ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНІКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «Магістр»

«Дослідження та оптимізація роботи пошукової системи бібліотек з елементами машинного навчання»

(тема кваліфікаційної роботи українською мовою)

«Research and optimization of the search system of libraries with elements of machine learning»

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач заочної форми навчання

спеціальності 122 Комп'ютерні науки

(код, назва спеціальності)

Освітня програма Комп'ютерні науки

(назва)

АЗІЗ Заур Вагіф огли

(прізвище, ім'я, по-батькові здобувача)

Керівник к.т.н., доцент Терещенко Т.М. _____

(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент к.т.н., доцент кафедри інженерії ПЗ національного університету "Одеська політехніка,Тройніна А.С.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:
Протокол засідання кафедри
Інформаційних технологій
№ ____ від ____ _____ 2024 р.

Завідувачка кафедри

_____ _____
(підпис) (прізвище,ім'я)

Захищено на засіданні ЕК № _____
протокол № ___ від _____ 2024 р.

Оцінка _____ / _____ / _____
(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

_____ _____
(підпис) (прізвище,ім'я)

Одеса 2024

АНОТАЦІЯ

Метою роботи є розробка сучасної пошукової системи для електронних бібліотек, яка забезпечує точний та персоналізований пошук за допомогою технологій обробки природної мови (NLP), автоматизує бібліотечні процеси та сприяє зручному доступу до інформації.

Методи розробки базуються на моделі обробки тексту BERT, мові програмування Python та її бібліотеках spaCy, Sentence-BERT, Elastic-Search для NLP.

Результатом роботи є пошукова система для електронної бібліотеки із підтримкою NLP.

Ключові слова: BERT, Elastic-Search, Natural Language Processing, Python, spaCy, Sentence-BERT, електрона бібліотека, пошукова система.

ABSTRACT

The purpose of the work is to develop a modern search system for electronic libraries, which provides accurate and personalized search using natural language processing (NLP) technologies, automates library processes and facilitates convenient access to information.

Development methods are based on BERT text processing model, Python programming language and its libraries spaCy, Sentence-BERT, Elastic-Search for NLP.

The result of the work is a search system for an electronic library with NLP support.

Keywords: BERT, Elastic-Search, Natural Language Processing, Python, spaCy, Sentence-BERT, electronic library, search engine.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ

Скорочення_м

BERT – Bidirectional Encoder Representations from Transformers

CBOW – Continuous Bag of Words

CRF – Conditional Random Fields

ESP – Evidence-based Selection Planning)

GPT – Generative Pre-trained Transformer

HMM – Hidden Markov Models

NER – Named Entity Recognition

NLP – Natural Language Processing

SVM – Support Vector Machines

ЕБ – Електрона Бібліотека

ВСТУП

Тривалий час основним засобом зберігання інформації залишалися паперові носії (документи, книги тощо), доступ до яких надавали бібліотеки. Однак із розвитком інформаційних технологій та їх інтеграцією в різні сфери життя процеси організації та функціонування інформаційних сховищ стали більш ефективними. Появився термін «електронна бібліотека», що об’єднує різні підходи до створення таких систем. Замість надання доступу до паперових матеріалів бібліотеки тепер забезпечують доступ до електронної інформації, яка легко тиражується, доступна у глобальних комп'ютерних мережах, незалежно від часу та місця звернення користувачів. В останні роки технології стали важливими не лише для управління бібліотечними колекціями, але й для самої структури бібліотеки. У багатьох випадках бібліотеки є однією з перших точок контакту багатьох клієнтів з інноваціями.

Останні роки спостерігається зростання цифрових ресурсів бібліотеках. Під час опитування понад 1500 бібліотек у США Асоціація публічних бібліотек (PLA) виявила, що 95% бібліотек зараз пропонують своїм відвідувачам електронні книги. Однак у бібліотеках нова тенденція полягає в тому, що цифрові колекції розширюються далеко за межі книг, включаючи: аудіокниги, журнали, газети, комікси.

