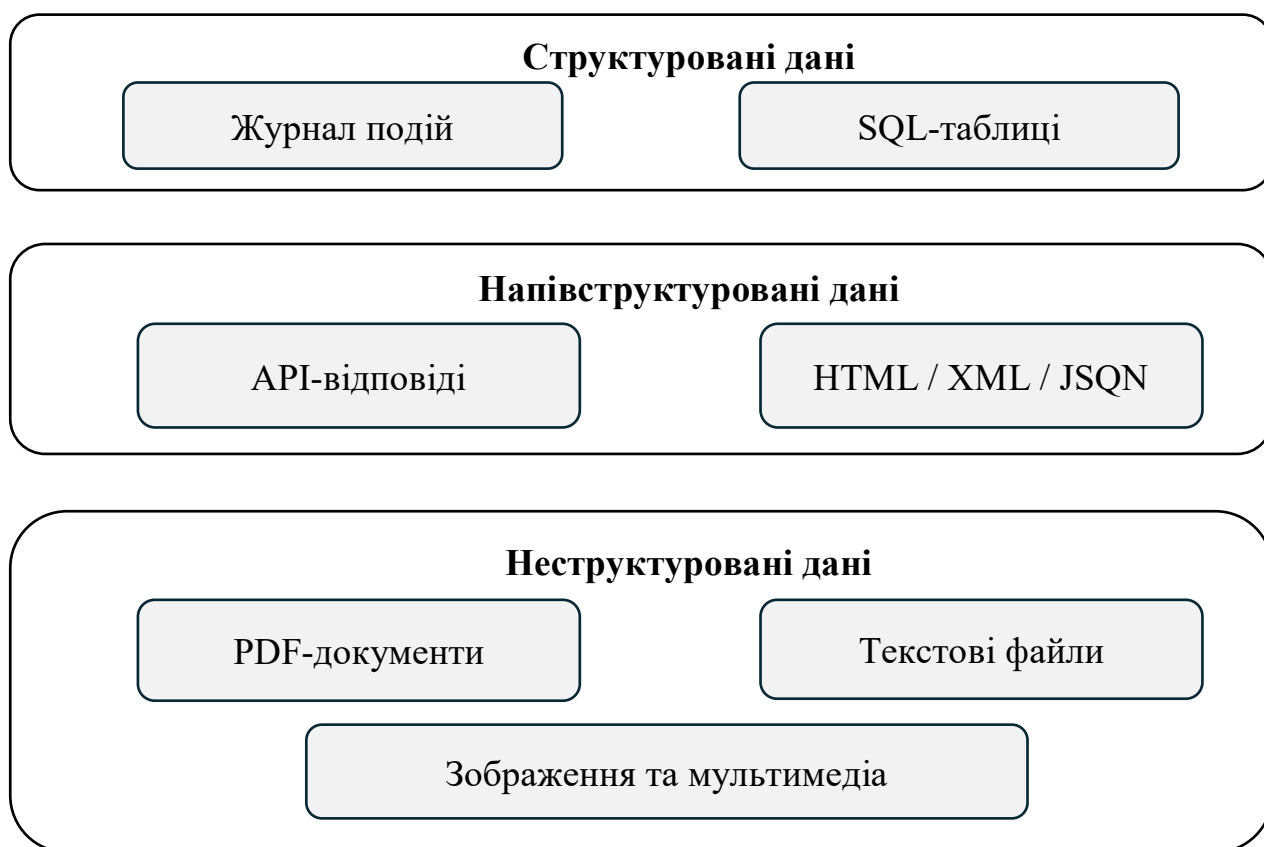


Рис. 1.1 Узагальнена схема класифікації інформаційних джерел

Ця класифікація прямо впливає на систему аналізу, оскільки методи обробки даних різні: структуровані джерела легко індексуються, а

неструктуровані й напівструктуровані вимагають глибокого аналізу тексту (NLP), витягання об'єктів, перетворення на вектори та створення карт знань.

У контексті великих веб-ресурсів важливо враховувати не лише формат, а й **ієрархічну організацію джерел**, яка включає розгалужену структуру тематичних розділів, вкладені меню, різнорівневі статичні й динамічні сторінки, внутрішні та зовнішні гіперпосилання. Для формального відтворення таких зв'язків застосовують моделі семантичних графів знань (knowledge graphs), що дозволяють фіксувати взаємини між сутностями й контекст їхнього використання [17]. На рис. 1.2 подано спрощений фрагмент графа знань, який відображає зв'язки між документами, категоріями та користувацькими запитами.

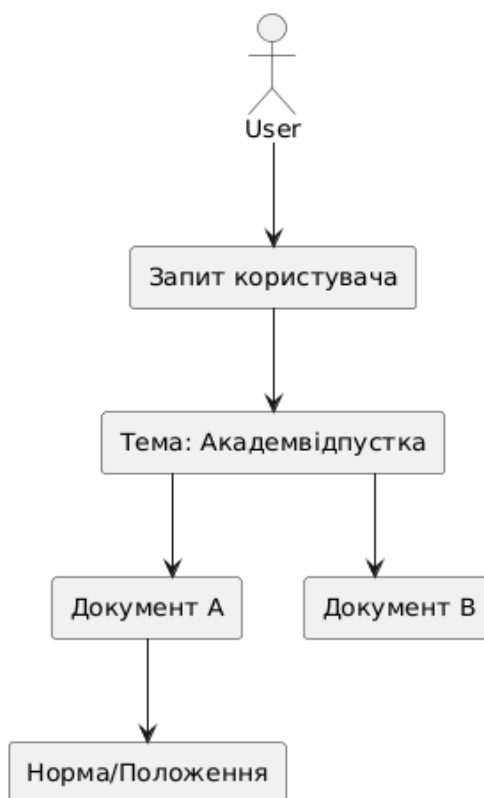


Рис. 1.2 Фрагмент семантичного графа знань веб-ресурсу

Для побудови високоточних моделей пошуку та аналізу запитів необхідна інтеграція джерел у централізоване середовище зберігання. Найбільш ефективними підходами вважають використання архітектур типу Data Warehouse та Data Lakehouse, які поєднують переваги зберігання як структурованих, так і неструктурованих даних з можливістю інкрементального

оновлення, формування часових зрізів та підтримки аналітичних операцій [19]. У такому середовищі зберігаються журнали взаємодії користувачів, метадані веб-ресурсів, тексти документів, результати попереднього NLP-аналізу, тематичні моделі та векторні уявлення (embeddings) [18].

У таблиці 1.3 наведено узагальнену характеристику трьох основних категорій інформаційних джерел та їхню роль у функціонуванні системи аналізу запитів.

Таблиця 1.3

Категорії інформаційних джерел для системи аналізу запитів

Категорія джерела	Рівень структурованості	Основні характеристики	Приклади даних
Структуровані	Високий	Таблиці, чіткі схеми, SQL-запити	База даних користувачів, журнал логів
Напівструктуровані	Середній	HTML/JSON/XML з метаданими, API-виклики	Веб-сторінка з тегами, API відповіді
Неструктуровані	Низький	Тексти, PDF, документи, мультимедіа без схеми	Стаття, звіт, блог-пост

Ефективність інтеграції таких джерел залежить від коректності виконання операцій ETL/ELT-оброблення, що забезпечує вилучення (Extract), трансформацію (Transform) і завантаження (Load) даних до сховища. Обов'язковими є нормалізація, стандартизація форматів, створення метамоделей та системи узгоджених словників. Окреме значення має оцінювання якості джерел: актуальність, повнота метаданих, їхня узгодженість та можливість побудови міждокументних зв'язків. Згідно з результатами сучасних досліджень, використання графів знань значно підвищує здатність системи відтворювати контекст і суттєво покращує релевантність результатів пошуку [18; 20].

У підсумку, структура інформаційних джерел визначає організаційний фундамент системи аналізу запитів. Від якості класифікації, способів інтеграції та повноти метаданих залежить можливість моделювання семантичних зв'язків,

які впливають на точність пошуку та здатність системи адаптуватися до змін інформаційного середовища. Докладний аналіз інформаційних джерел є передумовою побудови архітектури системи та модулів інтелектуального опрацювання даних.

1.4 Огляд сучасних інструментів і систем підтримки користувачів

Розвиток складних веб-ресурсів, зокрема державних порталів, університетських сайтів та корпоративних інформаційних систем, супроводжується зростанням вимог до якості пошуку та підтримки користувачів. Класичні засоби пошуку, орієнтовані лише на ключові слова, вже не забезпечують очікуваного рівня сервісу, оскільки користувачі формулюють запити природною мовою, часто не знаючи точних назв документів або розташування потрібної інформації в структурі сайту [2; 7]. У відповідь на ці виклики сформувався спектр інструментів, що поєднують методи інформаційного пошуку, обробки природної мови та елементи інтелектуальної підтримки, забезпечуючи інтерактивну взаємодію з користувачем і більш гнучке використання наявних інформаційних ресурсів [9].

Загальні підходи до побудови систем підтримки користувачів на складних веб-ресурсах доцільно класифікувати за двома осями: тип пошукового ядра (лексичний, семантичний або гібридний) та ступінь інтеграції з мовними моделями, здатними формувати узагальнені відповіді. На цій основі можна виокремити щонайменше три великі групи рішень: корпоративні пошукові платформи, інструменти семантичного пошуку з векторними базами даних та інтерактивні системи на основі великих мовних моделей і фреймворків інтеграції знань. Узагальнену схему розподілу сучасних інструментів за вказаними вимірами подано на рисунку 1.3.

У межах цієї класифікації корпоративні пошукові платформи, такі як Elastic Enterprise Search, забезпечують високопродуктивну індексацію та ранжування документів, працюючи переважно з текстовими та напівструктурованими даними. Вони орієнтовані на точний лексичний пошук,

доповнений можливістю врахування полів метаданих, вагових коефіцієнтів і складних фільтраційних умов [23].

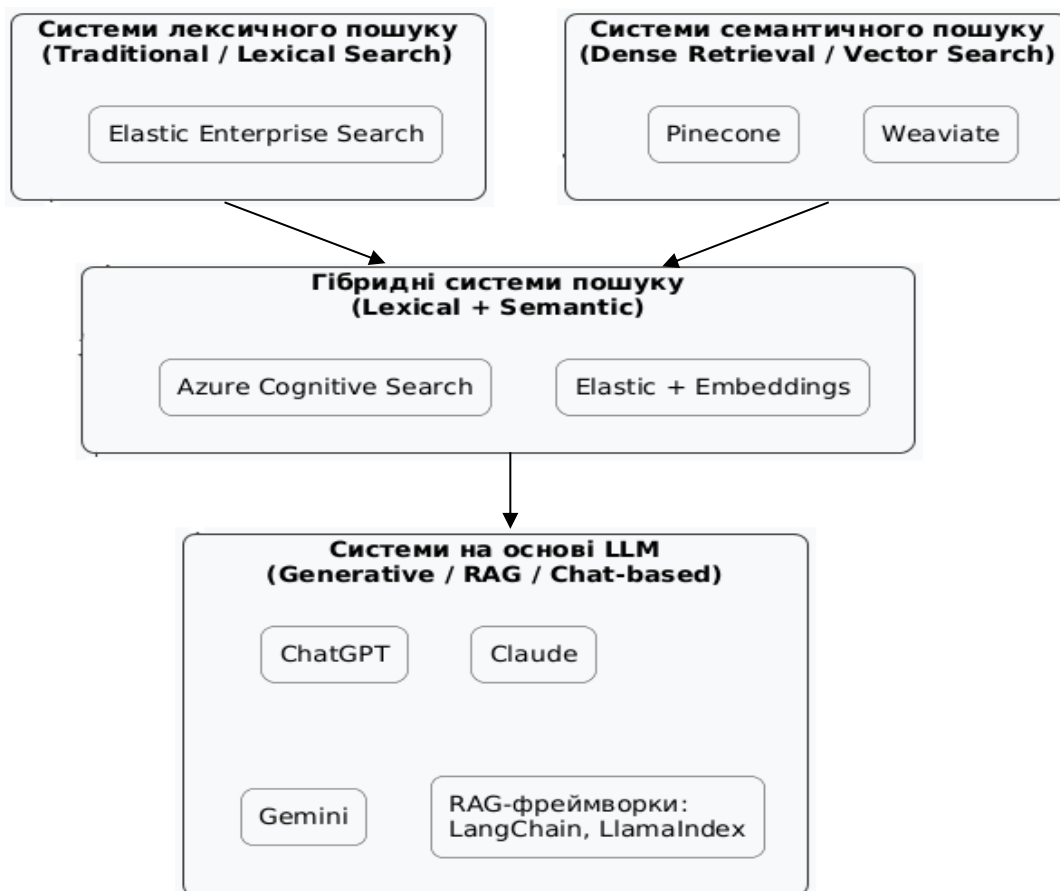


Рис. 1.3 Узагальнена класифікація сучасних інструментів підтримки користувачів за типом пошуку та рівнем інтеграції з мовними моделями

Корпоративні пошукові системи використовуються для індексації внутрішніх порталів і баз знань, забезпечуючи масштабованість і стійкість, але без NLP-модулів залишаються обмеженими у розумінні природномовних запитів та формуванні узагальнених відповідей. Семантичний пошук спирається на векторні бази даних, такі як Weaviate чи Pinecone, що працюють із векторними поданнями тексту та дають змогу знаходити змістово близькі документи навіть без лексичного збігу [18; 27]. Вони формують семантичний рівень аналізу, однак потребують додаткових компонентів для генерації пояснень та діалогової взаємодії.

Інструменти на основі великих мовних моделей (ChatGPT, Gemini, Claude, Copilot) демонструють високу здатність інтерпретувати наміри користувача та

підтримувати контекст багатокрокового діалогу [9; 28]. У поєднанні з механізмом Retrieval-Augmented Generation (RAG) такі системи можуть генерувати точні відповіді на основі знайдених документів [25], що реалізовано, зокрема, у Perplexity AI [23]. Проте більшість цих сервісів комерційні та недостатньо адаптовані до українських державних і освітніх веб-ресурсів.

Окремий напрям формують фреймворки LangChain та LlamaIndex, що поєднують доступ до локальних джерел, семантичний пошук і генеративні моделі, забезпечуючи розробку кастомізованих RAG-рішень [25]. Вони методично привабливі для систем аналізу запитів, але потребують високої кваліфікації розробників і значних ресурсів для керування векторними індексами. На корпоративному рівні прикладом є ServiceNow із модулем Now Assist, який використовує генеративні моделі для автоматизації обробки запитів і узагальнення статей бази знань [26], але орієнтований на закриті середовища. Векторні бази на кшталт Weaviate залишаються ключовим компонентом семантичного пошуку, хоча потребують додаткових генеративних модулів для створення відповідей [27].

Таблиця 1.4

Функціональні можливості сучасних інструментів і систем підтримки користувачів

Система	Perplexity AI	Elastic Enterprise Search	LangChain + LlamaIndex	ServiceNow (Now Assist)	Weaviate	ChatGPT
1	2	3	4	5	6	7
Пошук по контенту	+	+	+	+	+	-
Генерація коротких відповідей	+	-	+	+	-	+
Робота з локальними даними	-	+	+	+	+	-
Надання прямих посилань	+	+	+	+	+	+
Аналіз документів	-	+	+	-	+	-

Продовження табл. 1.4

1	2	3	4	5	6	7
Інтеграція на веб-ресурс	-	+	+	-	+	+
Можливість кастомізації	-	+	+	-	+	+
Розуміння контексту	+	-	+	+	-	+
Підтримка української мови	-	-	+	-	+	+

Примітка: Підтримка української мови залежить від конкретної конфігурації, версії моделі та мовних налаштувань середовища [9; 23–28].

Узагальнене місце розроблюваної системи аналізу запитів у просторі сучасних рішень можна схематично показати на рисунку 1.4. На ньому відображено, як цільова система поєднує властивості корпоративного пошуку (робота з локальним контентом і журналами запитів), семантичних індексів (векторні подання текстів) та інтерактивних мовних моделей (формування узагальнених відповідей і діалогова взаємодія).

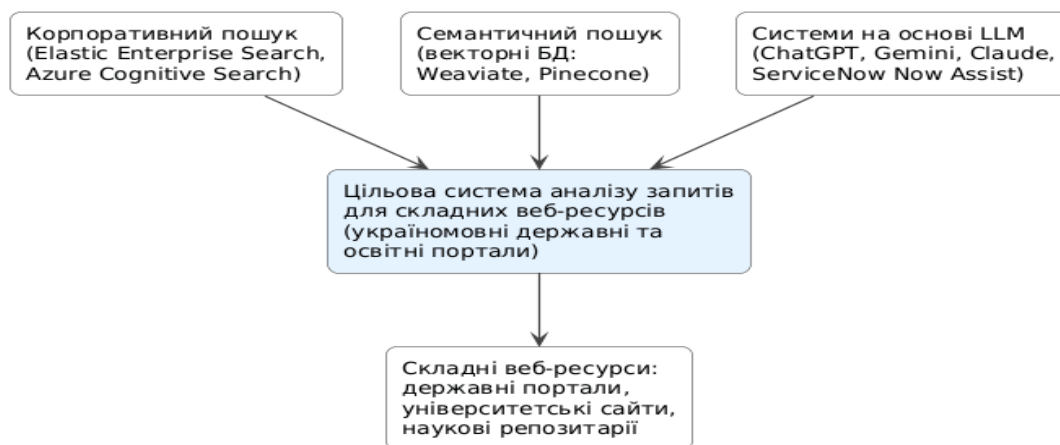


Рис. 1.4 Концептуальне положення цільової системи аналізу запитів серед сучасних інструментів підтримки користувачів

Таким чином, огляд сучасних інструментів і систем підтримки користувачів свідчить, що наявні рішення частково покривають потреби складних веб-ресурсів, але рідко забезпечують повну інтеграцію лексичного та семантичного пошуку, глибокого контент-аналізу й адаптації до специфіки україномовного контенту. Це обґрунтовує доцільність створення спеціалізованої

системи аналізу запитів, орієнтованої на роботу з даними державних та освітніх веб-ресурсів, з урахуванням вимог, сформульованих у попередніх підрозділах.

1.5. Проблеми та обмеження класичних методів пошуку інформації

Ефективність доступу до інформації на складних веб-ресурсах значною мірою залежить від здатності пошукових систем інтерпретувати запити користувача та коректно співвідносити їх зі змістом документів. У традиційних системах інформаційного пошуку, побудованих на лексичних моделях (TF-IDF, BM25), релевантність визначається співпадінням слів у запиті та документі, а контекст і семантика залишаються поза увагою алгоритму [7; 10]. Такий підхід був достатнім на ранніх етапах розвитку вебу, однак сучасні інформаційні системи характеризуються великим обсягом, різноманітністю даних, складною структурою та високою динамікою оновлень, що призводить до низки обмежень, які ускладнюють застосування класичних моделей у практиці – особливо в умовах державних, освітніх і корпоративних порталів [2; 6].

Однією з ключових проблем є залежність результату пошуку від точного формулювання запиту. Моделі TF-IDF і BM25 оперують статистикою слів: якщо термін не з'являється у документі, система не здатна оцінити змістовну близькість текстів, навіть якщо вони мають однаковий сенс, поданий іншими словами [9]. Користувачі часто формулюють запити природною мовою, у вигляді запитань, синонімічних фраз або неповних конструкцій, що ускладнює пошук і призводить до низької релевантності видачі. На рисунку 1.5 подано узагальнену схему роботи лексичної моделі пошуку, яка ілюструє, як система порівнює запит із документами виключно на рівні збігу термінів.

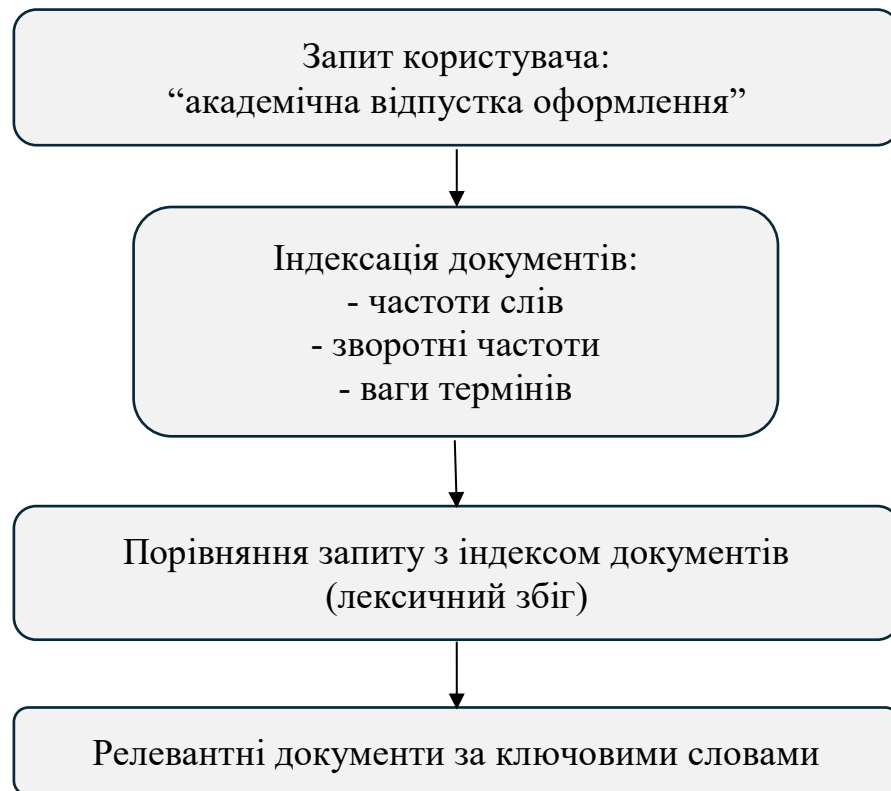


Рис. 1.5 Принцип роботи класичного лексичного пошуку (TF-IDF/BM25)

Обмеження лексичних моделей виявляються також у контекстах, де запит містить полісемію, багатозначність, або де зміст документа визначається структурними елементами (таблицями, вкладеними файлами, сканованими PDF), які класичний пошук не може інтерпретувати без додаткових модулів попередньої обробки. В освітніх і адміністративних системах часто зустрічаються нормативні документи, у яких одні й ті самі поняття формулюються різними термінами. У таких випадках система повертає нерелевантні результати через відсутність у запиті «правильного» слова, що не відповідає очікуваній поведінці користувача [11].

Окремий виклик становлять короткі запити, характерні для мобільних пристроїв або швидкої навігації. Вони не містять достатньої кількості сигналів для статистичної моделі. Лексичний пошук не здатний інтерпретувати намір короткого звернення, що призводить до шуму у видачі або, навпаки, до пропуску важливих документів. Дослідження Zhu та співавт. підтверджують, що короткі

запити є однією з найбільш проблемних категорій для класичних методів пошуку, особливо в умовах великих інформаційних порталів [12].