Є побоювання, що перехід до цифрових колекцій може зменшити використання фізичних бібліотечних просторів. Однак доступність матеріалів онлайн покращує доступність колекції для всіх відвідувачів [1].

Бібліотеки можуть використовувати штучний інтелект, щоб передбачати не лише майбутні потреби та вподобання, але й рекомендувати особисті пропозиції щодо матеріалів для читання окремим користувачам на основі їхніх уподобань, історії запозичень, попереднього перегляду та інтересів.

ШІ може відігравати вирішальну роль, допомагаючи бібліотекарям приймати ключові рішення, які впливатимуть на майбутнє бібліотеки.

Аналізуючи дані про моделі використання, бібліотекари можуть збирати інформацію, яка допоможе в плануванні заходів, розподілі бюджету, розвитку послуг і створенні колекцій, які залучатимуть користувачів бібліотеки.

Оскільки ця технологія продовжує набувати все більшого поширення, бібліотекари також повинні навчати користувачів бібліотек штучному інтелекту та тому, як він може покращити їхнє повсякденне життя.

Метою роботи є розробка сучасної пошукової системи для електронних бібліотек, яка забезпечує точний та персоналізований пошук за допомогою технологій обробки природної мови (NLP), автоматизує бібліотечні процеси та сприяє зручному доступу до інформації.

Об’єкт дослідження – сукупність процесів організації та функціонування пошукових систем в електронних бібліотеках для забезпечення доступу до інформаційних ресурсів.

Предмет дослідження – методи та інструменти обробки природної мови (NLP) і пошукові технології для підвищення ефективності роботи пошукових систем електронних бібліотек.

1 СИСТЕМНИЙ АНАЛІЗ ТА ОГЛЯД АНАЛОГІВ

1.1 Характеристика предметної області

Предметна область – бібліотечна інформаційна система, зокрема, процеси пошуку, збереження та надання доступу до літератури. Із зростанням обсягів цифрових бібліотечних ресурсів та можливостей для дистанційного доступу, бібліотеки активно вдосконалюють свої інформаційні системи для забезпечення зручного і точного пошуку літератури.

ШІ більше не є футуристичною концепцією; це практичний інструмент, який бібліотеки використовують, щоб революціонізувати свої послуги. Шляхом автоматизації каталогізації та індексування ШІ забезпечує більш ефективну організацію та пошук інформації, роблячи бібліотечні колекції більш доступними для пошуку та зручнішими для користувачів. Крім того, послуги мовного перекладу, керовані штучним інтелектом, відкрили бібліотеки для тих, хто не розмовляє англійською мовою, розширивши базу їх користувачів. Алгоритми персоналізації можуть запропонувати ресурси, адаптовані до індивідуальних інтересів, тоді як технології штучного інтелекту допомагають оцифровувати, зберігати та робити доступними рідкісні та історичні документи, які інакше можуть зіпсуватися з часом. Ці програми штучного інтелекту не тільки покращують досвід користувачів, але й зберігають багатство знань, що зберігаються в бібліотеках [2].

Особлива увага приділяється інтеграції елементів машинного навчання, що дозволяє створювати інтелектуальні пошукові системи. Такі системи використовують обробку природної мови (NLP) для розуміння користувацьких запитів, рекомендаційні алгоритми для підбору відповідної літератури та системи класифікації для аналізу контенту книг.

Основні завдання предметної області:

1. Надання користувачам можливості зручного пошуку та отримання інформації про літературу, враховуючи їхні потреби та інтереси.

2. Автоматизація та оптимізація процесу пошуку, що забезпечує швидкий та точний доступ до джерел, які можуть бути корисними для навчання, дослідження та розвитку.
3. Застосування аналітичних інструментів, що дозволяють бібліотекам вивчати інтереси користувачів та на основі цього налаштовувати рекомендації та вдосконалювати бібліотечний фонд.

Бібліотечна інформаційна система з елементами машинного навчання надає зручний інтерфейс, через який користувачі можуть шукати літературу за конкретними темами, запитами або загальними інтересами. Вона забезпечує обробку тексту, включаючи виділення ключових слів, семантичний аналіз, розпізнавання синонімів, контексту та інших важливих параметрів, що впливають на релевантність пошукових результатів.

Система також займається персоналізованим підбором літератури, використовуючи історію запитів і поведінкові дані для покращення користувацького досвіду.

Чат-боти зі штучним інтелектом надають чудову можливість звільнити час бібліотекарів для виконання більш важливих галузевих завдань. Їх можна налаштувати як віртуальний довідник для бібліотечних послуг , відповідаючи на поширені запитання та надаючи інформацію відвідувачам у неробочий час. Вони можуть навіть вирішувати технічні проблеми, наприклад проблеми з доступом до електронних ресурсів.

Чат-боти зі штучним інтелектом також надають відвідувачам інші переваги та можуть бути чудовим інструментом для покращення доступності бібліотечних послуг . Вони допомагають користувачам орієнтуватися в книгах і журналах і навіть запропонують різні бази даних, якщо там найімовірніше знайти матеріали [1].

Деякі бібліотеки використовують технологію штучного інтелекту для підготовки списків рекомендованих матеріалів для відвідувачів на основі їхньої попередньої взаємодії з колекціями. В академічних бібліотеках віртуальні асистенти-дослідники використовуються з великою ефективністю,

коли вони навчаються за допомогою рецензованих статей, щоб допомогти дослідникам знаходити відповідні статті, оцінювати їх вміст і генерувати ідеї та резюме.

ШІ став невід'ємною частиною багатьох щоденних операційних завдань у бібліотеках. Одним із прикладів є розвиток колекції. ESP (Evidence-based Selection Planning) включає складне машинне навчання, щоб використовувати аналіз поведінки та дані для допомогти бібліотекарям приймати більш обґрунтовані рішення щодо вибору.

1.2 Актуальність дослідження та розробки пошукової системи

Трансформація бібліотек від їх традиційних, фізичних втілень до цифрових форм являє собою глибоку зміну способів зберігання, доступу та поширення знань та інформації. Ця метаморфоза не відбулася миттєво, а була ретельною та цілеспрямованою еволюцією. Спочатку це почалося з оцифрування систем бібліотечних каталогів, що стало величезним кроком у відхід від ящиків карткових каталогів, які вимагали фізичного перегляду. Ця початкова фаза заклала основу для ширшої цифрової революції в бібліотеках, проклавши шлях до впровадження електронних книг, онлайнових баз даних, наукових статей і мультимедійних ресурсів, доступних через веб-сайти бібліотек і цифрові платформи.

Каталізатором цієї кардинальної зміни стала поява та поширення цифрових технологій, які стали потужним вирівнювачем доступу до інформації. До цифрової ери доступ до інформації часто визначався близькістю людини до фізичних бібліотек і ресурсів, які в них зберігалися. Студенти, дослідники та широка громадськість змушені були фізично відвідувати бібліотеку, щоб отримати доступ до книг, журналів чи архівів. Однак із цифровими бібліотеками географічні та соціально-економічні бар'єри, які колись обмежували доступ, значно зникли. Тепер кожен, хто має доступ до Інтернету, може отримати доступ до величезних сховищ знань

цифрових бібліотек, які містять мільйони цифрових книг, статей та інших ресурсів [2].

Електронні бібліотеки відкривають широкий спектр нових можливостей для всіх, хто взаємодіє з ними. Для бібліотекарів це – шанс модернізувати автоматизовані бібліотечні системи, трансформуючи їх у публічні електронні бібліотеки нового покоління, що надають доступ до різноманітних цифрових ресурсів. Такі системи пропонують не лише вдосконалені інструменти для пошуку й доступу до інформації, але й враховують інтеграцію бібліотечних і видавничих технологій, що робить їх більш функціональними і зручними.