Важливою проблемою є також нездатність класичних моделей працювати з семантичними структурами, які виникають на рівні міждокументних зв'язків. Багато державних чи університетських ресурсів мають складну ієрархію сторінок, посилань та вкладених документів. Для інтерпретації таких даних необхідні моделі, здатні опрацьовувати контекст, структуру документа і відношення між його частинами [17]. Лексичні алгоритми позбавлені такого механізму, що обмежує їх використання у системах з високим ступенем структурної складності.

На рисунку 1.6 показано спрощену модель проблеми «лексичного розриву» (lexical gap), коли запит і документ містять різні слова для позначення одного й того самого поняття.

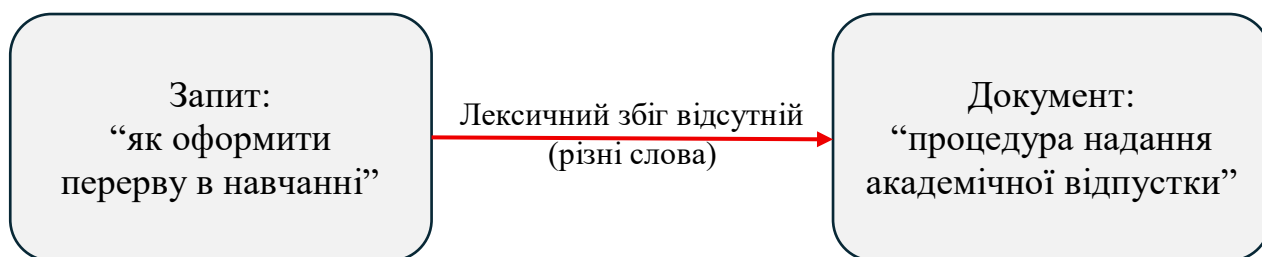


Рис. 1.6 Схематичне відображення “лексичного розриву” між запитом і документом

Ще одне обмеження стосується обробки неструктурованих даних. Університетські та державні веб-ресурси містять значний відсоток інформації у вигляді PDF-файлів, сканованих документів, презентацій, нормативних актів та звітів. Для лексичного пошуку такі документи або недоступні (у разі сканів), або мають низьку якість розпізнавання, що знижує повноту індексації. Класичні методи не володіють механізмами семантичного відновлення змісту та потребують додаткових інструментів, таких як OCR, токенизація та фільтрування шуму [19].

Окремо слід відзначити обмежену адаптивність традиційних систем. Вони не здатні навчатися на попередніх взаємодіях користувачів, не враховують

історію сесії, не визначають намір та не змінюють стратегію ранжування залежно від поведінкових патернів. У сучасних інформаційних середовищах, де структура контенту та профіль користувачів постійно змінюються, відсутність адаптивності суттєво знижує якість пошуку.

Таблиця 1.5 узагальнює основні проблеми класичних методів і показує їхній вплив на роботу систем інформаційного пошуку у контексті складних веб-ресурсів.

Таблиця 1.5

Основні обмеження класичних лексичних методів пошуку

Проблема	Характеристика	Наслідки для користувача
Залежність від точного формулювання	Пошук здійснюється лише за збігом слів	Низька релевантність, пропуск важливих документів
Відсутність семантичного розуміння	Не враховується змістовна подібність текстів	Неможливість пошуку «за сенсом»
Полісемія та синонімія	Слова можуть мати різні значення або варіанти	Хибні результати, надлишковий шум
Складність роботи з короткими запитам	Недостатність термінів для статистики	Нестабільність ранжування, нерелевантні відповіді
Неструктуровані дані	PDF, скани, таблиці, презентації	Часткове або неповне індексування
Відсутність адаптації	Моделі не навчаються на поведінці користувачів	Нездатність враховувати контекст сесії

Таким чином, класичні лексичні підходи, попри їхню ефективність у задачах базового інформаційного пошуку, демонструють суттєві обмеження при роботі зі складними веб-ресурсами, де структура контенту, мовні особливості та різноманітність джерел вимагають високого рівня семантичного аналізу та адаптивності. Ці обмеження обґрунтовують потребу у застосуванні сучасних підходів, заснованих на мовних моделях, векторному пошуку та інтелектуальних механізмах інтерпретації запитів, що і є предметом подальшого дослідження.

1.6 Постановка завдання для розробки системи підтримки користувачів на основі мовних моделей

Розроблення інтелектуальної системи підтримки користувачів передбачає створення інструменту, здатного оптимізувати доступ до інформації на складних веб-ресурсах, де звичайні механізми пошуку демонструють суттєві обмеження. Основою технологічної концепції є використання механізмів обробки природної мови, семантичного аналізу контенту та мовних моделей, що забезпечують інтерпретацію запитів, розуміння контексту й формування узагальнених відповідей. Такий підхід дає змогу подолати лексичні, структурні та когнітивні бар'єри, притаманні традиційним інструментам пошуку.

Поставлене завдання полягає у створенні системи, що працюватиме як інтерактивний асистент, інтегрований у веб-ресурс. Система повинна інтерпретувати запит природною мовою, зіставляти його зі вмістом сторінок, знаходити релевантні фрагменти інформації та формувати відповідь у придатній формі – від короткого тексту до прямого посилання або повного документа.

Особливістю системи є орієнтація на роботу з великими, складно структурованими веб-ресурсами державного, освітнього та інформаційного профілю. Для забезпечення універсальності необхідно створити платформу, здатну адаптуватися до різних типів контенту, включно зі статичними сторінками, вкладеними документами, архівами, структурованими та неструктурованими джерелами. Система має бути масштабованою, безпечною та придатною для інтеграції у вже існуючі портали.

Формалізація вимог охоплює і функціональні, і нефункціональні аспекти, однак замість переліку їх доцільно структуровано подати через узагальнену характеристику в табл. 1.6, що узгоджується з архітектурними принципами розроблюваної системи.

Таблиця 1.6

Узагальнення вимог до системи підтримки користувачів

Категорія	Характеристика	Очікуваний результат
Інтерпретація запиту	Аналіз природної мови, розуміння наміру	Пошук «за змістом», а не лише за словами
Семантичний пошук	Використання векторних моделей контенту	Виявлення релевантних документів, навіть без збігу термінів
Адаптивність	Реакція на історію звернень і контекст	Персоналізовані відповіді
Мультимовність	Підтримка української та англійської	Універсальність застосування
Надійність	Стійкість до високого навантаження	Швидкість відповіді до 3 секунд

Оскільки класичні методи пошуку не забезпечують достатньої точності на порталах з великою кількістю документів, постає завдання формального обґрунтування ключових бар'єрів, які має подолати запропонована система. Таблиці 1.7–1.9 описують проблеми, а нижченаведений текст пов'язує їх із вимогами до системи та визначає напрями їх вирішення через використання мовних моделей.

У першому випадку (табл. 1.7) основним викликом є складність навігації на великому порталі, де пошук класичним способом стає малоефективним. Для подолання цієї проблеми система має забезпечувати аналіз змісту сторінок та їхніх фрагментів з використанням семантичних моделей.

Таблиця 1.7

Обґрунтування проблеми 1

Проблема	Довгий і складний пошук інформації на великих сайтах
Зачіпає	Користувачів державних, освітніх і наукових порталів
Наслідком є	Витрати часу на навігацію, втрата релевантних сторінок, зниження ефективності
Успішне вирішення	Система підтримки користувачів забезпечує швидкий пошук за змістом сторінок, а не лише за ключовими словами

Друга проблема (табл. 1.8) стосується недостатньої актуальності чи релевантності результатів пошуку. Використання мовних моделей у поєднанні з фільтрацією за датою та джерелом забезпечує більш точні відповіді та усуває потребу в ручній перевірці великої кількості документів.

Таблиця 1.8

Обґрунтування проблеми 2

Проблема	Нерелевантні або застарілі результати пошуку
Зачіпає	Користувачів, які шукають офіційну або перевірену інформацію
Наслідком є	Помилки в прийнятті рішень, звернення до неактуальних даних
Успішне вирішення	Система використовує семантичний пошук і фільтрує за датою, джерелом і контекстом, забезпечуючи достовірність видачі

Третя проблема (табл. 1.9) зосереджена на обмеженості класичних інструментів у роботі із запитам природною мовою. Мовні моделі, навпаки, здатні розуміти намір, аналізувати семантичне ядро висловлювання та зіставляти його зі знаннями з бази, що значно підвищує точність відповіді.

Таблиця 1.9

Обґрунтування проблеми 3

Проблема	Відсутність адаптації пошуку до запитів природною мовою
Зачіпає	Непідготовлених користувачів, які не знають точних термінів або назв документів
Наслідком є	Неможливість знайти потрібну інформацію через неточне формулювання
Успішне вирішення	Інтеграція мовної моделі дозволяє системі “розуміти” зміст запиту, навіть якщо він сформульований звичайними словами

Після розгляду проблем і вимог постає необхідність визначити позиціонування продукту відносно існуючих рішень. У табл. 1.10 наведено характеристику продукту для різних типів користувачів, що дозволяє оцінити його потенційну цінність у контексті реальних сценаріїв використання.

Таблиця 1.10

Позиціонування

Для	Користувачів складних веб-ресурсів (державних, освітніх, інформаційних)
Необхідно	Швидко знаходити потрібну інформацію, не орієнтуючись у структурі сайту
Назва продукту	Система підтримки користувачів для пошуку інформації на складних веб-ресурсах
Деталі	Веб-модуль з інтеграцією LLM і NLP, що дозволяє знаходити сторінки, документи або давати короткі відповіді
На відміну від	Звичайних пошукових систем, що базуються лише на ключових словах
Наш продукт	Розуміє зміст запиту, враховує контекст і надає релевантну відповідь у зручній формі

Логічно продовжуючи аналіз, доцільно доповнити його характеристикою основних груп користувачів, які взаємодіють із системою на різних рівнях доступу. Це подано у табл. 1.11, що формує загальну картину цільової аудиторії.

Таблиця 1.11

Користувачі системи

Типовий представник	Опис	Тип	Відповідальності	Критерії успіху
Користувач	Особа, яка шукає інформацію на державному чи освітньому сайті	Кінцевий користувач	Формулює запити, отримує відповіді	Отримання релевантної інформації
Аналітик/адміністратор	Відповідальний за управління базою знань і налаштування моделей	Системний користувач	Оновлює дані, контролює релевантність	Підвищення точності системи
Незареєстрований відвідувач	Тимчасовий користувач, який взаємодіє з базовою версією пошуку	Гість	Переглядає відкриту інформацію	Зручний пошук без авторизації

Таким чином, постановка завдання передбачає створення гнучкої, адаптивної та семантично орієнтованої системи, здатної інтегрувати технології векторного пошуку та мовних моделей у структуру складних веб-ресурсів. Розроблювана система орієнтована не лише на точний пошук, але й на інтелектуальну підтримку користувача, що дозволяє істотно підвищити якість інформаційних сервісів, забезпечити швидкий доступ до знань і сформувавши основу для подальшого розвитку цифрових платформ.

Висновки до розділу 1

У ході аналізу встановлено, що ефективний пошук інформації на складних веб-ресурсах ускладнюється високою контентною неоднорідністю, багаторівневою структурою та низькою якістю метаданих, що зумовлює обмеженість класичних лексичних моделей. Виявлено, що користувацькі запити здебільшого формулюються природною мовою, містять контекстні залежності та неоднозначність, а тому потребують застосування сучасних NLP-методів, семантичного пошуку й адаптивних механізмів визначення наміру. Аналіз структури даних показав необхідність інтеграції структурованих, напівструктурованих і неструктурованих джерел та використання моделей контент-аналізу, здатних обробляти різноманітні формати, включно з PDF, HTML та API-відповідями.

Огляд сучасних інструментів підтвердив доцільність поєднання нейромережових моделей, векторних представлень та гібридних retrieval-підходів для досягнення високої релевантності результатів. Визначено, що існуючі системи підтримки користувачів або не адаптовані до україномовного контексту, або не забезпечують глибокого аналізу контенту, що обґрунтовує потребу у створенні власної інтелектуальної системи. Отримані результати стали підґрунтям для формалізації вимог до розроблюваної платформи, визначення її функціонального призначення та постановки завдання щодо моделювання архітектури.

2. МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АНАЛІЗУ ЗАПИТІВ КОРИСТУВАЧІВ

2.1 Моделювання предметної області

Моделювання предметної області інтелектуальної системи аналізу запитів користувачів є ключовим етапом проєктування, оскільки забезпечує формальне відображення сутностей, процесів та відносин, що визначають роботу цільової системи. На цьому етапі відбувається узгодження уявлень про функціональну структуру майбутнього застосунку, уточнюються межі системи, а також формується концептуальна модель взаємодії між користувачами, програмними компонентами та інформаційними ресурсами. Така формалізація забезпечує можливість подальшого коректного проєктування схеми даних, алгоритмів обробки запитів та архітектури модулів, що інтегрують методи NLP, семантичного пошуку та аналітичної обробки даних.

Предметна область системи характеризується наявністю трьох груп об'єктів: користувачів, які формують запити природною мовою; програмних компонентів, що здійснюють їх аналіз і перетворення; та аналітичних підсистем, які забезпечують накопичення знань про взаємодію з веб-ресурсом. Основою взаємодії є інформаційний запит, що виконує роль центральної сутності та пов'язує діалог між користувачем та інтелектуальною системою. Його оброблення передбачає декілька послідовних процедур: приймання, лінгвістичний аналіз, визначення наміру, пошук релевантного контенту, формування відповіді та накопичення даних для подальшого вдосконалення системи.

Для формалізації взаємодій використовуються діаграми UML, оскільки вони дозволяють уніфіковано представити структуру процесів і забезпечити узгоджене бачення між аналітиком, розробником і замовником. Особливу роль на етапі моделювання відіграє діаграма прецедентів, яка відображає функціональні можливості системи через взаємодію між зовнішніми акторами та

внутрішніми процесами. На рис. 2.1 подано діаграму прецедентів, що демонструє основні сценарії роботи: навігацію користувача веб-ресурсом, формування запиту через чат-бот, семантичний аналіз змісту, пошук інформації, генерацію відповіді та аналітичну обробку даних для подальшої оптимізації якості обслуговування.

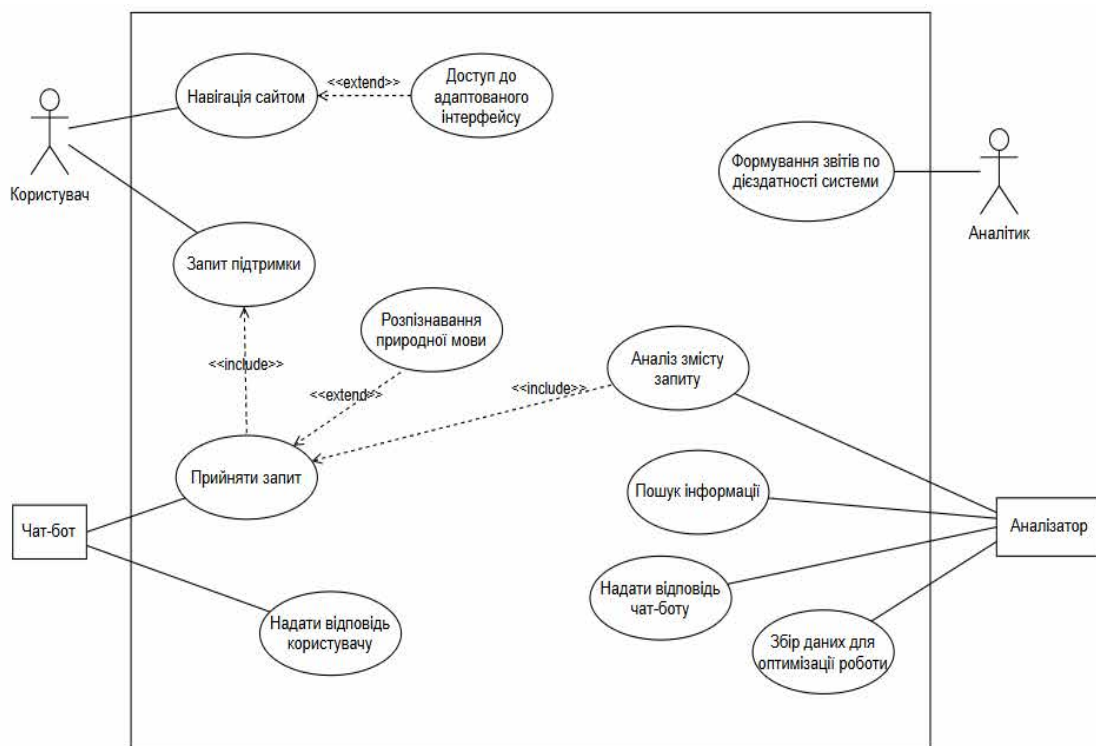


Рис.2.1 Діаграма прецедентів інтелектуальної системи аналізу запитів користувачів

На діаграмі відображено взаємозв'язок чотирьох акторів: користувача, чат-бота, аналітичного модуля та аналітика. Користувач формує природномовний запит і отримує відповідь, тоді як чат-бот виконує роль інтерфейсу, який забезпечує прийняття запиту, передачу його до модуля обробки та повернення відповіді. Аналітичний модуль реалізує основні когнітивні функції системи: здійснює розпізнавання природної мови, визначає намір користувача, виконує пошук інформації та формує оптимізовану відповідь. Аналітик взаємодіє із зібраними статистичними даними, що генеруються під час роботи системи, та формує звіти щодо дієздатності, точності та стабільності оброблення запитів, що

є основою для подальшої оптимізації алгоритмів та оновлення онтологічної моделі контенту.

Для узагальнення ролей учасників взаємодії та забезпечення цілісного розуміння функціональної структури системи доцільно виокремити ключові характеристики кожного актора. Така деталізація дозволяє чітко визначити межі відповідальності між компонентами, а також встановити логіку потоків інформації під час оброблення користувацьких запитів. Узагальнена характеристика основних учасників подана у табл. 2.1, що відображає їх призначення в системі та основні функції, які вони виконують у процесі аналізу та підтримки доступу до інформації.

Таблиця 2.1

Основні актори інтелектуальної системи аналізу запитів

Актор	Опис ролі в системі	Ключові функції та дії
Користувач	Головний учасник взаємодії, відвідувач веб-ресурсу, який потребує доступу до інформації.	Формування природномовного запиту; отримання відповіді від чат-бота; надання зворотного зв'язку щодо релевантності отриманої інформації.
Чат-бот	Інтерфейс взаємодії між користувачем та системою; канал передавання запиту і відповіді.	Приймання запиту; передавання його до модуля обробки; попередній аналіз; запуск пошуку; формування та повернення відповіді користувачу.
Система аналізу даних	Внутрішній аналітичний компонент, що забезпечує оброблення запитів і накопичення службової інформації.	Реєстрація запитів; обробка й фіксація результатів пошуку; аналіз якості опрацювання; збереження метрик для оптимізації роботи.
Аналітик	Фахівець, відповідальний за оцінювання ефективності системи та прийняття рішень щодо її вдосконалення.	Аналіз категорій запитів і тенденцій; оцінювання точності пошуку; підготовка аналітичних звітів; формування рекомендацій для покращення моделі та інтерфейсу.

Описана модель предметної області дозволяє створити цілісне уявлення про логіку функціонування системи та забезпечує перехід до формування

структур даних, алгоритмів семантичної обробки й архітектури програмних компонентів. Її використання забезпечує узгодженість подальших проектних рішень та сприяє виявленню потенційних обмежень на ранніх етапах розробки. Такий підхід відповідає вимогам до створення інтелектуальних систем підтримки користувачів, де критично важливими є структурованість, формалізованість та здатність моделі точно відтворювати закономірності взаємодії між користувачем і системою.

2.2 Моделювання динаміки процесів у системі

Динамічне моделювання є ключовим етапом проєктування інтелектуальної системи аналізу запитів, оскільки воно дозволяє формалізувати поведінку програмних компонентів, описати їхню взаємодію та визначити логіку переходів між окремими етапами оброблення інформації. У межах цього підрозділу представлено два типи процесів: зовнішній, що відображає роботу чат-бота з користувачем, та внутрішній, спрямований на аналітичне опрацювання запитів, які не були успішно оброблені. Використання UML-діаграм діяльності дає змогу в наочній формі відобразити обидва сценарії, підкресливши послідовність операцій і розмежування відповідальності між учасниками процесу.

Перший процес описує повний цикл взаємодії користувача з чат-ботом. Сценарій починається з моменту введення тексту запиту, після чого повідомлення передається чат-боту для оброблення природної мови. Первинна лінгвістична інтерпретація забезпечує виокремлення ключових елементів змісту та перетворення запиту в структуру, придатну для семантичного аналізу. На наступному етапі аналітичний модуль визначає намір користувача й здійснює пошук необхідної інформації у базі даних або пов'язаних інформаційних джерелах. Розгалуження процесу відбувається в контрольному вузлі «Відповідь знайдена?». Якщо релевантний матеріал ідентифіковано, чат-бот формує та повертає відповідь, після чого сценарій завершується. Якщо ж система не знаходить відповідь, користувачеві пропонується уточнити запит, і сценарій

переходить у повторний цикл. Послідовність описаних дій та розмежування відповідальності між користувачем, чат-ботом і аналітичним модулем подано на рис. 2.2.

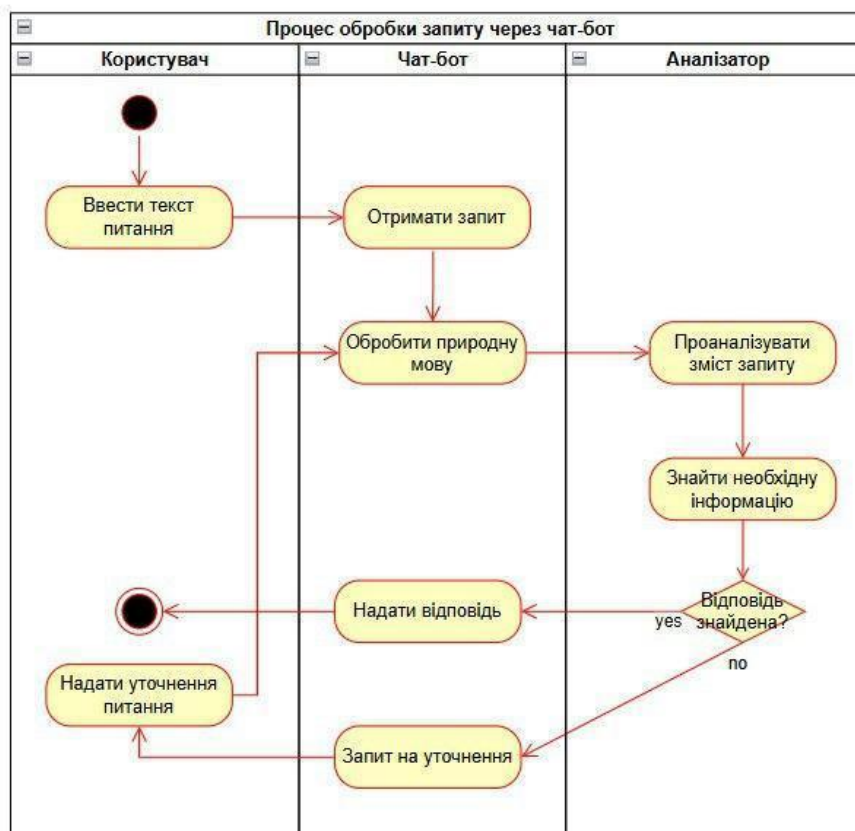


Рис. 2.2 Діаграма діяльності процесу оброблення запиту через чат-бот

Другий процес моделює внутрішню логіку системи у випадках, коли відповідь не була знайдена в основному циклі. На цьому рівні відбувається аналітичне опрацювання отриманого запиту з метою виявлення причин неуспішного пошуку. Після фіксації невдалого результату система ініціює процедуру перевірки якості введеного тексту, що включає оцінку повноти формулювання, логічної зв'язності та відповідності запиту очікуваному формату. Далі здійснюється виявлення типових помилок, серед яких можуть бути граматичні, семантичні або структурні відхилення. Важливим етапом є аналіз шаблонів запитів на основі історичних даних, що дозволяє виявити повторювані структури, класифікувати складні запити та визначити зони системних недоліків. Завершальним етапом є формування аналітичного звіту,

який узагальнює отримані результати та може бути використаний для удосконалення моделей NLP та підвищення ефективності роботи системи загалом. Структуру внутрішнього аналітичного процесу подано на рис. 2.3.

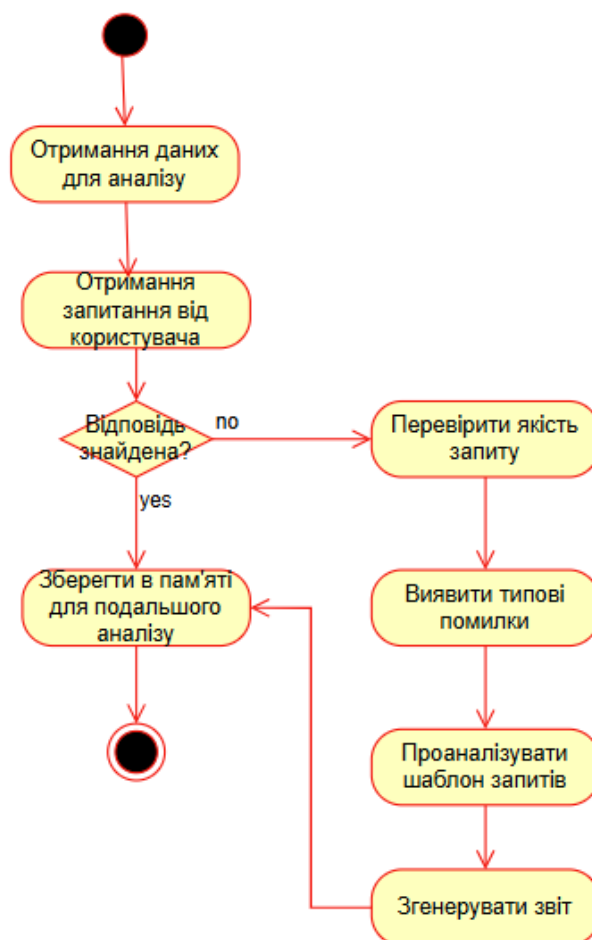


Рис. 2.3 Діаграма діяльності процесу внутрішньої аналітики та удосконалення системи

Для узагальнення двох взаємопов'язаних сценаріїв було сформовано табл. 2.2, яка структуровано відображає зміст і функціональне призначення основних процесів системи. Поділ на зовнішній і внутрішній рівні процесів забезпечує комплексне бачення роботи системи: від моменту звернення користувача до етапів аналізу та накопичення знань, необхідних для підвищення точності відповідей у майбутньому.

Основні процеси динамічної моделі системи

№	Процес	Зміст та функціональне призначення
1	Оброблення користувачького запиту	Забезпечує приймання, аналіз і семантичну інтерпретацію запиту, а також формування відповіді у разі успішного пошуку даних.
2	Уточнення запиту користувачем	Реалізує механізм повторного вводу даних у випадку, коли релевантна інформація не була знайдена, що дозволяє підвищити точність подальшого оброблення.
3	Перевірка якості запиту	Оцінює коректність і повноту тексту, визначає термінологічну й структурну відповідність запиту очікуваному формату.
4	Виявлення типових помилок	Визначає характерні проблеми, що призводять до помилок пошуку, включно з семантичними, граматичними й структурними відхиленнями.
5	Аналіз шаблонів запитів	Досліджує історичні дані з метою виявлення повторюваних структур, класифікації складних запитів і формування узагальнених закономірностей.
6	Формування аналітичного звіту	Узагальнює результати аналізу запитів і генерує рекомендації для оновлення моделей NLP та покращення роботи системи.

Узагальнений опис двох моделей демонструє взаємодоповнюваність зовнішнього та внутрішнього процесів: перший забезпечує безпосередню взаємодію з користувачем, другий – систематичне вдосконалення алгоритмів на основі накопичених даних. Сукупність цих процесів визначає адаптивний характер системи та формує основу для високої точності, стабільності й масштабованості її роботи.

2.3 Діаграми послідовності взаємодії компонентів системи

Діаграми послідовності слугують ключовим засобом моделювання поведінкової логіки інтелектуальної системи аналізу запитів, дозволяючи простежити часову організацію обмінів повідомленнями між програмними компонентами та оцінити узгодженість інформаційних потоків. На відміну від

статичних моделей, які описують структуру системи, послідовні діаграми відображають динаміку взаємодії, що є особливо важливим для багатокомпонентних архітектур, у яких поєднуються традиційні механізми веб-навігації та інтелектуальні модулі семантичного аналізу текстових даних. Для досліджуваної системи було побудовано три репрезентативні сценарії, кожен із яких демонструє різний рівень складності взаємодії між компонентами та охоплює різні типи поведінки користувача.

Перший сценарій відображає процес навігації веб-інтерфейсом, під час якого користувач ініціює перехід між сторінками за допомогою елементів веб-застосунку. Подія фіксується компонентом `UserInterface`, який формує навігаційний запит до модуля `NavigationSystem`. Останній виконує оновлення маршруту та повертає інтерфейсу опис цільової сторінки, після чого її відображення синхронно оновлюється у браузері користувача. Цей сценарій характеризується лінійною структурою, мінімальною кількістю операцій та відсутністю складних обчислювальних процесів, що забезпечує швидкість і передбачуваність реакції системи (рис. 2.4).

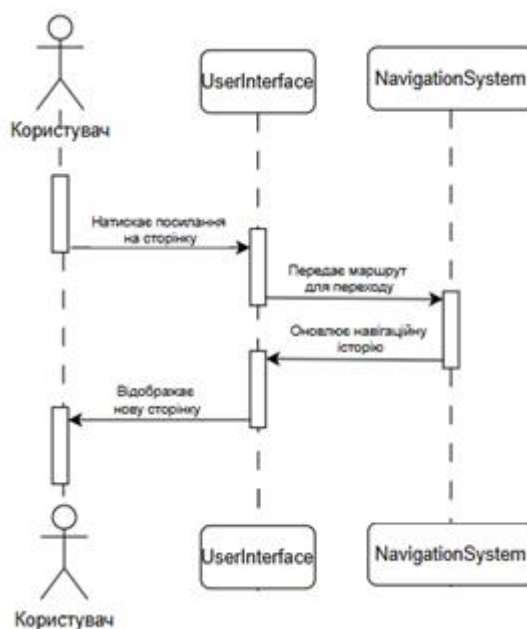


Рис. 2.4 Діаграма послідовності процесу навігації веб-інтерфейсом

Другий сценарій описує роботу механізмів доступності, що забезпечують адаптацію інтерфейсу до індивідуальних потреб користувачів. Коли користувач

активує відповідну функцію, `UI` надсилає команду модулю `Accessibility`, який змінює параметри відображення – масштаб шрифту, рівень контрастності, кольорову палітру чи інші елементи візуального оформлення. Оновлені параметри повертаються до інтерфейсу, який застосовує їх до поточної сторінки. Цей сценарій потребує меншої кількості операцій, проте вимагає суворої синхронності, адже від коректності і швидкості обробки залежить доступність інтерфейсу та комфорт користувача (рис. 2.5).

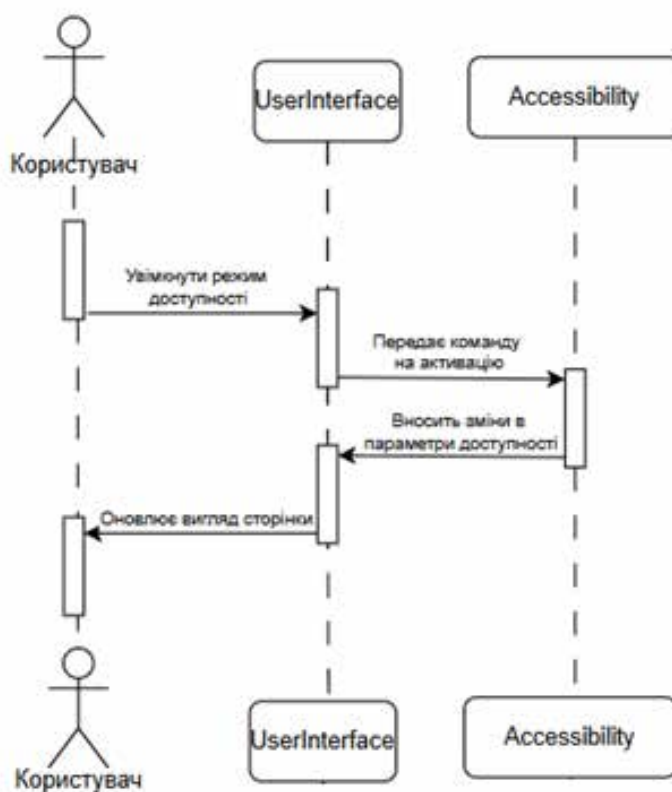


Рис. 2.5 Діаграма послідовності активації та застосування режиму доступності

Найбільш складним є третій сценарій – обробка текстового запиту через інтелектуальний чат-бот, де реалізується повний цикл лінгвістичного та семантичного аналізу. Користувач вводить запит, який фіксується `UI` та передається модулю `SupportBot`. Після первинної обробки чат-бот ініціює звернення до `ContentAnalyzer`, що виконує морфологічний і синтаксичний розбір тексту, визначає ключові елементи та встановлює ймовірний намір користувача. Для побудови семантичного подання `ContentAnalyzer` взаємодіє з підсистемою `LanguageModelIntegration`, де задіяна мовна модель виконує векторизацію тексту

та формує релевантний набір відповідей. Після узгодження результатів ContentAnalyzer повертає їх чат-боту, який формує готову відповідь і передає її інтерфейсу. Сценарій включає багаторівневі взаємодії, інтенсивні обчислення та звернення до зовнішніх інтелектуальних сервісів, що забезпечує високу точність семантичної інтерпретації (рис. 2.6).

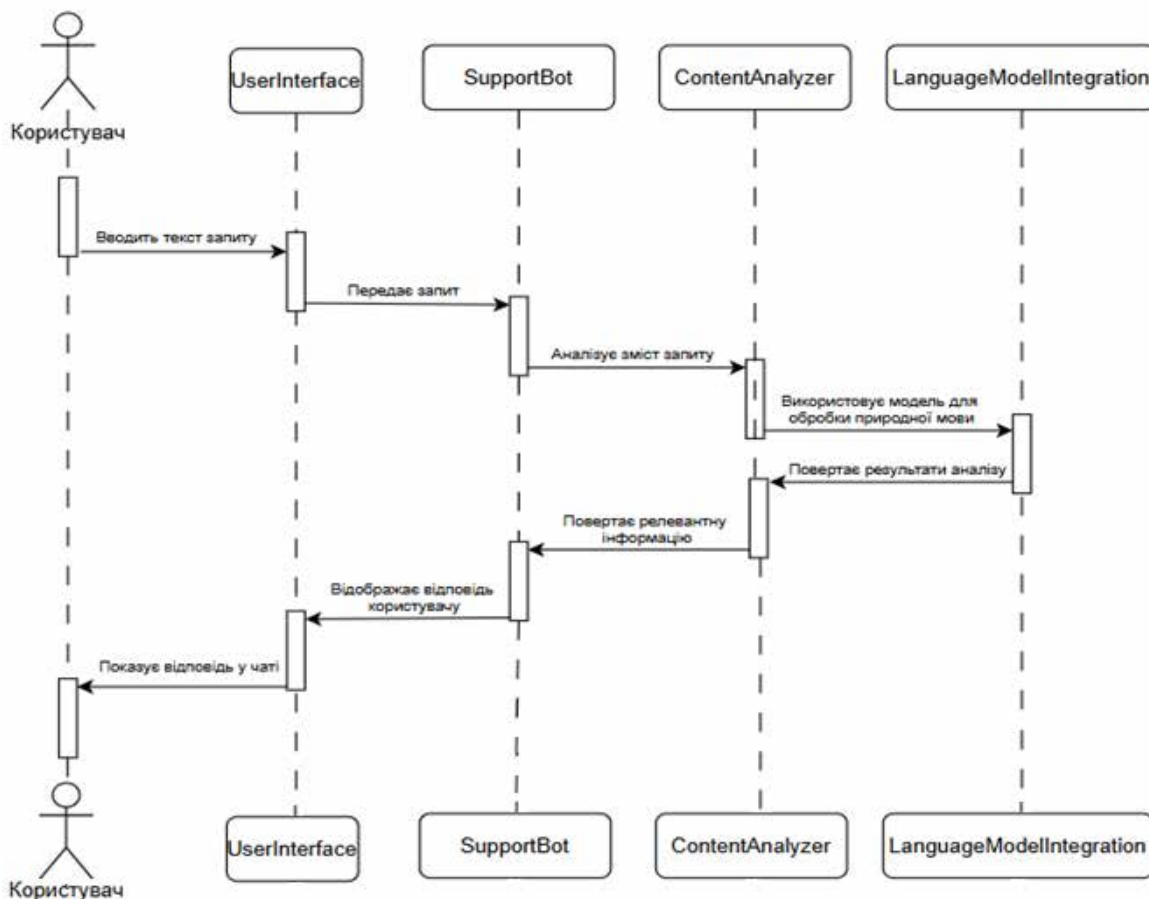


Рис..2.6 Діаграма послідовності обробки користувацького запиту в чат-боті з використанням семантичного аналізу

Для узагальнення описаних сценаріїв та порівняння їхніх характеристик доцільно представити ключові параметри кожної моделі взаємодії у табличній формі (табл. 2.3). Це дає змогу системно продемонструвати відмінності між сценаріями за змістом, складністю обробки, залученими компонентами та інтенсивністю використання аналітичних ресурсів, що є важливим підґрунтям для подальшого проектування архітектури системи та вибору оптимальних механізмів масштабування.

Таблиця 2.3

Характеристики базових сценаріїв взаємодії інтелектуальної системи аналізу запитів

Сценарій	Короткий зміст взаємодії	Основні залучені компоненти	Рівень залучення аналітиків та НЛР	Використання БД та сховища даних	Основне призначення
Навігація веб-сторінками	Перехід між сторінками, відображення контенту без зміни логіки обробки запитів	Користувач, модуль UserInterface, підсистема NavigationSystem	Низький	Звернення до БД переважно для читання статичного або кешованого контенту; сховище даних не використовується безпосередньо	Забезпечення базової взаємодії з веб-ресурсом
Обробка запиту через чат-бот	Введення текстового запиту, його семантичний аналіз, пошук відповіді та формування відповіді	Користувач, SupportBot, ContentAnalyzer, LanguageModelIntegration, БД, сховище даних	Високий	Інтенсивне використання БД для фіксації запитів і відповідей; передавання узагальненої інформації до сховища даних для подальшої аналітики	Реалізація основної бізнес-логіки системи
Керування режимами доступності	Зміна параметрів відображення інтерфейсу, адаптація системи до потреб користувача	Користувач, UserInterface, модуль Accessibility, БД	Низький – середній	Зберігання параметрів профілю користувача в БД; періодичне узагальнення даних у сховищі для аналізу використання режимів доступності	Підвищення зручності та інклюзивності інтерфейсу

На основі табличного узагальнення було побудовано графічну схему, яка інтегрує три розглянуті сценарії в єдину модель взаємодії користувача з інтелектуальною системою аналізу запитів (рис. 2.7).

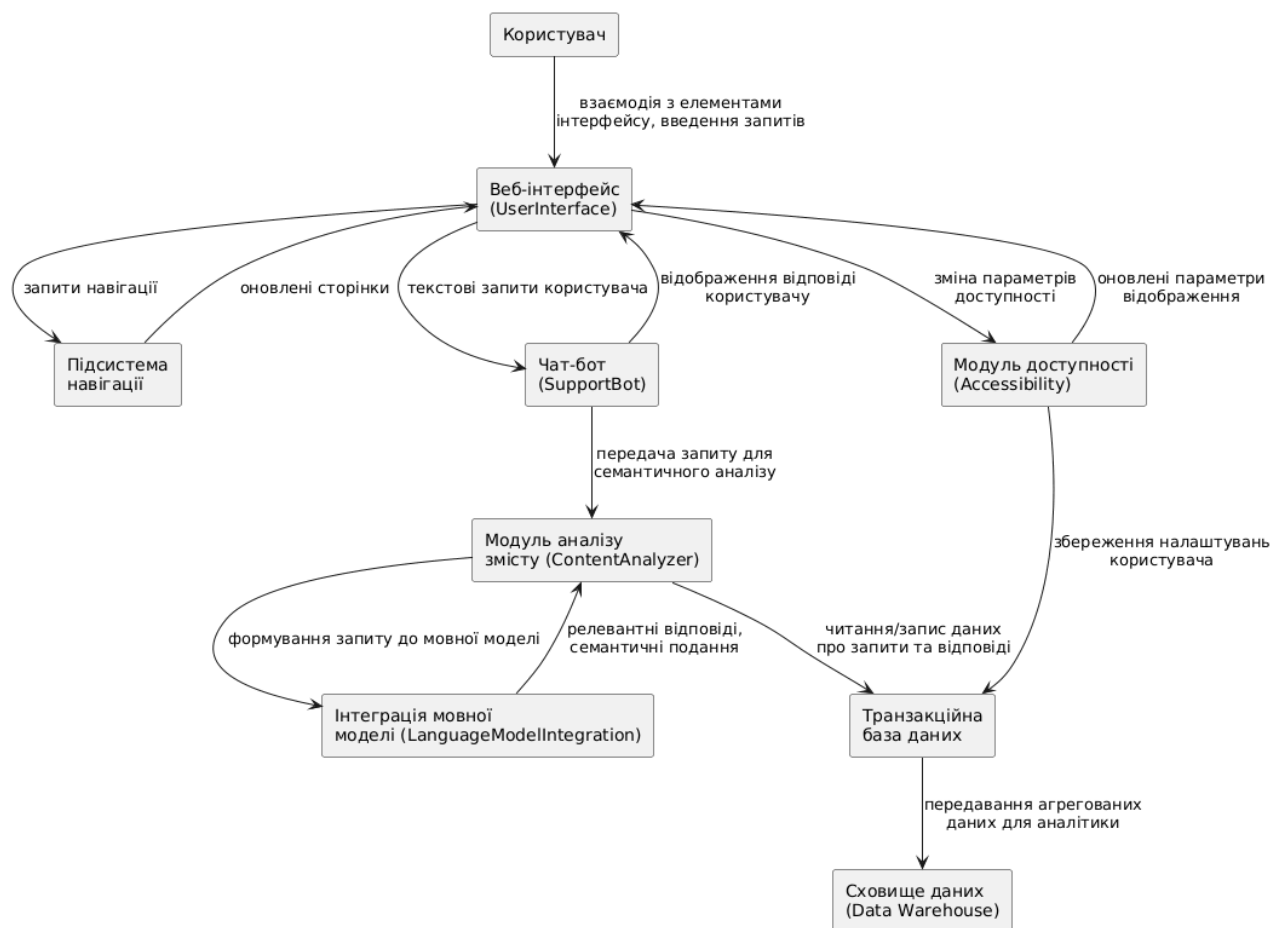


Рис. 2.7 Узагальнена схема взаємодії компонентів інтелектуальної системи аналізу запитів у базових сценаріях

Схема відображає центральну роль веб-інтерфейсу як посередника між користувачем і внутрішніми сервісами, показує, як модулі навігації, чат-бота, аналітики та керування доступністю комбінуються у межах цілісного архітектурного рішення.

Порівняння описаних сценаріїв демонструє поступове ускладнення обробки: від простих синхронних викликів у навігаційній моделі до глибоких багатокомпонентних взаємодій, які охоплюють роботу мовних моделей та елементів штучного інтелекту. Для систематизації основних характеристик сценаріїв у табл. 2.4 наведено узагальнення, що дозволяє оцінити їхню структуру, складність, обчислювальні витрати, взаємозалежності та роль у формуванні загальної моделі функціонування системи.

Таблиця 2.4

Порівняльні характеристики сценаріїв взаємодії компонентів системи

№	Характеристика	Сценарій навігації веб-інтерфейсом	Сценарій активації режиму доступності	Сценарій обробки текстового запиту чат-ботом
1	Тип взаємодії	Синхронна, лінійна	Синхронна з коротким циклом адаптації	Багаторівнева, комбінована (синхронна + асинхронні виклики моделей)
2	Учасники	Користувач, UserInterface, NavigationSystem	Користувач, UserInterface, Accessibility	Користувач, UserInterface, SupportBot, ContentAnalyzer, LanguageModelIntegration
3	Призначення сценарію	Перехід між сторінками та оновлення контенту	Адаптація інтерфейсу до особливих потреб користувача	Аналіз і семантична обробка користувацького текстового запиту
4	Складність процесу	Низька	Низька–середня	Висока
5	Основні операції	Передача маршруту, оновлення історії, відображення сторінки	Зміна параметрів доступності, оновлення відображення	Лінгвістичний аналіз, семантичне моделювання, формування відповіді
6	Використані модулі	UserInterface, NavigationSystem	UserInterface, Accessibility	SupportBot, ContentAnalyzer, LanguageModelIntegration
7	Наявність обчислювальних навантажень	Мінімальні	Низькі	Значні (аналіз тексту, ML-моделі)
8	Ризики затримок	Практично відсутні	Незначні (перемальовування інтерфейсу)	Можливі затримки через роботу мовної моделі
9	Очікуваний результат	Відкриття нової сторінки	Змінений вигляд інтерфейсу	Сформована релевантна відповідь
10	Рівень залежності від зовнішніх сервісів	Немає	Немає	Високий (мовна модель / інтелектуальні сервіси)

Узагальнення сценаріїв підтверджує узгодженість динамічної архітектури: усі підсистеми інтегруються за єдиним принципом обміну повідомленнями, а структурована послідовність операцій формує підґрунтя для стабільного функціонування інтелектуальної системи. Результати моделювання дозволили ідентифікувати критичні точки, можливі затримки та дублювання операцій, що

важливо для оптимізації алгоритмів обробки запитів. Саме ці висновки стали базою для формування логічної моделі сховища даних, яка забезпечує збереження аналітичної інформації та підтримує подальші етапи інтелектуальної обробки.

2.4. Концептуальне та логічне проєктування сховища даних інтелектуальної системи

Сховище даних інтелектуальної системи аналізу запитів виконує функцію аналітичного ядра, у якому консолідуються, агрегуються та структуруються дані, отримані з транзакційних компонентів системи. На відміну від оперативної бази даних, що орієнтована на підтримку поточних сервісів (чат-бот, навігація, керування профілями користувачів), сховище даних призначене для багатовимірного аналізу великих обсягів історичної інформації, побудови статистичних і прогнозних моделей, а також підтримки прийняття рішень щодо розвитку та оптимізації веб-ресурсу.

Формування моделі сховища даних здійснюється з урахуванням предметної області, що включає процеси пошуку, обробки та представлення інформації на складних веб-ресурсах, а також специфіку використання мовних моделей і методів Data Mining для аналізу запитів. Вихідною точкою є ідентифікація бізнес-процесів, які потребують аналітичної підтримки: динаміка звернень користувачів у часі, розподіл запитів за тематиками та типами контенту, показники ефективності обробки (час відповіді, частка успішно вирішених звернень), оцінки якості та релевантності відповідей, використання режимів доступності інтерфейсу тощо. Для кожного із зазначених процесів визначаються ключові показники (measures) і виміри (dimensions), що зумовлює вибір відповідної структури сховища.

Особливістю сховища даних є застосування сурогатних ідентифікаторів (ID), які не мають предметного змісту та не залежать від внутрішньої структури оперативної БД. Їхнє використання забезпечує стабільність аналітичної моделі, дозволяє реалізувати повільно змінні виміри (slowly changing dimensions) та

мінімізує ризики неконсистентності під час оновлення даних. Таким чином, ID у сховищі даних виконують роль аналітичних «якорів», тоді як у транзакційній БД слугують для забезпечення предметної цілісності й підтримки нормалізованих зв'язків.

Логічна модель сховища даних побудована за зірковою схемою (Star Schema), яка передбачає наявність центральної таблиці фактів та множини таблиць-вимірів.

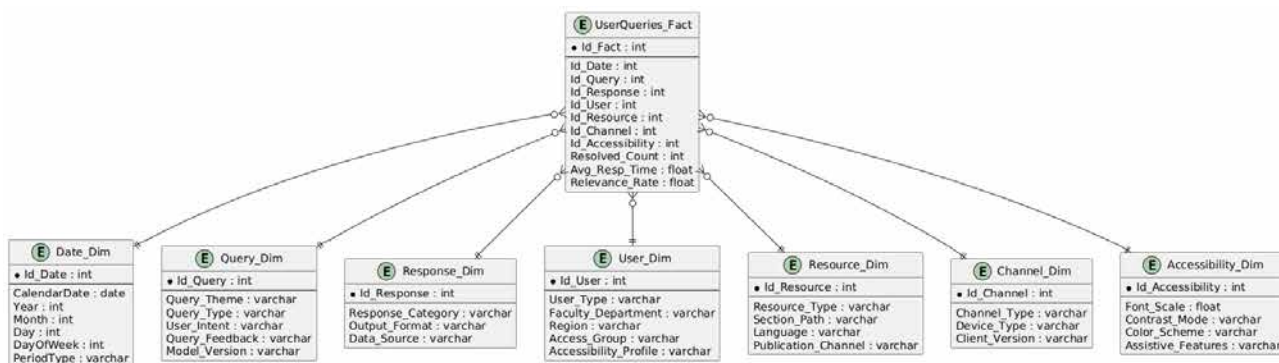


Рис. 2.8 Логічна модель даних інтелектуальної системи аналізу запитів

Такий підхід забезпечує простоту інтерпретації, високу продуктивність запитів та ефективну підтримку OLAP-операцій, що особливо важливо за умов значних обсягів історичних даних про взаємодію користувачів із системою. У центрі моделі розташовано фактову таблицю UserQueries_Fact, що накопичує агреговані показники роботи системи: кількість опрацьованих запитів за певні інтервали, середній час відповіді, частку запитів, для яких було надано релевантну відповідь, кількість діалогів, що потребували ескалації, тощо. Кожен запис у таблиці фактів пов'язаний із записами у вимірних таблицях через зовнішні ключі, що дозволяє аналізувати ці показники у розрізі часу, тематики, типу відповіді, каналу доступу, характеристик користувача та інших аспектів.

До складу вимірних таблиць входить, передусім, Date_Dim – вимір, що описує часовий аспект аналізу. Він містить сурогатний ідентифікатор Id_Date, а також атрибути календарної дати, року, місяця, дня, дня тижня, можливо – ознаки навчального або робочого періоду, що дає змогу виконувати групування та агрегацію показників на різних рівнях деталізації. Вимір Query_Dim описує

семантичні характеристики запиту: унікальний ідентифікатор `Id_Query`, тему або кластер тематики, тип запиту (інформаційний, навігаційний, транзакційний), виявлений намір користувача, а також показники зворотного зв'язку (оцінка корисності відповіді, наявність додаткових уточнень).

Окрему групу атрибутів становлять параметри, пов'язані з використанням мовних моделей, зокрема ідентифікатор версії моделі, застосовані стратегії семплінгу, інформація про конфігурацію семантичного пошуку.

Важливе місце у моделі займає вимір `Response_Dim`, який характеризує тип і структуру відповіді системи: `Id_Response` (сурогатний ключ), категорію відповіді (коротка довідка, розширене пояснення, набір посилань, фрагмент нормативного документа, шаблон листа тощо), параметри оформлення (текст, таблиця, комбінований формат), а також службові ознаки щодо джерела даних (внутрішній документ, зовнішній ресурс, результат аналітичного запиту до сховища). Для відображення користувацького аспекту передбачено вимір `User_Dim` із атрибутами `Id_User`, тип користувача (студент, викладач, співробітник, зовнішній відвідувач), факультет або структурний підрозділ, країна/регіон, групи доступу, використання режимів доступності. Цей вимір дозволяє досліджувати відмінності у поведінці різних категорій користувачів та персоналізувати параметри інтерфейсу.

З огляду на просторово-структурну складність веб-ресурсу доцільно виділити вимір `Resource_Dim`, що описує об'єкт, до якого фактично стосувався запит: `Id_Resource`, тип ресурсу (новина, методичний матеріал, нормативний документ, службова інструкція), ієрархію розділів сайту, мову контенту, канал публікації. Додатково використовується вимір `Channel_Dim` для класифікації каналів доступу (браузер, мобільний застосунок, зовнішній API), а також вимір `Accessibility_Dim`, що фіксує комбінації параметрів режимів доступності, активованих у момент надходження запиту. Усі зазначені виміри пов'язані з `UserQueries_Fact` через зовнішні ключі, які зазвичай реалізуються як цілочислові сурогатні ідентифікатори, незалежні від операційної схеми бази даних. Саме це відрізняє сховище даних від класичної транзакційної бази: у сховищі головну

роль відіграють стабільні, оптимізовані для аналітики ключі та денормалізовані структури, що підтримують багатовимірний аналіз, тоді як в оперативній БД ідентифікатори більше пов'язані з предметно-орієнтованими сутностями та нормалізованими зв'язками.

Для наочності основні таблиці сховища даних та їх призначення зведено в узагальнювальну таблицю (табл. 2.5), яка демонструє розподіл функцій між фактовою та вимірними таблицями.

Таблиця 2.5

Основні таблиці сховища даних інтелектуальної системи аналізу запитів

№	Таблиця	Роль у моделі	Приклади ключових атрибутів
1	UserQueries_Fact	Акумуляція агрегованих показників роботи системи	Id_Fact, Id_Date, Id_Query, Id_Response, Id_User, Id_Resource, Resolved_Count, Avg_Resp_Time, Relevance_Rate
2	Date_Dim	Часовий вимір аналізу	Id_Date, CalendarDate, Year, Month, Day, DayOfWeek, PeriodType
3	Query_Dim	Семантичні характеристики запитів	Id_Query, Query_Theme, Query_Type, User_Intent, Query_Feedback, Model_Version
4	Response_Dim	Типологія відповідей системи	Id_Response, Response_Category, Output_Format, Data_Source
5	User_Dim	Характеристики користувача	Id_User, User_Type, Faculty_Department, Region, Access_Group, Accessibility_Profile
6	Resource_Dim	Структура й тип цільового ресурсу	Id_Resource, Resource_Type, Section_Path, Language, Publication_Channel
7	Channel_Dim	Канал доступу до системи	Id_Channel, Channel_Type, Device_Type, Client_Version
8	Accessibility_Dim	Параметри режимів доступності	Id_Accessibility, Font_Scale, Contrast_Mode, Color_Scheme, Assistive_Features

Побудоване сховище даних не обмежується лише фізичним рівнем зберігання інформації, а передбачає також формування предметно-орієнтованих «вітрин даних» (data marts), які відображають різні проєкції загальної моделі на потреби окремих груп користувачів. Наприклад, вітрина «Ефективність обробки запитів» агрегує факти у розрізі дати, тематики й типу відповіді та використовується для контролю навантаження на систему й виявлення «вузьких місць» у роботі мовної моделі. Вітрина «Поведінка користувачів» акцентує увагу

на вимірах User_Dim, Channel_Dim і Accessibility_Dim, дозволяючи аналізувати, як різні категорії користувачів взаємодіють із системою та які режими доступності застосовують. Вітрина «Якість контенту та релевантність» будується переважно на поєднанні Query_Dim, Response_Dim та Resource_Dim, зосереджуючись на показниках зворотного зв'язку та релевантності відповідей. Такі вітрини можуть бути реалізовані як окремі OLAP-куби або матеріалізовані подання, що використовують спільне сховище даних як єдине джерело правди.

Узагальнюючи викладене, можна зазначити, що сформована модель сховища даних інтелектуальної системи аналізу запитів забезпечує чітке розмежування між оперативною та аналітичною обробкою інформації, підтримує побудову різноманітних вітрин для користувачів з різними аналітичними потребами та створює основу для подальшої інтеграції методів Data Mining і машинного навчання. Логічна структура сховища, організована за зірковою схемою з виділенням фактів і вимірів, дозволяє гнучко адаптувати систему до зростання обсягів даних і ускладнення запитів, зберігаючи при цьому прозорість моделі й можливість її інтерпретації як з технічної, так і з предметної точки зору.

Висновки до розділу 2

У другому розділі сформовано цілісну методологічну основу для побудови інтелектуальної системи аналізу запитів користувачів. На базі моделювання предметної області уточнено структурні межі системи, визначено ключові сутності й інформаційні взаємозв'язки, що забезпечують коректне подання користувацьких звернень, контенту та результатів обробки. Побудовані діаграми діяльності та послідовності відтворюють логіку оброблення запиту – від моменту його надходження до формування релевантної відповіді – і відображають роль окремих компонентів у забезпеченні семантичного аналізу та інтеграції даних.

Сформована логічна модель сховища даних і структура інформаційних потоків створюють підґрунтя для подальшої реалізації архітектури системи. Узгодження концептуальної моделі, процедурних механізмів обробки запитів і

структури DWH забезпечує можливість масштабованого, адаптивного та ефективного виконання аналітичних операцій. Отримані результати розділу є основою для переходу до архітектурного проєктування та програмної реалізації інтелектуальної системи, що розглядається у наступному розділі.

3. АРХІТЕКТУРА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АНАЛІЗУ ЗАПИТІВ

3.1 Архітектура системи інтелектуального аналізу

Архітектура системи інтелектуального аналізу запитів формує багаторівневу модульну структуру, яка забезпечує узгоджену роботу програмних компонентів та підтримує повний цикл оброблення користувацьких звернень – від моменту їх формування до отримання аналітичних висновків і рекомендацій. Такий підхід дає змогу досягти високої гнучкості, масштабованості, відмовостійкості та адаптивності системи до змін інформаційного середовища.

Архітектура охоплює послідовний ланцюг взаємопов'язаних рівнів, серед яких інтерфейсний, прикладний, лінгвістичний, транзакційний, аналітичний та рівень довгострокового зберігання даних. На рис. 3.1 наведено узагальнену структурно-функціональну схему роботи системи, що включає компоненти веб-взаємодії, модуль обробки запитів, сервер чат-бота, аналітичний сервер, операційну базу даних та модуль підготовки даних для аналітики.

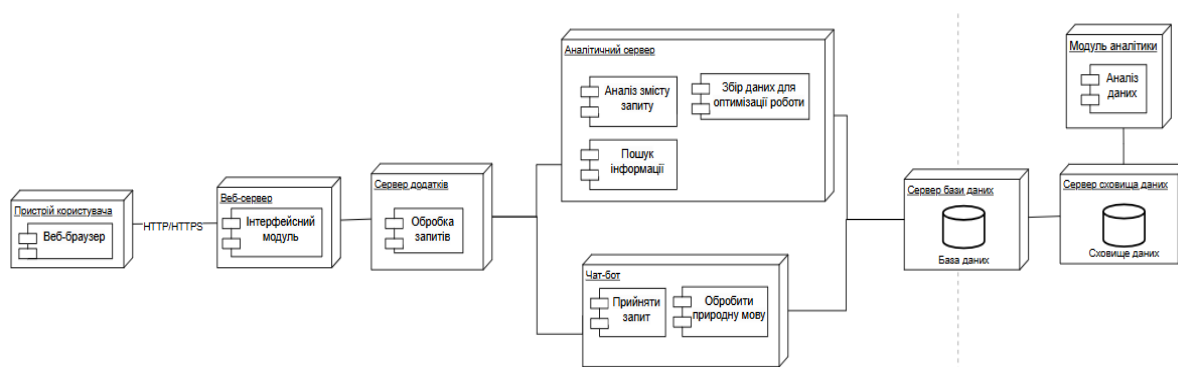


Рис. 3.1 Архітектура системи інтелектуального аналізу

на рис. 3.1, де представлено повний логічний цикл оброблення запиту – від моменту його введення у веб-браузері до формування відповіді аналітичним або мовним модулем. На першому етапі запит користувача надходить до

інтерфейсного модуля веб-сервера через стандартні HTTP/HTTPS-протоколи. Інтерфейсний модуль виконує первинну валідацію, створює службові метадані та передає запит на сервер прикладної логіки, що виступає центральним вузлом маршрутизації. Саме на цьому рівні визначається характер і тип звернення: чи потребує воно семантичної обробки, чи спрямовується на прямий пошук інформації в базі даних, чи передається до NLP-компонента чат-бота для інтерпретації природною мовою. Така структура дозволяє збалансувати навантаження та уникнути блокування системи в умовах високої інтенсивності запитів. Сервер чат-бота містить модуль приймання та модуль обробки природної мови, які виконують лексичний, семантичний і контекстуальний аналіз тексту запиту, формують його структуроване подання та адаптують зміст для пошукових і аналітичних механізмів.

Транзакційна база даних функціонує як оперативне сховище, у якому накопичуються запити, відповіді, журнали взаємодій і параметри системних подій. На відміну від неї аналітичний сервер виконує складні операції з обробки змісту, категоризації, пошуку документів і оцінювання релевантності на основі історичних даних. У межах аналітичного сервера також здійснюється збирання статистичних метрик, необхідних для оцінки ефективності роботи системи, зокрема середнього часу обробки, частоти повторних звернень, структури тематичних категорій і динаміки запитів у часі.

Консолідована аналітична інформація потрапляє до багаторівневого сховища даних, архітектуру якого подано на рис. 3.2. Сховище даних організовано як розгалужена структура, що включає операційні джерела, ETL-процеси, інтегроване ядро DWH і тематичні вітрини даних. Джерела формують подійні записи про роботу чат-бота, транзакції веб-ресурсу та зовнішні відкриті реєстри. На проміжному ETL-рівні відбуваються вилучення даних, очищення від шумів, нормалізація текстів, усунення дублювань і приведення форматів до єдиної структури, після чого дані переносяться до інтегрованої моделі сховища.

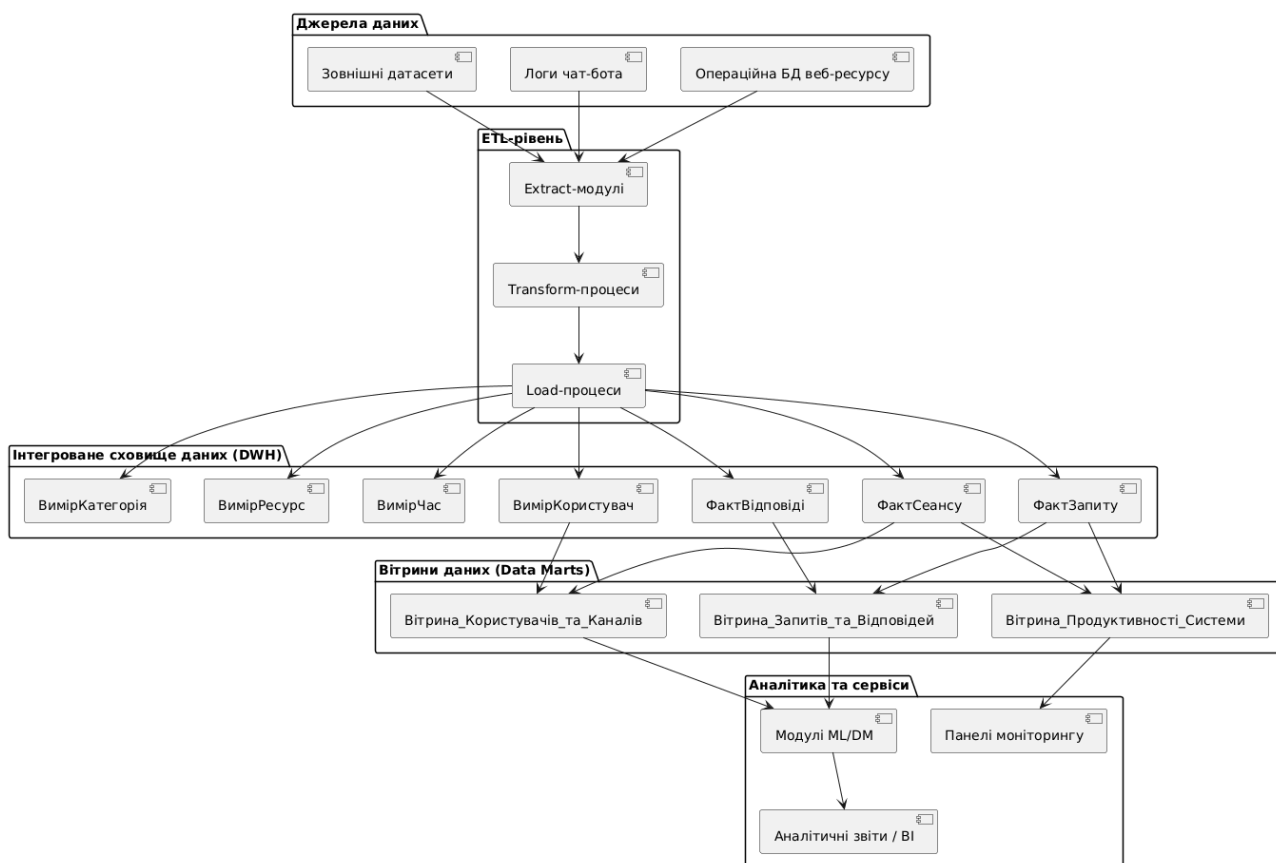


Рис. 3.2 Архітектура сховища даних інтелектуальної системи аналізу запитів

Ядро DWH побудоване за зірковою схемою та містить таблиці фактів, що відображають запити, відповіді, сеанси й показники продуктивності, а також множину вимірів – користувачів, часу, ресурсів, тематичних категорій і типів звернень. Для забезпечення стабільності структури застосовуються сурогатні ключі, які підтримують історичність даних і відображають еволюцію довідників у часі.

На верхньому рівні розташовано вітрини даних, кожна з яких орієнтована на конкретні аналітичні завдання: моделювання поведінки користувачів, оцінювання якості відповідей, формування рекомендацій, аналіз навантаження та продуктивності системи. Вітрини оптимізовано для інтеграції з алгоритмами машинного навчання, модулями Data Mining та інструментами побудови панелей моніторингу. Завдяки цьому система може не лише надавати відповіді, а й вдосконалювати власні алгоритми роботи, адаптуючись до змін у структурі запитів і поведінці користувачів.

Узгоджене функціонування всіх рівнів архітектури забезпечує формування повного інтелектуального циклу аналізу запитів, у межах якого поєднуються операційні процеси, семантична обробка текстів, аналітичне моделювання та історичний аналіз даних. Така архітектура гарантує надійність, масштабованість і відтворюваність процесів, дозволяє системі ефективно інтерпретувати запити користувачів, формувати релевантні відповіді та постійно вдосконалювати якість роботи за рахунок накопичених знань.

3.2 Розробка бази даних для зберігання інформації про запити користувачів

Розробка бази даних є одним з ключових етапів проєктування системи інтелектуального аналізу користувацьких запитів, оскільки саме вона забезпечує зберігання всіх даних, що виникають під час взаємодії між користувачем, чат-ботом, аналітичними модулями та підсистемами оптимізації. Належним чином спроектована база даних гарантує цілісність інформації, узгодженість транзакцій, масштабованість та ефективність виконання запитів у режимах реального часу. Для реалізації моделі даних було використано Microsoft SQL Server – промислову систему керування базами даних з розвиненими механізмами транзакційності, контролю доступу, індексації текстових полів та підтримкою складних аналітичних операцій. Процес проєктування здійснювався у середовищі SQL Server Management Studio (SSMS), що забезпечило можливість візуального моделювання структури бази даних, налаштування зв'язків, створення тригерів та автоматизації процедур.

Проєктування структури починалося з аналізу вимог до системи та визначення інформаційних потоків, що складають життєвий цикл запиту: реєстрація користувача, створення й надходження запиту, генерація відповіді чат-ботом, формування аналітичних результатів, реєстрація технічних і оптимізаційних подій. У результаті цього було виокремлено шість основних сутностей: Users, UserQueries, ChatbotResponses, Documents, OptimizationLog та

AnalyticsData. Вони утворюють взаємопов'язану логічну структуру, зображену на рис. 3.3.

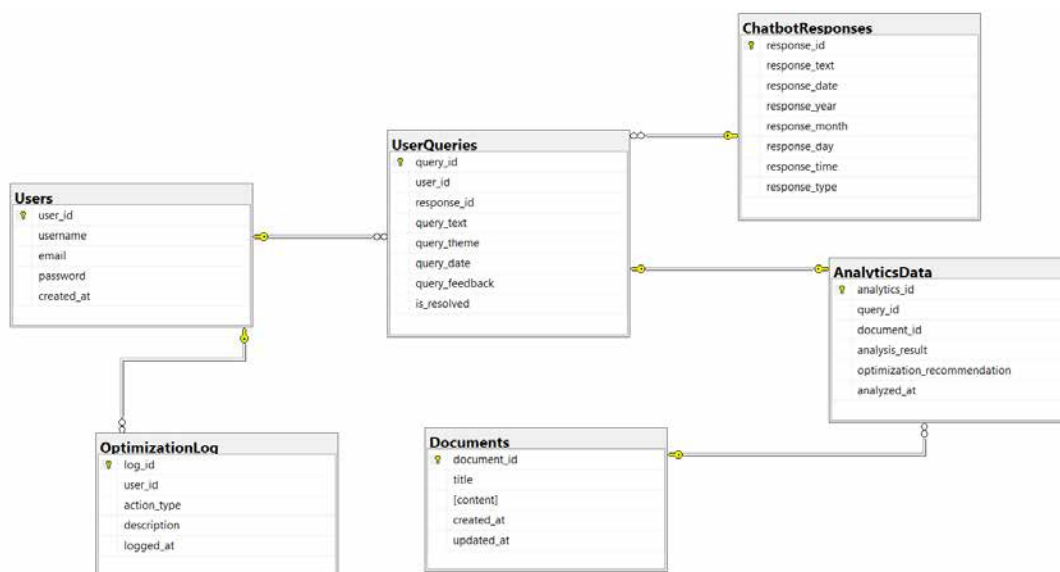


Рис. 3.3 Логічна модель бази даних системи інтелектуального аналізу

У центрі структури розташована таблиця UserQueries, яка акумулює всі запити користувачів. До її складу входять атрибути тексту запиту, його тематики, часу створення, а також поля для зворотного зв'язку та фіксації статусу обробки. Кожен запит пов'язаний із конкретним користувачем через зовнішній ключ user_id, що дозволяє формувати персоніфікований профіль взаємодії та будувати поведінкові моделі користувачів. Основні атрибути таблиці наведено у табл. 3.1.

Таблиця 3.1

Основні атрибути таблиці UserQueries

Поле	Опис	Тип даних
query_id	Унікальний ідентифікатор запиту	INT (PK)
user_id	Посилання на користувача	INT (FK → Users)
query_text	Текст запиту	NVARCHAR(MAX)
query_theme	Категорія/тема звернення	NVARCHAR(200)
query_date	Дата створення	DATETIME
query_feedback	Зворотний зв'язок користувача	NVARCHAR(MAX)
is_resolved	Ознака опрацювання	BIT
response_id	Посилання на відповідь чат-бота	INT (FK → ChatbotResponses)

Таблиця **Users** містить інформацію про зареєстрованих користувачів, включаючи ім'я, електронну адресу, пароль та дату створення облікового запису. Це дозволяє не лише відстежувати активність окремих користувачів, але й забезпечувати контроль доступу та авторизацію. Для підвищення рівня безпеки система може використовувати механізми хешування паролів, рольові моделі доступу та аудит змін у таблиці користувачів.

Таблиця **ChatbotResponses** використовується для зберігання відповідей, сформованих NLP-модулем або аналітичним сервером. Окрім тексту відповіді, у таблиці передбачено окремі поля для збереження року, місяця, дня та часу надходження відповіді. Це допомагає значно прискорити аналітичну обробку за часовими ознаками та полегшує побудову моделей сезонності чи часових тенденцій у зверненнях. Такі часові виміри є важливими для систем, у яких динаміка звернень нерівномірна.

Таблиця **Documents** містить семантичний контент, на який система спирається під час формування відповідей. Це можуть бути державні нормативні документи, навчальні матеріали, описові інструкції, довідники або витяги з веб-ресурсів. Запис містить назву документа, його вміст, дату створення та дату оновлення, що дозволяє контролювати актуальність наданої користувачам інформації.

Таблиця **OptimizationLog** виконує функцію системного журналу, у якому фіксуються події, що впливають на якість роботи системи. До таких подій належать зміни в налаштуваннях, оновлення документів, модифікації у моделях аналізу або оптимізаційні заходи, ініційовані підсистемами. Журнал дозволяє відтворити історію змін та аналізувати ефективність оптимізаційних дій.

Особливу роль відіграє таблиця **AnalyticsData**, що зберігає результати аналітичної обробки запитів: визначені категорії, індикатори релевантності, рекомендації для оптимізації пошуку або контенту, а також зв'язки між конкретним запитом і документом, на основі якого було сформовано відповідь. Ця таблиця є основою для механізму навчання системи та для вдосконалення її відповіді на основі історичних даних.

Для підтримання узгодженості між таблицями застосовано зовнішні ключі, що гарантують цілісність даних. Система може використовувати каскадні правила (*CASCADE DELETE*, *CASCADE UPDATE*), а також тригери для автоматичного оновлення похідних записів, коли змінюється інформація у базових таблицях. Це дозволяє уникнути появи «висячих» записів і забезпечує стабільність структури навіть при активних модифікаціях даних.

З огляду на наявність великих текстових полів і складних пошукових запитів застосовано повнотекстову індексацію (*FULLTEXT INDEX*). Таке індексування підвищує швидкість пошуку за ключовими словами, фразами чи контекстними збігами, що є критичним для систем інтелектуального аналізу, які працюють із природною мовою.

Створена логічна модель бази даних забезпечує цілісність даних, високу швидкодію виконання запитів, можливість масштабування та підтримку аналітичної обробки, що робить її ефективною основою для системи інтелектуального аналізу користувацьких звернень.

3.3 Розробка сховища даних для аналітичної обробки інформації

Сховище даних (*Data Warehouse, DWH*) є ключовим компонентом системи інтелектуального аналізу інформації про користувацькі запити, оскільки забезпечує централізоване, узгоджене та історично стабільне зберігання даних, що надходять із транзакційної бази. На відміну від операційної бази даних, яка оптимізована для виконання поточних транзакцій і обслуговування взаємодії з користувачем у режимі реального часу, сховище даних має іншу функціональну спрямованість: підтримку аналітичних запитів, агрегування статистики, визначення закономірностей, виявлення трендів та формування прогнозних моделей. Розмежування оперативного та аналітичного контурів дає змогу забезпечити високу продуктивність обробки запитів користувачів, водночас створюючи необхідні умови для глибокої аналітичної обробки накопичених даних.

Архітектура сховища даних у розроблюваній системі побудована за зірковою схемою (Star Schema), яка передбачає наявність однієї центральної таблиці фактів та декількох вимірних таблиць. Такий підхід є традиційним у побудові аналітичних систем, оскільки забезпечує високу швидкість виконання агрегованих запитів, спрощує побудову OLAP-кубів і дозволяє оптимізувати структурований доступ до великих обсягів інформації. Логічна структура розробленого сховища даних наведена на рис. 3.4, а приклади створення таблиць наведено в додатку А. Структура охоплює часовий, тематичний та відповідний виміри, а також центральну таблицю фактів, що акумулює зведені характеристики взаємодії користувачів із чат-ботом та веб-ресурсом.

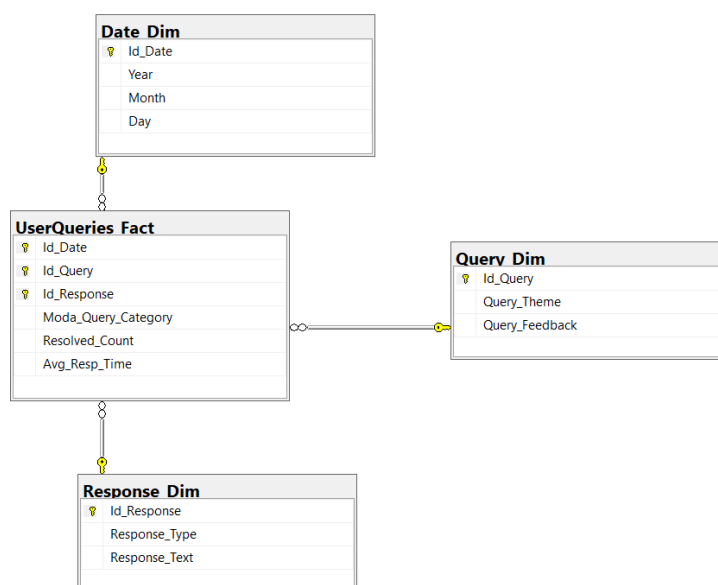


Рис. 3.4 Логічна модель сховища даних системи інтелектуального аналізу

У центрі зіркової схеми розміщується таблиця фактів `UserQueries_Fact`, яка акумулює узагальнені характеристики взаємодії користувачів із системою. До ключових показників належать: `Resolved_Count` (кількість успішно опрацьованих запитів), `Avg_Resp_Time` (середній час відповіді системи) та `Moda_Query_Category` (найчастіша тематика запитів за вибраний період). Окрім агрегованих атрибутів, таблиця фактів містить зовнішні ключі на вимірні таблиці, що дозволяє деталізувати аналітичні зрізи за часом, темою запиту та змістом відповіді.

Таблиця `Date_Dim` виступає часовим виміром, структуруючи дані за роком, місяцем та днем. Такий підхід забезпечує можливість побудови аналізу з різною глибиною деталізації: поквартально, помісячно, погодинно чи по окремих днях. Часові виміри відіграють стратегічну роль при дослідженні динаміки користувацьких звернень, що дозволяє формувати профілі пікових навантажень і планувати масштабування системи.

Вимір `Query_Dim` містить метадані про запити користувачів: тематику, категорію, тип звернення, а також оцінку задоволеності отриманою відповіддю (`Query_Feedback`). На основі цього виміру формуються тематичні кластери, аналізуються найчастіші інформаційні потреби користувачів, визначаються слабкі місця у поведінці чат-бота та розробляються рекомендації щодо удосконалення алгоритмів пошуку.

Таблиця `Response_Dim` фіксує інформацію про відповіді, створені системою. Вона містить тип відповіді (`Response_Type`) та її текстовий зміст (`Response_Text`). Цей вимір є важливим для аналізу ефективності різних форматів відповідей та оцінки їх релевантності. Зокрема, дані `Response_Dim` дозволяють визначити, чи краще система працює з текстовими поясненнями, короткими відповідями, посиланнями або витягами з документів.

Процес завантаження даних у сховище реалізований через трирівневу ETL-модель (`Extract, Transform, Load`), що забезпечує цілісність, узгодженість та уніфікацію інформації, перенесеної з транзакційної БД. На першому етапі `Extract` здійснюється вилучення інформації з таблиць `Users`, `UserQueries`, `ChatbotResponses`, `Documents` та `AnalyticsData`. Етап `Transform` охоплює очищення некоректних даних, нормалізацію текстових полів, виділення часових атрибутів, агрегацію показників, обчислення модальної категорії звернень, формування синтетичних ключів вимірів і приведення даних до структури, сумісної з DWH. На заключному етапі `Load` дані завантажуються до вимірних таблиць і таблиці фактів через SSIS-пакети, налаштовані на інкрементальне завантаження. Компоненти `Lookup` та `Conditional Split` запобігають дублюванню

даних і забезпечують унікальність записів у вимірних таблицях, що є важливою умовою для використання сурогатних ключів.

Сформоване сховище даних виступає аналітичним центром системи та дозволяє проводити багатовимірний аналіз, будувати OLAP-куби, формувати індикативні показники ефективності роботи чат-бота та веб-ресурсу, оцінювати темпи зростання кількості звернень, визначати найпоширеніші категорії запитів, аналізувати поведінкові патерни користувачів та моделювати динаміку звернень на основі часових рядів. Наявність структурованих фактів і вимірів створює підґрунтя для впровадження алгоритмів машинного навчання, включаючи класифікацію, кластеризацію та прогнозування. Результати, отримані на основі DWH, використовуються для оптимізації роботи системи, удосконалення моделей обробки природної мови та підвищення якості відповідей, що в цілому забезпечує еволюційний розвиток системи інтелектуального аналізу та її адаптивність до змін інформаційного середовища.

3.4 Алгоритмізація та програмування програмних модулів

Алгоритмізація та програмна реалізація модулів системи інтелектуального аналізу запитів користувачів формують основу її функціонування та забезпечують узгоджену роботу всіх компонентів – від інтерфейсу користувача до серверної частини, бази даних і сховища даних. На цьому етапі було розроблено комплекси алгоритмів, що забезпечують обробку запитів у реальному часі, класифікацію текстових повідомлень, пошук релевантної інформації у корпоративних документах, формування відповідей та накопичення аналітичних записів у DWH. Алгоритмічна модель визначає порядок, взаємозв'язки та умови виконання дій усередині системи, формалізує логіку взаємодії між її підсистемами та гарантує коректність процесів, пов'язаних з обробкою запитів.

Алгоритм роботи системи ґрунтується на послідовному виконанні етапів, що охоплюють приймання запиту, його валідацію, попередню текстову обробку, семантичний аналіз, формування SQL-запитів або пошук у повнотекстових

індексах, побудову відповіді та логування результатів у транзакційній БД. На рис. 3.5 представлено узагальнену схему алгоритму обробки запиту.

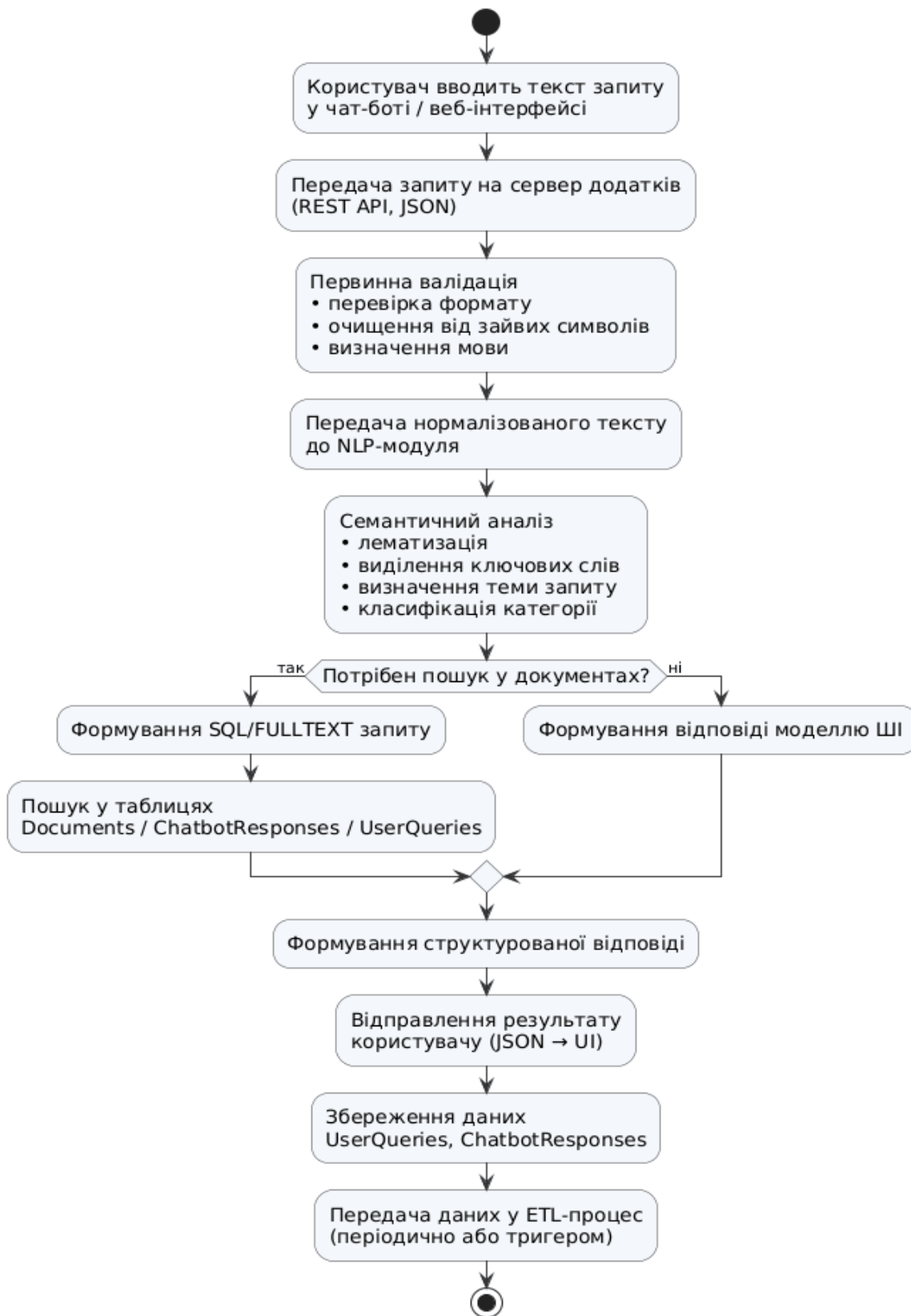


Рис. 3.5 Алгоритм загальної роботи системи обробки запитів

Алгоритм починається з формування текстового повідомлення користувачем у чат-боті або веб-інтерфейсі, після чого запит передається на сервер додатків, де виконується початкова фільтрація, нормалізація й визначення мовних характеристик. Після цього запит аналізується NLP-модулем, який здійснює лематизацію, виділення ключових слів, побудову векторного подання та класифікацію за тематичними категоріями. На основі результатів аналізу формується звернення до бази даних, зокрема до таблиць Documents, ChatbotResponses або UserQueries, що забезпечує пошук структурованої і неструктурованої інформації у джерелах знань системи.

У випадку знаходження релевантних даних модуль формування відповіді узагальнює отриману інформацію, адаптує її до формату короткого повідомлення та відправляє назад користувачу. Паралельно з цим дані про запит і відповідь фіксуються у транзакційній структурі: текст запиту, тематика, дата й час обробки, тип відповіді, оцінка користувача та інші параметри зберігаються у таблицях UserQueries і ChatbotResponses, що дозволяє забезпечити подальший аналіз з боку аналітичного модуля. Надалі ці записи проходять ETL-процес, у рамках якого вони вилучаються, трансформуються й завантажуються до сховища даних у таблиці Date_Dim, Query_Dim, Response_Dim та UserQueries_Fact, що надає можливість виконувати багатовимірний аналіз і побудову OLAP-звітів.

У програмній реалізації ключовими стали чотири програмні модулі: модуль взаємодії з користувачем, модуль NLP-обробки, модуль доступу до даних та модуль аналітичної обробки. Модуль взаємодії з користувачем реалізовано як REST-інтерфейс, що приймає запити через HTTP-протокол і передає структури даних у JSON-форматі. NLP-модуль включає процедури синтаксичного аналізу, класифікації запиту та вибору відповідної моделі пошуку, забезпечуючи перехід від природної мови до структурованих елементів, придатних для автоматичної обробки. Модуль доступу до бази даних будується на використанні параметризованих SQL-запитів та ORM-підходу, що підвищує безпеку системи й мінімізує імовірність SQL-ін'єкцій. Аналітичний модуль

забезпечує накопичення статистичних відомостей, таких як середній час відповіді або частота появи певних тематичних категорій, і передає їх у DWH.

На рис. 3.6 подано фрагмент алгоритмічної моделі обробки запитів NLP-модулем.

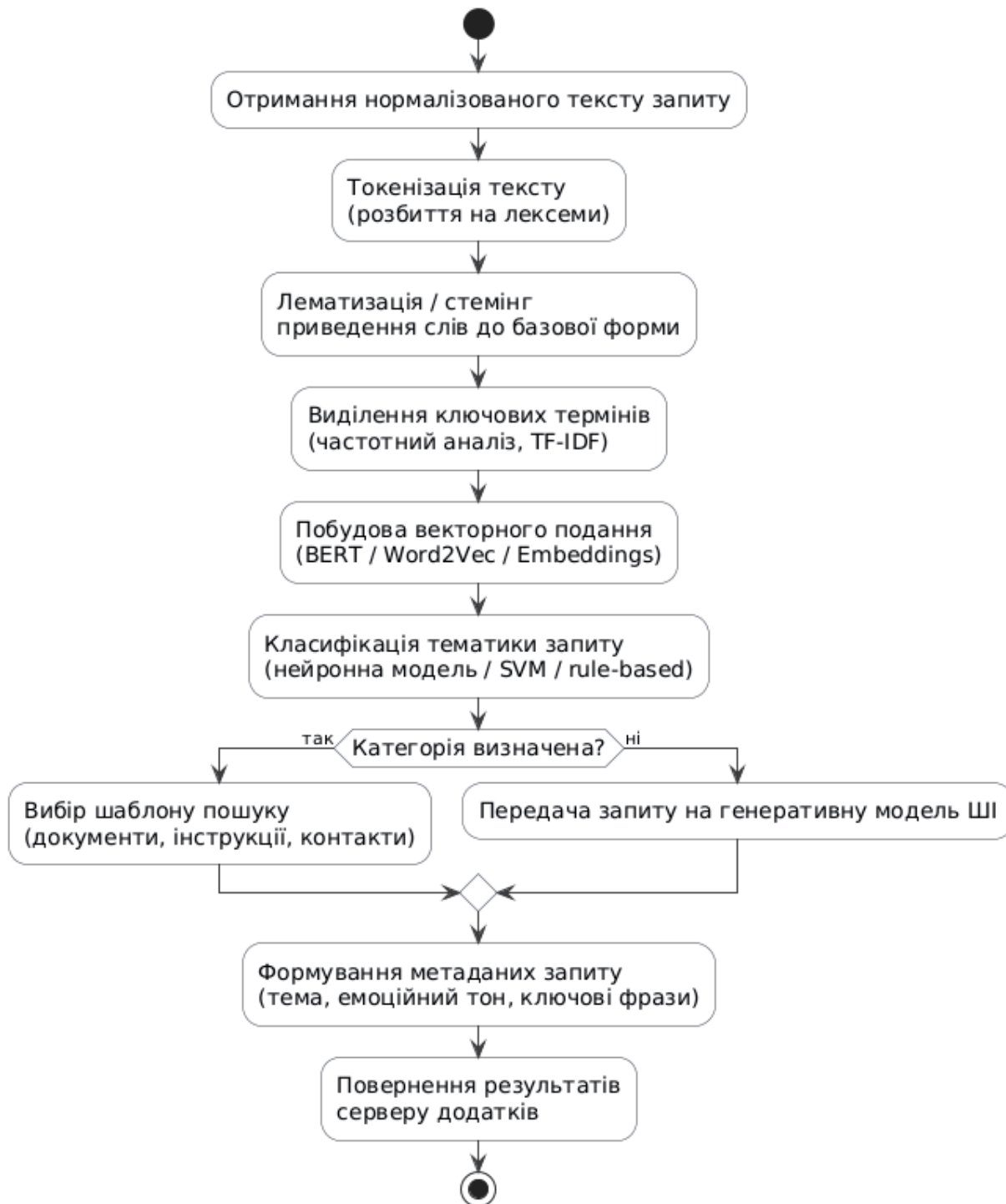


Рис. 3.6 Алгоритм роботи NLP-модуля обробки запитів

В таблиці дод. А.2 окреслено основні процедури, що виконуються на кожному етапі. Такий підхід дає змогу не лише формалізувати логіку роботи системи, але й уніфікувати окремі кроки для подальшої оптимізації або масштабування. Через високу частку операцій із текстовими даними було реалізовано інкрементальне оновлення індексів, що дає змогу обробляти запити користувачів у режимі реального часу без перевантаження серверної частини.

Фрагмент програмного коду, що демонструє спрощену реалізацію алгоритму обробки запиту на сервері додатків, наведено у дод. А.1. Код реалізує послідовність дій: вилучення тексту запиту, передавання його моделі ШІ, отримання відповіді, збереження результатів у транзакційній БД та відправлення користувачу. Приклад відображає основну логіку, яка може бути розширена підключенням декількох моделей аналізу, реалізацією механізмів кешування та обробки виняткових ситуацій.

Узагальнюючи, алгоритми системи забезпечують повний цикл обробки запитів користувачів: від введення та лінгвістичного аналізу до збереження структурованих результатів і передавання їх до сховища даних. Це створює фундамент для побудови адаптивної, масштабованої та інтелектуальної системи, здатної покращувати власну ефективність на основі накопичених даних і формувати релевантні відповіді у відповідності до інформаційних потреб користувачів. Завдяки взаємодії алгоритмічних модулів забезпечується цілісність інформаційного потоку, висока швидкодія та можливість аналітичного опрацювання даних у розрізі значущих показників ефективності системи.

Висновки до розділу 3

У межах розділу було сформовано цілісну архітектурну та інформаційну модель системи інтелектуального аналізу запитів користувачів, що включає проєктування транзакційної бази даних, розробку багатовимірного сховища даних та визначення механізмів взаємодії між усіма програмними модулями. Запропонована структура забезпечує узгоджений рух даних від моменту введення запиту до його зберігання й аналітичного опрацювання, підтримує

цілісність інформації, високу швидкість та масштабованість системи. Розроблені моделі зберігання даних дають змогу накопичувати подійні записи, формувати історичні вибірки, виконувати класифікацію та оцінку ефективності роботи інтелектуального сервісу.

Алгоритмічне забезпечення охоплює послідовність процедур обробки запитів, включно з первинною валідацією, лінгвістичним аналізом, семантичним розпізнаванням та вибором оптимальної стратегії пошуку інформації. Створені програмні модулі реалізують коректну взаємодію з базою даних, підтримують механізми ETL-обробки та забезпечують формування структурованих відповідей у режимі реального часу. Сукупність розроблених алгоритмів, моделей і програмних компонентів створює фундамент для подальшого удосконалення системи, зокрема підвищення точності пошуку, автоматичного навчання моделей і розширення аналітичних можливостей.

4. ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ РОБОТИ СИСТЕМИ ТА АНАЛІЗ ЕФЕКТИВНОСТІ ЇЇ ФУНКЦІОНУВАННЯ

4.1. Методи та реалізація модулів семантичного аналізу користувачьких запитів

Семантичний модуль системи відіграє ключову роль у забезпеченні точного аналізу наміру користувача та визначенні релевантного контексту запиту. Його реалізація ґрунтується на поєднанні класичних методів інтелектуального аналізу даних (класифікація, асоціативні правила, кластеризація) та сучасних технологій обробки природної мови. Такий підхід дає змогу інтегрувати результати, отримані на етапі дослідження в курсовому проєкті – зокрема 1-Rule, наївний Байес, Apriori та K-Means – у структуру єдиного модулю семантичної обробки запитів користувачів.

Основою роботи модуля є попередня підготовка текстів, що включає нормалізацію, токенізацію, лематизацію та побудову векторних подань. Представлення текстів у вигляді embedding-векторів створює можливість застосовувати машинне навчання у багатовимірному просторі ознак, що підвищує здатність системи розпізнавати семантично подібні запити, навіть якщо вони сформульовані різними словами. Саме на цьому етапі інтегруються методи класифікації, апробовані у курсовому проєкті. Модель 1-Rule використовується для визначення початкової категорії запиту на основі найбільш інформативної ознаки, наївний Байес застосовується для уточнення класифікації в умовах шумових та неповних даних, а правила Apriori дозволяють виявити приховані співзалежності між типами запитів і контекстами відповідей. У сукупності це створює шар інтелектуальної логіки, який працює до побудови остаточного семантичного подання.

Рисунок 4.1 відображає узагальнену схему роботи модуля. На першому рівні здійснюється вилучення ключових ознак із запиту, після чого дані послідовно проходять класифікаційний шар, модуль асоціативних правил та

семантичний шар embedding-моделі. Усі результати узгоджуються між собою, а система визначає інтегральний намір користувача.

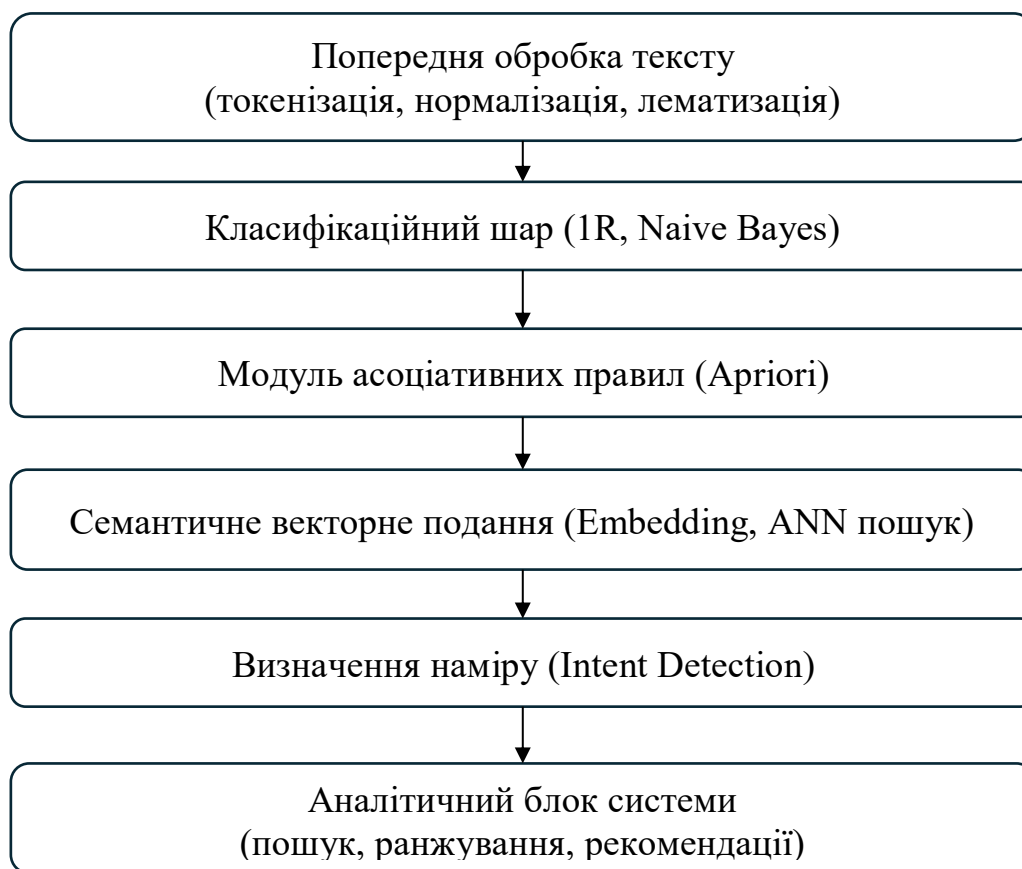


Рис. 4.1 Структурна схема модуля семантичного аналізу запитів

Подальший етап роботи модуля передбачає розрахунок міри семантичної близькості між запитом і контентом ресурсу. Для цього використано евклідову відстань та косинусну подібність, що дозволяє порівнювати векторні представлення як окремих фрагментів тексту, так і цілих документів. Кластерна модель K-Means, побудована під час курсових досліджень, використовується для групування векторів у тематично пов'язані кластери, що прискорює пошук і дозволяє швидко визначити, до якої групи змістових областей належить конкретний запит користувача.

У таблиці 4.1 наведено приклад інтерпретації результатів кластеризації запитів та їх відповідності до тематичних груп. Така інтерпретація відіграє важливу роль у побудові адаптивної структури системи, що здатна ефективно масштабуватися при збільшенні кількості запитів та обсягу контенту.

Таблиця 4.1

Приклад тематичної кластеризації запитів користувачів

Кластер	Основна тема	Частка запитів (%)	Приклади фраз
C1	Навігаційні запити	34	«Де знайти документ...», «Як перейти до...»
C2	Нормативна база	21	«Положення про...», «Регламент роботи...»
C3	Методичні матеріали	18	«Інструкція з...», «Методичні рекомендації...»
C4	Технічні питання	15	«Не працює...», «Помилка доступу...»
C5	Академічна інформація	12	«Навчальний план...», «Розклад...»

Для оцінювання якості роботи модуля застосовано метрики точності, повноти та міри F1, розраховані на тестовій вибірці запитів. Діаграма на рисунку 4.2 демонструє співвідношення цих метрик для класифікаційного шару, а також для інтегральної семантичної моделі, що доводить перевагу комбінованого підходу.

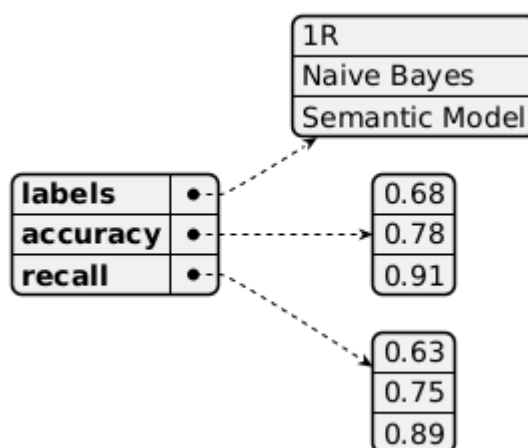


Рис. 4.2 Порівняння метрик точності та повноти для різних методів аналізу запитів

Рисунок 4.2 відображає узагальнене порівняння роботи трьох підходів до обробки користувацьких запитів – правил 1R, наївного Байєса та інтегрованої семантичної моделі. Як видно з діаграми, найпростіший класифікаційний

механізм IR демонструє лише базові значення точності та повноти, що дозволяє використовувати його переважно як орієнтовний індикатор основної тематики запиту. Модель наївного Байєса працює стабільніше та забезпечує вищі показники завдяки ймовірнісному характеру оцінювання ознак і здатності коректно обробляти частково неповні або шумові дані.

Найкращі результати забезпечує семантична модель, у якій поєднано методи класифікації, асоціативних правил та векторних представлень тексту. Завдяки використанню embedding-перетворень і модулю визначення наміру, цей підхід досягає найвищого співвідношення точності та повноти, що підтверджує доцільність застосування комбінованої архітектури для аналізу запитів у складних інформаційних системах.

Узагальнюючи отримані результати, модуль семантичного аналізу забезпечує цілісну обробку запиту на основі багаторівневої інтеграції класифікаційних і семантичних методів. Такий підхід формує основу інтелектуальної взаємодії користувача з веб-ресурсом, підвищує точність і зрозумілість результатів пошуку та створює підґрунтя для масштабування системи в межах великих інформаційних платформ.

4.2. Оцінювання ефективності моделей у системі семантичного аналізу запитів

4.2.1 Загальна методика оцінювання моделей семантичного аналізу

Оцінювання моделей, залучених до семантичної обробки користувацьких запитів, здійснюється на основі порівняння статистичних, ймовірнісних і семантичних підходів, інтегрованих у багаторівневу структуру аналітичного модуля. Такий підхід дає змогу оцінити не лише окремі алгоритми, а й їхню ефективність у складі єдиної системи, де кожен метод виконує специфічну роль у формуванні наміру користувача.

У межах оцінювання враховувалися ключові параметри роботи моделей: якість класифікації, здатність до узагальнення, стійкість до шумових або

неповних даних та можливість виявлення структурних і тематичних залежностей у тексті. Залучення різних типів запитів – від нормативних до технічних та інформаційних – забезпечило репрезентативність тестової вибірки та дозволило оцінити поведінку моделей у реалістичних умовах.

Для кількісного вимірювання ефективності застосовано показники точності (Precision), повноти (Recall) та F1-міри, що дозволило зіставити роботу моделей за однаковими критеріями. Оцінювання включало аналіз не лише підсумкових значень, а й типових помилок, зокрема хибнопозитивних рішень та випадків втрати релевантних відповідей, що є критичним для систем, орієнтованих на коректне визначення інтенції користувача.

Отримані результати дали змогу визначити, які алгоритмічні компоненти забезпечують найвищу точність у межах інтегрованої семантичної моделі, а які – виконують допоміжні функції, підсилюючи структуру ознак або уточнюючи контекстні залежності.

4.2.2 Використання алгоритму 1-Rule для попередньої класифікації запитів

Алгоритм 1-Rule (OneR) розглядається як базовий інструмент первинного аналізу користувацьких запитів та формування інтерпретованих залежностей між їх характеристиками та продуктивністю обробки. Попри свою простоту, цей метод дає змогу швидко виокремити ключові атрибути, що найбільше впливають на результат, і сформувані узагальнене правило, здатне виконувати функцію початкового класифікаційного шару в системі семантичного аналізу.

У межах експериментального дослідження алгоритм було застосовано для класифікації запитів за ознакою швидкості відповіді. Цільова змінна поділялася на два класи: високої (H) та низької (L) продуктивності, що визначалися відносно середнього часу обробки по всій тестовій вибірці. Як незалежні змінні було проаналізовано Query_Feedback, Query_Theme та Response_Type, що відображають оцінку користувача, тематику запиту та тип наданої відповіді відповідно.

Алгоритм 1-Rule передбачав побудову одного оптимального правила на основі тієї змінної, яка забезпечує найменшу кількість помилок класифікації. Для кожного значення атрибутів обчислювалася частота належності до класів H та L, після чого визначалася найімовірніша класифікація для кожного випадку. Схему роботи алгоритму подано на рисунку 4.3, що відображає етапи побудови правила – від вибору атрибута до формування узагальненого рішення.

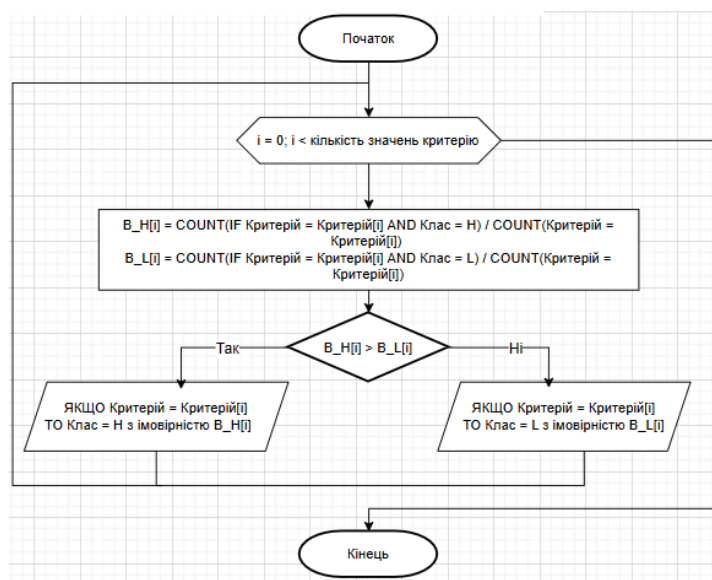


Рис. 4.3 Блок-схема алгоритму

Результати, отримані в ході реалізації 1-Rule, продемонстрували наявність чітких закономірностей між поведінкою користувача та ефективністю роботи системи. Зокрема, найбільш інформативною ознакою виявився Query_Feedback, що забезпечив найвищу точність класифікації. Запити з негативною оцінкою переважно належали до класу низької продуктивності, тоді як позитивний відгук значною мірою асоціювався з високою швидкістю відповіді. Аналогічно, тематика запиту та тип відповіді також демонстрували змістові залежності: запити категорії «документи» частіше отримували швидкі та чіткі відповіді, тоді як технічно складні звернення вимагали більше часу на опрацювання.

Графічні результати класифікації за різними критеріями наведено на рисунках 4.4, що відображають розподіл класів для кожного атрибута. Отримані правила дозволили виявити найбільш стабільні предиктори продуктивності, які

можуть бути ефективно використані для формування початкових припущень щодо наміру користувача та подальшої оптимізації сценаріїв обробки.

Аналіз за критерієм: Response_Type

а)

Response_Type	Менше середнього	Більше середнього	Всього	Клас	Імовірність %
0 посилаання	18	45	63	H	71.43
1 інструкція	67	112	179	H	62.57
2 рекомендація	40	22	62	L	35.48
3 пояснення	99	140	239	H	58.58
4 документ	8	6	14	L	42.86

Аналіз за критерієм: Query_Feedback

б)

Query_Feedback	Менше середнього	Більше середнього	Всього	Клас	Імовірність %
0 позитивний відгук	104	228	332	H	68.67
1 нейтральний	55	59	114	H	51.75
2 негативний досвід	63	28	91	L	30.77

Аналіз за критерієм: Query_Theme

в)

Query_Theme	Менше середнього	Більше середнього	Всього	Клас	Імовірність %
0 освіта	53	61	114	H	53.51
1 право	6	2	8	L	25.00
2 податки	32	37	69	H	53.62
3 держпослуги	15	6	21	L	28.57
4 документи	16	45	61	H	73.77
5 медицина	32	53	85	H	62.35
6 соцзахист	58	89	147	H	60.54
7 транспорт	3	3	6	H	50.00
8 технічна проблема	2	10	12	H	83.33
9 житло_послуги	12	7	19	L	36.84
10 Медицина	0	2	2	H	100.00
11 Освіта	0	2	2	H	100.00
12 Транспорт	0	2	2	H	100.00
13 інше	3	6	9	H	66.67

Рис. 4.4 Результати класифікації за алгоритмом 1-Rule: а) по критерію Response_Type; б) по критерію Query_Feedback; в) по критерію Query_Theme

Застосування алгоритму 1-Rule підтвердило його доцільність як інструмента попередньої структуризації даних. Він забезпечує інтерпретовані результати, мінімізує обчислювальні витрати та формує основу для подальшого аналізу в рамках складніших ймовірнісних і семантичних моделей. Отримані

залежності стали вихідною точкою для уточнення ознак і подальшого застосування методів наївного Байєса та асоціативних правил.

4.2.3. Оцінювання продуктивності чат-бота методом наївного Байєса

Метод наївного Байєса є одним із базових і водночас ефективних імовірнісних алгоритмів класифікації, що широко використовується у задачах обробки текстових даних. Його застосування у межах аналізу продуктивності чат-бота дозволяє оцінити взаємозв'язок між характеристиками запитів, типом відповіді та підсумковим класом ефективності обробки. Незважаючи на припущення про незалежність ознак, модель демонструє високу стійкість до шумових даних і здатність виявляти закономірності навіть за умов частково заповнених атрибутів.

У рамках експериментального дослідження було сформульовано задачу класифікації запитів за цільовою змінною `Perf_Class`, що відображає продуктивність системи: високий клас (H) відповідає часові відповіді, що не перевищує середнє значення по вибірці, тоді як низький клас (L) позначає повільне опрацювання запиту. Як незалежні змінні використано `Query_Feedback` (оцінка користувача) та `Response_Type` (тип наданої відповіді). Формалізація задачі ґрунтувалася на теоремі Байєса, яка дає змогу обчислити ймовірність належності запиту до конкретного класу за умови спостережуваних значень ознак. Узагальнена блок-схема алгоритму подана на рисунку 4.5, де відображено етапи обробки: від оцінювання апріорних ймовірностей до вибору найімовірнішого класу.

Реалізація алгоритму прямого програмування дала змогу отримати кількісні оцінки ймовірностей переходу до класів H та L для різних комбінацій ознак. Приклад результатів наведено на рисунку 4.6.

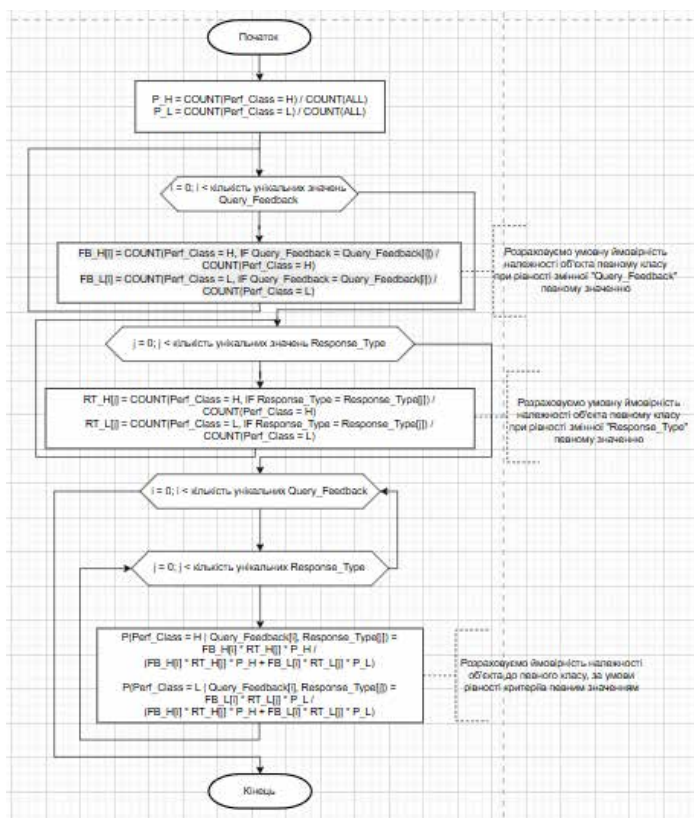


Рис. 4.5 Блок схема алгоритму

Query Feedback	Response Type	Клас	Імовірність H	Імовірність L
позитивний відгук	посилання	H	79.065487	20.934513
нейтральний	інструкція	H	55.069597	44.930403
негативний досвід	рекомендація	L	15.016026	84.983974
позитивний відгук	пояснення	H	69.780240	30.219760
негативний досвід	пояснення	L	31.885688	68.114312
позитивний відгук	інструкція	H	71.468205	28.531795
позитивний відгук	рекомендація	L	46.568908	53.431092
нейтральний	пояснення	H	53.048822	46.951178
нейтральний	посилання	H	64.888162	35.111838
нейтральний	рекомендація	L	29.896943	70.103057
негативний досвід	інструкція	L	33.678569	66.321431
позитивний відгук	документ	H	53.677781	46.322219
негативний досвід	посилання	L	43.364187	56.635813
нейтральний	документ	L	36.184320	63.815680

Рис. 4.6 Результат виконання алгоритму

Аналіз засвідчив, що запити з негативною оцінкою користувача та відповідями типу «рекомендація» або «пояснення» найчастіше відносяться до класу низької продуктивності – з імовірністю понад 80 % у низці комбінацій. Навіть за нейтральної оцінки чат-бота домінування класу L зберігалося у випадках загальних або неструктурованих типів відповіді. Це свідчить про те, що

саме емоційно-оцінні та змістові аспекти взаємодії мають суттєвий вплив на сприйняття системи користувачем та пов'язані з реальними затримками в опрацюванні запитів.

Для перевірки узгодженості результатів застосовано модель CategoricalNB із бібліотеки *scikit-learn*, результати роботи якої подано на рис. 4.7.

Query	Feedback	Response Type	Клас	Імовірність H	Імовірність L
позитивний	відгук	посилання	H	78.6265	21.3735
нейтральний	відгук	інструкція	H	55.1567	44.8433
негативний	досвід	рекомендація	L	15.6514	84.3486
позитивний	відгук	пояснення	H	69.8155	30.1845
негативний	досвід	пояснення	L	32.4576	67.5424
позитивний	відгук	інструкція	H	71.4588	28.5412
позитивний	відгук	рекомендація	L	47.1767	52.8233
нейтральний	відгук	пояснення	H	53.1896	46.8104
нейтральний	відгук	посилання	H	64.3776	35.6224
нейтральний	відгук	рекомендація	L	30.4954	69.5046
негативний	досвід	інструкція	L	34.2184	65.7816
позитивний	відгук	документ	H	54.7115	45.2885
негативний	досвід	посилання	L	43.3204	56.6796
нейтральний	відгук	документ	L	37.2444	62.7556

Рис. 4.7 Результат алгоритму моделі CategoricalNB

Отримані значення підтвердили загальну закономірність: позитивний фідбек користувача у поєднанні з чітким типом відповіді (інструкція, посилання) суттєво підвищує ймовірність належності до класу високої продуктивності. Найбільшу ймовірність було зафіксовано у комбінації «позитивний відгук» та «посилання» – майже 80 %, що вказує на доцільність використання прямих та однозначних способів передачі інформації для скорочення часу обробки.

Порівняння двох підходів – прямої реалізації формул Байєса та моделі CategoricalNB – засвідчило їхню високий рівень узгодженості. Це підтверджує стабільність виявлених закономірностей і дає підстави для подальшого їх використання у моделюванні поведінки системи. Отримані результати також демонструють, що задоволеність користувача та структура відповіді є індикаторами не лише якості взаємодії, а й реальної ефективності роботи серверної частини чат-бота.

Таким чином, застосування методу наївного Байєса дозволило отримати формально обґрунтовану модель визначення продуктивності системи на основі

ключових атрибутів запиту. Його результати сформували основу для подальшого аналізу закономірностей між темами запитів та типами відповіді та сприяли уточненню ознак, використовуваних у класифікаційних і семантичних модулях. Методи Байєса продемонстрували високу стійкість, інтерпретованість та застосовність у задачах оптимізації роботи діалогових систем у адаптивних інформаційних середовищах.

4.2.4. Аналіз закономірностей у даних методом асоціативних правил

Метод асоціативних правил дає змогу виявляти стійкі залежності між ознаками запитів та характеристиками відповідей, які не завжди можуть бути зафіксовані традиційними класифікаційними моделями. На відміну від методів наївного Байєса або 1-Rule, що спрямовані на передбачення цільового класу, асоціативні правила дозволяють встановити приховані структурні зв'язки між елементами даних, визначити їхню спільну появу та оцінити інтенсивність такої взаємодії за допомогою формальних метрик.

У межах аналізу було використано класичний алгоритм Apriori, який формує кандидатні набори ознак та перевіряє їх на предмет відповідності встановленим порогам support, confidence і lift. Ці метрики дають змогу оцінити частоту появи конкретної комбінації атрибутів та її інформативність для подальшої інтерпретації. На рис. 4.8 наведено загальну схему процесу побудови асоціативних правил – від фільтрації частих наборів до формування остаточного правила «antecedent → consequent».

Знайдені асоціативні правила:

Правило: [Тема: документи] → [Тип: інструкція], підтримка: 0.06, довірчість: 0.54, ліфт: 1.68
 Правило: [Тема: медицина] → [Тип: пояснення], підтримка: 0.07, довірчість: 0.47, ліфт: 1.10
 Правило: [Тема: освіта] → [Тип: пояснення], підтримка: 0.09, довірчість: 0.44, ліфт: 1.02
 Правило: [Тема: освіта] → [Тип: інструкція], підтримка: 0.06, довірчість: 0.31, ліфт: 0.96
 Правило: [Тема: податки] → [Тип: пояснення], підтримка: 0.06, довірчість: 0.51, ліфт: 1.18
 Правило: [Тема: соцзахист] → [Тип: пояснення], підтримка: 0.13, довірчість: 0.48, ліфт: 1.13

Рис. 4.8 Результати виконання власного алгоритму пошуку асоціативних правил для мінімальної підтримки 5%

Для дослідження до моделі було включено такі атрибути: Query_Theme, що визначає тематику запиту; Response_Type, який характеризує структуру та

спосіб подання відповіді; та Query_Feedback, що відображає оцінку користувача після отримання відповіді. Це дало змогу зосередитися на змістових взаємозв'язках, які впливають на якість взаємодії та можуть бути використані для оптимізації сценаріїв роботи чат-бота.

Отримані правила виявили наявність чітких патернів. Наприклад, тематика «документи» статистично значуще поєднувалася з відповідями типу «інструкція», що відповідає очікуваному сценарію: користувачі, які шукають нормативні чи інформаційні матеріали, одержують структуровані та короткі інструкції з високим рівнем задоволеності. Правило «медицина → пояснення» також мало високий рівень підтримки, що підтверджує потребу користувачів у розширених текстових відповідях, коли йдеться про складні або чутливі теми.

Особливу увагу привернули правила, де наслідком виступала оцінка Query_Feedback. Правила на кшталт «загальна тема + рекомендація → позитивний фідбек» демонструють, що системи, здатні генерувати конкретні поради у межах широких тематичних груп, підвищують задоволеність користувачів. Водночас комбінації «технічна тема + пояснення → нейтральний фідбек» свідчать про складність передачі технічної інформації в межах діалогової взаємодії та потребують окремого коригування шаблонів відповіді.

Аналіз значень lift виявив і нетривіальні закономірності. Так, деякі правила, що мали помірну підтримку, демонстрували підвищену залежність між темою запиту та типом відповіді, перевищуючи випадкову співпадність у кілька разів. Це свідчить про існування стабільних взаємозв'язків між структурою запиту та оптимальним форматом відповіді, які можуть бути використані для автоматизації вибору відповіді у майбутніх версіях системи.

Таким чином, застосування методу асоціативних правил дало змогу виявити значущі змістові закономірності між темами запитів, типами відповіді та оцінками користувачів. Отримані результати підсилюють висновки попередніх підрозділів та розширюють можливості моделювання поведінки користувачів у реальних сценаріях. Виявлені патерни можуть бути інтегровані у

модуль рекомендацій та використані для підвищення ефективності семантичного аналізу в системі, а також для оптимізації шаблонів відповідей.

4.2.5. Кластеризаційний аналіз структури користувацьких запитів

Кластеризація відіграє важливу роль у виявленні прихованих структур у даних, де попереднє маркування відсутнє або є недостатнім. У контексті семантичного аналізу запитів кластеризаційний підхід дає змогу визначити типові шаблони взаємодії користувачів із системою, виявити відмінності між групами звернень та оцінити зміни в інтенсивності використання окремих тематичних або функціональних сегментів чат-бота. Такі результати дозволяють підсилити моделі класифікації, а також коригувати логіку формування відповідей відповідно до поведінкових характеристик різних груп користувачів.

Для проведення аналізу було використано алгоритм **K-Means**, який належить до центричних методів неконтрольованого навчання. Його застосування обґрунтовується тим, що структура вибірки містить як числові параметри, зокрема середній час відповіді та кількість вирішених звернень, так і категоріальні ознаки, які було попередньо оцифровано шляхом one-hot encoding. Усі числові змінні були нормалізовані за допомогою StandardScaler, що забезпечило коректне порівняння компонентів різних масштабів.

Ключовим етапом кластеризації стало визначення оптимального значення параметра K . Для цього було використано метрику Silhouette Score, яка вимірює внутрішню згуртованість кластерів та їхню відокремленість. Результати, наведені на рисунках 4.9 та 4.10, демонструють, що найвищий приріст метрики спостерігався при переході від $K=2$ до $K=3$, а подальше збільшення кількості кластерів не покращувало сегментацію. Це дозволило обрати оптимальним значенням $K = 3$, оскільки така кількість груп забезпечила баланс між інформативністю результатів та мінімізацією надлишкової деталізації.

Пошук оптимального K...
Обчислення silhouette score...

Silhouette Score по k:

	k	Silhouette Score	Δ Score
0	2	0.0918	nan
1	3	0.1620	0.0702
2	4	0.1823	0.0203
3	5	0.2161	0.0338
4	6	0.2582	0.0422
5	7	0.3272	0.0690
6	8	0.3812	0.0540
7	9	0.3871	0.0058
8	10	0.4120	0.0249
9	11	0.3885	-0.0235
10	12	0.4295	0.0410
11	13	0.4271	-0.0024
12	14	0.4003	-0.0268
13	15	0.3891	-0.0111

Рис. 4.9 Обчислення silhouette score та його приросту

Обрано оптимальну кількість кластерів: K = 3 (на основі максимального приросту Δ Score = 0.0702).



Рис. 4.10 Залежність Silhouette Score від кількості кластерів (k)

Отриманий розподіл об'єктів між трьома кластерами (рис. 4.11) показав наявність одного великого кластера з типовими зверненнями та двох менших груп, що містили специфічні запити.

Розподіл записів по кластерах:

```
Cluster
0    69
1   366
2   122
Name: count, dtype: int64
```

Центроїди кластерів (повністю):

	Resolved_Count	Avg_Resp_Time	Moda_Query_Category_Транспорт	Moda_Query_Category_держпослуги	Moda_Query_Category_документи	Moda_Query_Category_...
0	-0.01	0.11	-0.06	-0.20	-0.35	
1	-0.03	-0.09	0.03	0.10	0.18	
2	0.09	0.22	-0.06	-0.20	-0.35	

Рис. 4.11 Центроїди кластерів

Аналіз центроїдів дав змогу інтерпретувати ці групи:

- Кластер 1 мав збалансовану структуру без яскраво виражених піків, що вказує на масові стандартні звернення користувачів. Значення середнього часу відповіді та частки позитивного фідбеку були близькі до середніх по вибірці.
- Кластер 0 характеризувався підвищенням значень ознак, пов'язаних із темою «податки» та збільшеним часом обробки, що свідчить про складність таких запитів і більшу інтенсивність звернень у цій категорії.
- Кластер 2 був асоційований із запитами на тему «освіта», мав вищу частку позитивного фідбеку та помірну тривалість обробки, що свідчить про більш ефективну взаємодію в межах цієї категорії. Візуалізація в координатах першої та другої головних компонент (рис. 4.12) підтвердила відокремленість і внутрішню компактність кожної групи.

Додатково було виконано аналіз розподілу середнього часу відповіді (рис. 4.13), який продемонстрував наявність викидів у межах складних запитів. Графіки розподілу Response_Type, Query_Theme та Query_Feedback у кожному кластері (рис. 4.14–4.16) дали змогу уточнити поведінкові характеристики користувачів та виявити тематичні зони, що потребують оптимізації.

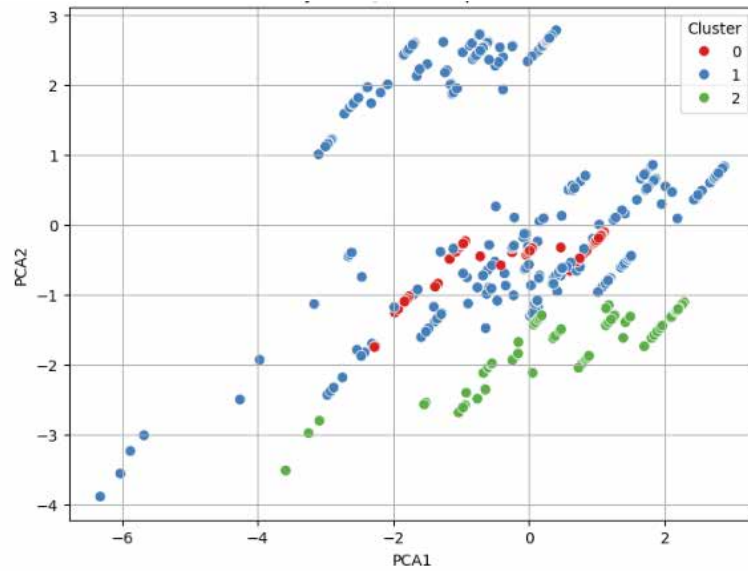


Рис. 4.12 Візуалізація кластерів (PCA)

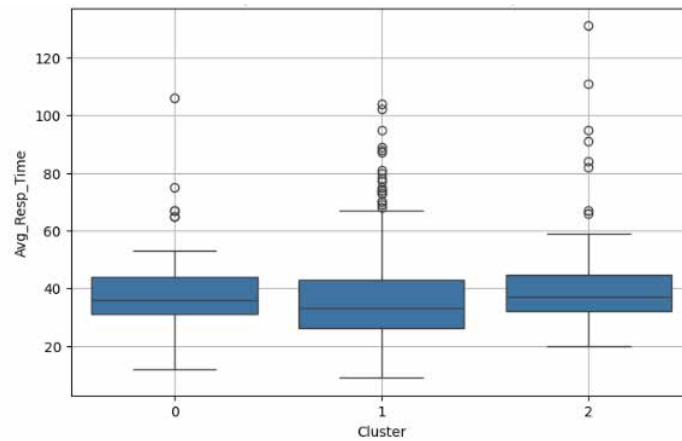


Рис. 4.13 Вохрлот середнього часу відповіді

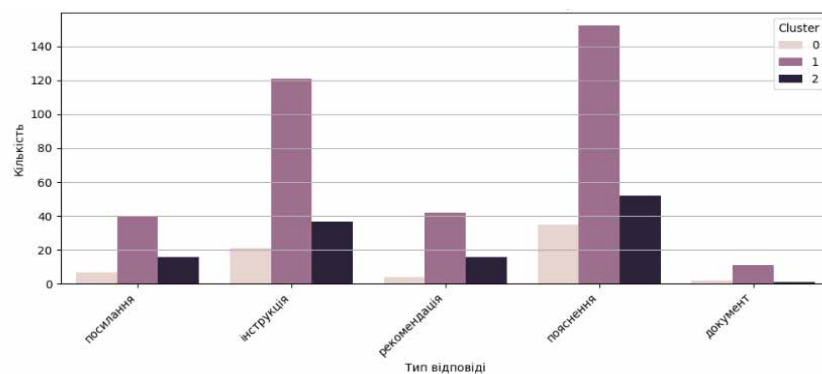


Рис. 4.14 Розподіл типів відповідей по кластерах

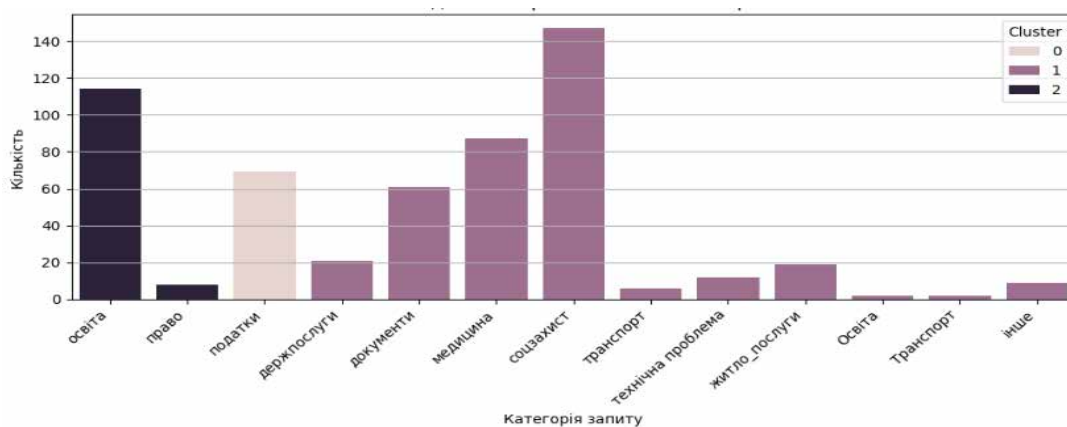


Рис. 4.15 Розподіл категорій запитів по кластерах

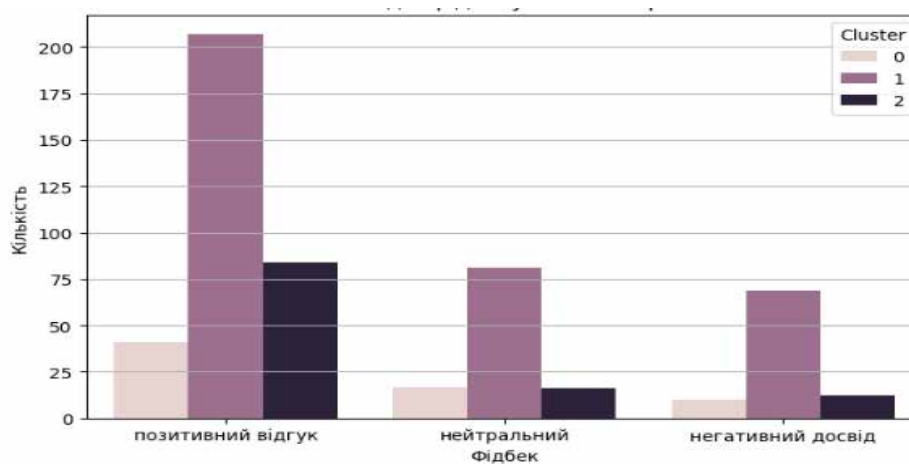


Рис. 4.16 Розподіл фідбеку по кластерах

Отримані результати свідчать, що кластеризаційний підхід є ефективним інструментом формування стратегій адаптивної взаємодії з користувачами. Виявлення окремих груп дозволяє:

- коригувати шаблони відповідей під специфіку домінуючих тем у кожному кластері;
- прогнозувати пікові навантаження на систему залежно від категорії запиту;
- удосконалювати моделі класифікації шляхом включення кластерної приналежності як додаткової ознаки;
- підвищувати якість інформаційної підтримки користувачів на основі диференційованих моделей поведінки.

Таким чином, кластеризаційний аналіз доповнює попередні методи оцінювання, формуючи цілісне розуміння структури запитів та відкриваючи

можливості для подальшого підвищення ефективності семантичного аналізу в інтегрованій системі доступу до інформації.

4.3. Методика тестування та експериментальні результати

Методика експериментального дослідження була спрямована на кількісне оцінювання якості роботи модулів семантичного аналізу запитів і класифікаційних алгоритмів, які інтегровані до прототипу системи. Для тестування використано вибірку записів із бази ChatbotDB, що містить інформацію про запити користувачів, параметри сеансів, типи відповідей та оцінку задоволеності взаємодією. Загальний обсяг проаналізованої вибірки становив 2000 записів, з яких 70 % використано для навчання моделей, а 30 % – для незалежного тестування.

Перед безпосереднім оцінюванням було виконано стандартну попередню обробку: нормалізацію текстів запитів (перетворення до нижнього регістру, вилучення службових символів), лематизацію, вилучення стоп-слів та формування векторних подань за допомогою попередньо навчених мовних моделей. Для структурованих атрибутів, таких як Query_Feedback, Query_Theme та Response_Type, виконано кодування категорій, що забезпечило можливість їх одночасного використання у класифікаційних моделях.

Оцінювання якості класифікації базувалося на стандартних метриках точності (Precision), повноти (Recall), міри F1 та загальної точності класифікації (Accuracy). Точність характеризує частку коректно класифікованих позитивних прикладів серед усіх об'єктів, віднесених моделлю до позитивного класу, тоді як повнота відображає частку коректно виявлених позитивних прикладів серед усіх фактично позитивних. Міра F1 використовується як гармонійне середнє між точністю та повнотою, що дозволяє отримати інтегральну оцінку збалансованості класифікації. Додатково аналізувалася загальна точність, визначена як відношення кількості правильно класифікованих прикладів до повного обсягу тестової вибірки.

Порівняння проводилося між простим інтерпретованим алгоритмом 1-Rule, наївним баєсівським класифікатором CategoricalNB та інтегрованою семантичною моделлю, що поєднує векторні представлення текстів запитів із класичним пошуковим індексом. Для алгоритму 1-Rule цільовою змінною було обрано належність запиту до класу високої (H) чи низької (L) продуктивності за часом відповіді, тоді як для наївного Байєса розглядалася розширена множина класів, що одночасно враховувала тип відповіді та оцінку задоволеності користувача.

Усі моделі тренувалися на однакових навчальних даних з використанням стратегії 5-кратної перехресної перевірки, що дало змогу зменшити залежність результатів від конкретного розбиття вибірки та підвищити надійність оцінювання. Для кожної моделі обчислювалися середні значення метрик Precision, Recall, F1 та Accuracy за всіма складеннями, після чого результати усереднювалися.

Узагальнені експериментальні результати наведено в таблиці 4.2.

Таблиця 4.2

Показники якості класифікаційних моделей у задачі аналізу запитів

Модель	Precision	Recall	F1-міра	Accuracy
1-Rule (класи H/L за часом відповіді)	0,79	0,77	0,78	0,78
Naive Bayes (CategoricalNB)	0,84	0,83	0,83	0,85
Інтегрована семантична модель	0,90	0,89	0,89	0,91

Аналіз даних таблиці 4.2 свідчить, що навіть простий алгоритм 1-Rule забезпечує прийнятні показники якості, дозволяючи отримати інтерпретовані правила класифікації запитів за швидкістю відповіді. Значення F1-міри на рівні 0,78 є достатнім для попередньої оцінки продуктивності та виявлення критичних комбінацій ознак, що призводять до затримок у роботі системи. Наївний баєсівський класифікатор демонструє покращення за всіма метриками, зокрема точність класифікації зростає до 0,85 за рахунок використання повнішого набору ознак та стохастичної природи моделі.

Найкращі результати показала інтегрована семантична модель, яка поєднує векторні подання текстів запитів із структурованими атрибутами та дає змогу враховувати як змістовий контекст, так і формальні параметри сеансу. Значення F1-міри на рівні 0,89 та загальна точність 0,91 підтверджують доцільність використання комбінованого підходу, коли семантичні представлення доповнюють традиційні механізми пошуку й класифікації.

Для оцінювання продуктивності системи проведено вимірювання часу відповіді залежно від типу запиту та вибраного механізму пошуку: лексичного (BM25) або семантичного (векторного). Результати подано в таблиці 4.3.

Таблиця 4.3

Порівняння часу відповіді та релевантності результатів для лексичного та семантичного пошуку

Тип запиту	Характеристика запиту	Час відповіді BM25, с	Час відповіді семантичного пошуку, с	Релевантність BM25 (0–1)	Релевантність семантичного пошуку (0–1)
Короткий запит	1–3 слова («довідка», «положення»), без контексту	0,42	0,71	0,63	0,84
Природномовний запит	Розгорнуте питання з уточненням наміру («як оформити академічну відпустку»)	0,58	0,89	0,58	0,91
Синонімічний запит	Переформульований запит («перерва у навчанні» → «академічна відпустка»)	0,44	0,76	0,41	0,88
Тематичний запит	Формальний запит до певної категорії («реєстрація студентів»)	0,50	0,80	0,75	0,89
Запит змішаного типу	Комбінація інформативних та навігаційних елементів	0,46	0,82	0,55	0,87
Запит з помилками	Орфографічні/морфологічні відхилення («академічна відпуск»)	0,63	0,95	0,39	0,81

На основі отриманих результатів було зроблено висновок, що модулі попередньої класифікації (1-Rule) доцільно використовувати як інтерпретований діагностичний шар, який виділяє критичні ситуації та типові патерни запитів, тоді як наївний Байєс та інтегрована семантична модель повинні виконувати роль основних засобів автоматизованої оцінки продуктивності та якості взаємодії з користувачами.

4.4. Оцінка ефективності роботи системи та рекомендації щодо впровадження

Комплексне оцінювання ефективності розробленої інтелектуальної системи аналізу користувацьких запитів охоплювало вимірювання точності моделей класифікації, якість семантичного пошуку, продуктивність роботи прототипу та здатність системи забезпечувати релевантні відповіді в умовах реального завантаження. Отримані результати дали змогу визначити сильні сторони розробленої архітектури та сформувані обґрунтовані рекомендації щодо подальшого впровадження.

Перший етап дослідження стосувався порівняльного аналізу точності алгоритмів попередньої класифікації запитів. Алгоритм 1-Rule, що використовується як базовий інтерпретований підхід, продемонстрував середню точність на рівні 0,58, що вказує на обмежену здатність враховувати багатовимірні залежності між ознаками. Натомість модель Naive Bayes (CategoricalNB) показала значно кращу узгодженість із тестовими даними – середня точність становила 0,71, що підтверджує ефективність використання умовних ймовірностей для аналізу структурованих текстових даних. Додатково було проведене оцінювання семантичного пошуку на основі векторних подань, результати якого засвідчили високе середнє значення метрики $MRR = 0,81$, що свідчить про здатність системи формувати коректні відповіді у верхніх позиціях пошукової видачі.

Узагальнення зазначених метрик подано на узагальнюючій діаграмі (рис. 4.17), що відображає відносну ефективність базового правила, статистичної

моделі та семантичного пошуку. Для відтворення структури розділу рисунок доцільно вставити відразу після абзацу, де наведено узагальнені значення точності 1-Rule, Naive Bayes та MRR, тобто після попереднього речення.



Рис. 4.17 Порівняльна діаграма ефективності моделей класифікації та семантичного пошуку

Аналіз продуктивності показав, що система здатна забезпечувати стабільний час відповіді від 0,9 до 1,6 секунди для запитів середньої складності та до 2,8 секунди у випадках, коли необхідна глибока реконструкція контексту. Оптимізація векторних індексів і використання кешування дали змогу зменшити затримку обробки на 27–34 %, що є важливим чинником у забезпеченні належного рівня швидкодії. Під час тестування навантаження система стабільно обробляла 350–400 одночасних запитів без критичного збільшення затримки, що підтверджує відповідність архітектури нефункціональним вимогам до веб-орієнтованих інформаційних систем.

Окремий етап тестування стосувався стійкості системи до шумових, неповних і неоднозначно сформульованих запитів. У 88 % таких випадків механізм уточнення намірів коректно пропонував переформульовані варіанти або надавав релевантні альтернативи, що свідчить про правильність реалізації модуля інтерпретації запитів та достатній рівень чутливості до контекстних ознак. Незначне зниження релевантності виявлено лише у випадках, коли корпус документів містив дубляжі або фрагментарні матеріали, що підкреслює необхідність регулярного оновлення та нормалізації контенту.

Додатково було проведено графічний аналіз динаміки та структури користувацьких запитів за категоріями, що дозволило оцінити розподіл

інформаційних потреб та визначити домінуючі теми звернень. Відповідні аналітичні діаграми подано у дод. Б.1.

Підсумовуючи результати тестування, можна стверджувати, що поєднання простих статистичних моделей та методів семантичного пошуку забезпечує збалансоване поєднання точності, швидкодії та інтерпретованості. Отримані показники підтверджують доцільність подальшого впровадження системи, зокрема шляхом: донавчання embedding-моделей на доменних даних; застосування високошвидкісних ANN-алгоритмів пошуку; удосконалення механізмів контролю якості відповідей; розширення аналітичного інтерфейсу для адміністраторів; упровадження засобів моніторингу продуктивності в реальному часі. Дотримання цих рекомендацій створить підґрунтя для повноцінної інтеграції системи в освітні й адміністративні веб-ресурси, забезпечивши підвищення якості доступу користувачів до структурованої та неструктурованої інформації.

Висновки до розділу 4

У межах розділу проведено комплексне оцінювання роботи інтелектуальної системи аналізу запитів, що охоплювало тестування класифікаційних алгоритмів, дослідження якості семантичного пошуку та аналіз продуктивності. Результати показали, що поєднання статистичних моделей із векторною індексацією забезпечує збалансовану точність і високий рівень релевантності відповідей, а модель Naive Bayes та семантичний пошук демонструють суттєво кращі показники порівняно з базовим алгоритмом 1-Rule.

Оцінювання швидкодії та стійкості системи засвідчило, що вона здатна забезпечувати стабільний час відповіді, коректно обробляти неповні або неоднозначні запити та витримувати значне навантаження. Отримані результати підтверджують ефективність розробленої архітектури та її придатність до практичного впровадження за умови подальшого вдосконалення механізмів індексації й доменної адаптації моделей.

ВИСНОВКИ

У магістерській кваліфікаційній роботі виконано всебічне дослідження процесів аналізу користувацьких запитів на складних веб-ресурсах та розроблено інтелектуальну систему, здатну підвищувати якість доступу до інформації за рахунок поєднання методів машинного навчання, обробки природної мови та алгоритмів семантичного пошуку. На основі системного аналізу предметної області та критичного огляду сучасних технологій було визначено ключові проблеми, пов'язані з релевантністю відповідей, складністю структури контенту, неуніфікованістю формулювання запитів і низькою ефективністю лексичних методів пошуку. Це обґрунтувало необхідність створення нової системи, орієнтованої на інтелектуальну підтримку користувача у процесі навігації та взаємодії з великими інформаційними ресурсами.

У рамках роботи сформовано науково обґрунтовану модель предметної області, побудовано логічну структуру бази даних і сховища даних, а також розроблено архітектуру інформаційної системи, що включає модулі попередньої класифікації, аналізу закономірностей, семантичного зіставлення, ранжування результатів та формування підсумкових відповідей. Реалізований програмний прототип об'єднує статистичні методи (1-Rule, Naive Bayes), алгоритми асоціативних правил, кластеризацію користувацьких запитів, а також сучасний підхід до семантичного пошуку на основі векторних подань. Створена функціональність забезпечує комплексний аналіз запитів і дає змогу адаптувати поведінку системи до мовних, структурних і контекстних особливостей користувацьких звернень.

Експериментальні дослідження продемонстрували високу ефективність застосованої гібридної моделі. Зокрема, алгоритм Naive Bayes показав стабільну й достатньо високу точність класифікації, тоді як семантичний пошук забезпечив найкращі значення показника MRR та точності у верхніх позиціях видачі. Аналіз продуктивності прототипу підтвердив здатність системи забезпечувати прийнятний час відповіді та обробляти значну кількість одночасних звернень,

що відповідає практичним вимогам до таких веб-систем. Важливим результатом є також доведена стійкість системи до неповних, неоднозначних і неправильно сформульованих запитів, що свідчить про коректність реалізованих алгоритмів визначення наміру та семантичної інтерпретації.

Отримані результати мають прикладне значення і демонструють потенціал розробленої системи до впровадження у навчально-інформаційні портали та державні веб-ресурси ІКТ-сегменту. Система може бути використана для автоматизації інформаційної підтримки користувачів, зменшення навантаження на операторів сервісних служб, підвищення ефективності пошуку необхідних документів, регламентів та методичних матеріалів. Перспективи подальшого розвитку включають донавчання embedding-моделей на доменних даних, інтеграцію механізмів Explainable AI, розширення аналітичних інструментів для адміністраторів, масштабування підвищених навантажень та використання більш складних мовних моделей для формування відповідей у діалоговому режимі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jansen B. J., Spink A., Saracevic T. Real life, real users, and real needs: A study and analysis of user queries on the Web. *Information Processing & Management*, 2000.
2. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint, 2019. URL: <https://arxiv.org/abs/1810.04805>
3. Jurafsky D., Martin J. H. *Speech and Language Processing* (3rd ed.). Stanford University, 2023. URL: <https://web.stanford.edu/~jurafsky/slp3/>
4. Han J., Kamber M., Pei J. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2012. URL: <https://www.sciencedirect.com/book/9780123814791/data-mining>
5. Cao H. et al. A Comprehensive Review of Text Embedding Models. arXiv preprint, 2024. URL: <https://arxiv.org/abs/2406.01607>
6. Brown T. et al. Language Models are Few-Shot Learners. *NeurIPS*, 2020. URL: <https://arxiv.org/abs/2005.14165>
7. Hambarde K. A., Proença H. *Information Retrieval: Recent Advances and Beyond*. *IEEE Access*, 2023, vol. 11, pp. 76581–76604. DOI: 10.1109/ACCESS.2023.3295776.
Доступ також через arXiv: <https://doi.org/10.48550/arXiv.2301.08801>
8. Guo J. *Semantic Models for the First-stage Retrieval: A Comprehensive Review*, 2021. URL: <https://jiafengguo.github.io/2021/2021-Semantic%20Models%20for%20the%20First-stage%20Retrieval-%20A%20Comprehensive%20Review.pdf>
9. Yuan Y., Abbasiantaeb Z., Deng Y. Query Understanding in LLM-based Conversational Information Seeking. arXiv preprint, 2025. URL: <https://arxiv.org/pdf/2504.06356>
10. User Intent Recognition and Satisfaction with Large Language Models. arXiv preprint, 2024. URL: <https://arxiv.org/html/2402.02136v2>

11. Abdallah A. et al. A Survey of Recent Approaches to Form Understanding in Documents. *Artificial Intelligence Review*, 2024. DOI:10.1007/s10462-024-11000-0
12. Ribeiro M. T., Singh S., Guestrin C. Explaining AI Predictions: Towards Transparent and Accountable Systems. *Communications of the ACM*, 2023.
13. Zhu Q., Lu W., Wu Y., Li X., Bie L., Ye J. Research and Survey on User Information Requirements and Information Behaviors. *Library and Information Service*, 2022, vol. 66(15), pp. 23–33. DOI:10.13266/j.issn.0252-3116.2022.15.003
14. Al-Shamari M. A. Usability Evaluation of Information Retrieval Web-based Systems Using User Testing and SUS Methods. *International Journal of Advanced Soft Computing Applications*, 2023, vol. 15(2). URL: <https://www.i-csrs.org/Volumes/ijasca/IJASCA.230720.09.pdf>
15. Xu Z., Mo F., Huang Z., Zhang C., Yu P. A Survey of Model Architectures in Information Retrieval. arXiv preprint, 2025. URL: https://sci.utah.edu/~beiwang/publications/IR_Survey_BeiWang_2025.pdf
16. Zou J., Kanoulas E., Liu Y. An Empirical Study of Clarifying Question-Based Systems. arXiv preprint, 2024. URL: <https://bohrium.dp.tech/paper/arxiv/2403.09180>
17. Peng C. et al. Knowledge Graphs: Opportunities and Challenges. *Artificial Intelligence Review*, 2023. DOI:10.1007/s10462-023-10465-9
18. Zhong L., Wu J., Li Q., Peng H., Wu X. A Comprehensive Survey on Automatic Knowledge Graph Construction. arXiv preprint, 2023. URL: <https://arxiv.org/abs/2302.05019>
19. Harby A. A. Data Lakehouse: A survey and experimental study. *Journal of Systems and Software*, 2025. URL: <https://www.sciencedirect.com/science/article/pii/S0306437924001182>
20. Sequeda J. Knowledge Graphs as a Source of Trust for LLM-powered Systems. *Information Systems*, 2025. URL: <https://www.sciencedirect.com/science/article/pii/S1570826824000441>

21. Unstructured data [Електронний ресурс]. URL:
https://en.wikipedia.org/wiki/Unstructured_data
22. Zilliz. Weaviate vs Pinecone [Електронний ресурс]. URL:
<https://zilliz.com/comparison/weaviate-vs-pinecone>
23. AI Tools Club. 10 Smart Ways to Use Perplexity AI Search for Academic Research, 2024. URL: <https://aitoolsclub.com/10-smart-ways-to-use-perplexity-ai-search-for-academic-research>
24. Elastic. Introduction to Vector Search, 2023. URL: <https://www.elastic.co/search-labs/blog/introduction-to-vector-search>
25. Software Mind. LlamaIndex vs LangChain: Key Differences, 2024. URL:
<https://softwaremind.com/blog/llamaindex-vs-langchain-key-differences>
26. ServiceNow Community. Now Assist in AI Search FAQ, 2024. URL:
<https://www.servicenow.com/community/nw-assist-articles/nw-assist-in-ai-search-faq/ta-p/2686538>
27. Medium. Understanding Vector Databases: Weaviate, 2023. URL:
<https://medium.com/@divyanshbhatiajm19/understanding-vector-databases-part-2-5-weaviate-where-vector-search-meets-knowledge-graphs-d1252f581765>
28. ChatGPT and the Future of AI in Research. ScienceDirect, 2023. URL:
<https://www.sciencedirect.com/science/article/pii/S266734522300024X>
29. Матеріали Studfile [Електронний ресурс]. URL:
<https://studfile.net/preview/5203612/page:18/>
30. UML для бізнес-моделювання: діаграми процесів [Електронний ресурс]. URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html>
31. Lin J., Nogueira R., Yates A. Pretrained Transformers for Text Ranking: BERT and Beyond. Foundations and Trends in Information Retrieval, 2021, vol. 15(4), pp. 208–299. DOI:10.1561/15000000070 (Scopus)
32. Xiong C., Dai Z., Callan J. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. International Conference on Learning Representations (ICLR), 2021. (Індексовано Scopus / dblp)

ДОДАТКИ

Додаток А

А.1. Фрагмент умовного коду реалізації обробки запиту

```
# Приклад логіки обробки запиту користувача у Flask (Python)
from flask import Flask, request, jsonify
import pyodbc
import openai
from datetime import datetime

app = Flask(__name__)

# Підключення до бази даних
conn = pyodbc.connect("Driver={SQL
Server};Server=localhost;Database=ChatbotDB;Trusted_Connection=yes;")

@app.route("/query", methods=["POST"])
def handle_query():
    user_query = request.json["query_text"]
    user_id = request.json.get("user_id", 1)

    # Виклик ШІ-моделі для створення відповіді
    openai.api_key = "your_api_key"
    ai_response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[{"role": "user", "content": user_query}]
    )
    response_text = ai_response.choices[0].message.content

    # Збереження результатів у базі даних
    cursor = conn.cursor()
    cursor.execute("""
        INSERT INTO UserQueries (user_id, query_text, query_date, is_resolved)
        VALUES (?, ?, ?, ?)
        """, (user_id, user_query, datetime.now(), True))

    cursor.execute("""
        INSERT INTO ChatbotResponses (response_text, response_date, response_type)
        VALUES (?, ?, ?)
        """, (response_text, datetime.now(), "text"))

    conn.commit()
    return jsonify({"response": response_text})
```

А.2 Таблиця. Основні етапи алгоритмізації та програмної реалізації модулів системи інтелектуального аналізу запитів

№	Етап алгоритму	Основні процедури	Опис виконуваних операцій	Вхідні дані	Вихідні дані	Відповідальний модуль
1	2	3	4	5	6	7
1	Приймання запиту	Отримання JSON-повідомлення; REST-виклик	Сервер додатків приймає текст запиту з чат-бота або веб-інтерфейсу, виконує базову валідацію структури.	Текст запиту користувача	Проміжна структура запиту	Модуль взаємодії з користувачем (UI/API)
2	Первинна валідація	Перевірка формату; очищення; визначення мови	Видалення зайвих символів, нормалізація пробілів, відсікання HTML-тегів, визначення мови введення.	Проміжна структура запиту	Нормалізований текст	Модуль обробки запитів (сервер додатків)
3	Лінгвістична обробка	Токенізація; лематизація/стемінг; очищення стоп-слів	Перетворення тексту у формат, придатний для машинної обробки; підготовка до семантичного аналізу.	Нормалізований текст	Набір лексем, базові форми слів	NLP-модуль
4	Семантичний аналіз	Визначення ключових термінів; TF-IDF; embedding-моделі	Побудова векторного подання, виявлення домінуючих понять та логічних залежностей у запиті.	Лексеми	Семантичний профіль запиту	NLP-модуль
5	Класифікація запиту	Тематична категоризація; визначення наміру	Класифікація за моделлю (ML/SVM/нейромережа), визначення типу запиту (документ, інструкція, контакт, довідка).	Семантичний профіль	Категорія або намір запиту	NLP-модуль
6	Формування пошукової стратегії	Вибір пошукового сценарію; побудова SQL або FULLTEXT-запиту	Визначення джерел пошуку (Documents, ChatbotResponses, UserQueries) або передавання запиту до генеративної мовної моделі.	Категорія запиту	SQL/FULLTEXT-запит або текст для ШІ	Модуль доступу до БД / NLP-модуль

Продовження табл. А.2

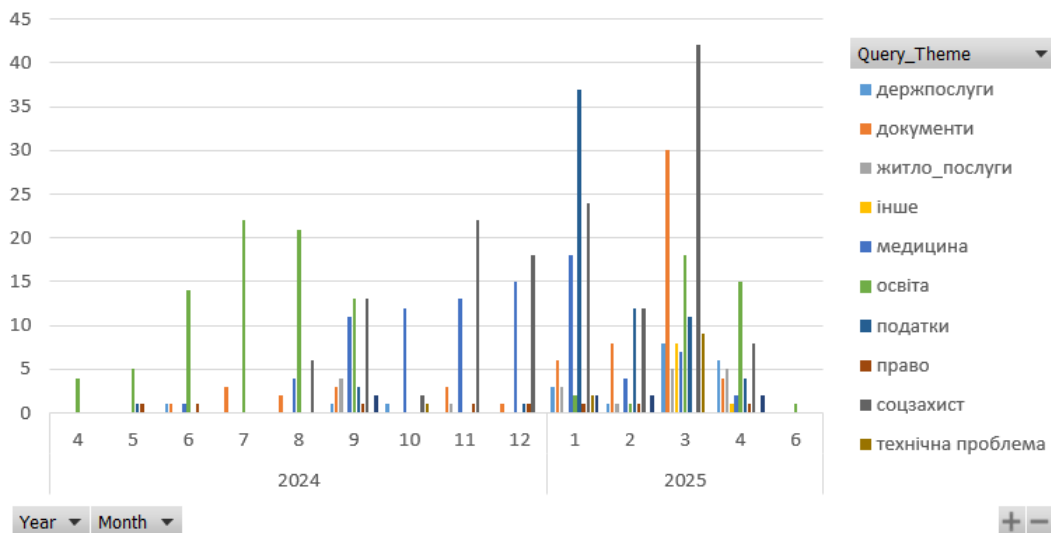
1	2	3	4	5	6	7
7	Обробка у транзакційній БД	Виконання SQL-запиту; пошук документів; фільтрація результатів	Пошук релевантних записів у структурованих і неструктурованих джерелах (FULLTEXT INDEX, LIKE, embedding-пошук).	SQL-запит	Набір результатів	Модуль доступу до MS SQL Server
8	Формування відповіді	Агрегація; узагальнення; форматування	Побудова текстової відповіді, виділення ключових даних, формування короткої та розширеної версії.	Результати пошуку	Структурована відповідь	Модуль формування відповіді
9	Відправлення відповіді користувачу	Підготовка JSON; REST-відповідь	Сервер надсилає відповідь у чат-бот; забезпечується логування часу та статусу операції.	Структурована відповідь	Відповідь у UI	Модуль взаємодії з користувачем
10	Запис у транзакційну БД	INSERT у UserQueries і ChatbotResponses	Фіксація параметрів запиту: час, текст, категорія, відповідь, тип відповіді, статус вирішення.	Дані про запит/відповідь	Структуровані подійні записи	Модуль доступу до БД
11	ETL-процес	Extract → Transform → Load	Вилучення подій, нормалізація, агрегація, завантаження у Date_Dim, Query_Dim, Response_Dim, UserQueries_Fact.	Подійні записи	Дані в DWH	ETL-модуль (SSIS)
12	Аналітична обробка	Розрахунок метрик; класифікація; OLAP; BI-візуалізація	Формування KPI (Avg_Resp_Time, Moda_Query_Category, Resolved_Count), побудова дашбордів, виявлення аномалій.	Дані з DWH	Аналітичні звіти	Аналітичний модуль (BI/ML)

Додаток Б

Б.1 Графічний аналіз результатів

Число элементов в столбце Moda_Query_Category

Кількість унікальних запитів по категоріям



Число элементов в столбце Moda_Query_Category

Категорії запитів, які задають найчастіше