Освітяни різних рівнів, інтегруючи електронні бібліотеки у навчальний процес, значно підвищують якість освіти, створюючи нову інформаційну інфраструктуру для навчання. Електронні бібліотеки допомагають надавати учням і студентам доступ до актуальних джерел знань, що сприяє більш глибокому розумінню та засвоєнню матеріалу.

Актуальність дослідження проблеми та розробки пошукової системи бібліотек з елементами машинного навчання:

1. Ріст цифрових ресурсів.

У зв’язку зі значним збільшенням обсягів інформації, яка доступна в цифровій формі, пошук релевантної літератури стає все більш складним. Користувачі часто не можуть знайти потрібні матеріали через невідповідність пошукових запитів та бібліотечних записів.

2. Покращення користувацького досвіду.

Завдяки обробці природної мови, система здатна більш точно розпізнавати інтенцію користувача, що робить процес пошуку швидшим і приємнішим.

3. Персоналізовані рекомендації.

Інтелектуальна система дозволяє створювати персоналізовані рекомендації, що сприяє розширенню кругозору користувачів, підвищенню їхнього інтересу до нових тем та кращому задоволенню їхніх освітніх потреб.

4. Оптимізація роботи бібліотек.

Завдяки аналізу запитів та інтересів користувачів, бібліотеки можуть оптимізувати структуру своїх фондів і формувати пропозиції для покращення наявних ресурсів.

5. Стимулювання досліджень.

Розвиток системи з елементами машинного навчання сприяє новим дослідженням у сфері семантичного пошуку та застосування штучного інтелекту для роботи з великими обсягами текстових даних. Таким чином, розробка інформаційної системи для бібліотек із елементами машинного навчання не тільки забезпечує оптимізацію доступу до знань, але й відкриває нові можливості для розвитку та дослідження інформаційних технологій [3].

1.3 Структура електронної бібліотеки

Структура електронної бібліотеки включає декілька типів інформаційних систем, які працюють разом для забезпечення повноцінного функціонування бібліотеки. Основні структурні елементи електронної бібліотеки (ЕБ) можна умовно поділити на три взаємопов'язані компоненти, які формують єдину систему:

1. Контентна складова.

Вона містить усі цифрові ресурси бібліотеки, такі як книги, статті, технічні звіти, мультимедійні матеріали, архівні документи та інші інформаційні джерела.

До контенту також належать метадані (описові дані про ресурси), що полегшують пошук, категоризацію та ідентифікацію матеріалів. Контент регулярно оновлюється та підтримується для забезпечення актуальності ресурсів, які є доступними для користувачів.

2. Технологічна складова.

Технологічна складова охоплює всі технічні засоби, необхідні для зберігання, обробки та доступу до інформаційних ресурсів, зокрема сервери,

бази даних, програмне забезпечення для керування контентом, інтерфейси для взаємодії з користувачами та інструменти для забезпечення безпеки та захисту даних.

Сюди також належать пошукові системи та алгоритми обробки природної мови і машинного навчання, які покращують точність пошуку та забезпечують персоналізовані рекомендації для користувачів. Інструменти для аналізу поведінки користувачів допомагають вдосконалювати доступ до ресурсів та покращувати загальну функціональність системи.

3. Складова управління та адміністрування.

Вона включає організаційні процеси, що забезпечують ефективне керування бібліотекою, наприклад, управління правами доступу, оновленням контенту та підтримкою користувачів. Адміністративні інструменти дають можливість контролювати роботу бібліотеки, управляти обліковими записами користувачів, підтримувати авторське право та ліцензії на використання контенту. Також передбачаються механізми для збору й аналізу статистики використання, що допомагають у подальшому розвитку бібліотеки та її адаптації до потреб аудиторії.

Ці три структурні елементи працюють як єдина система, яка забезпечує зручний, безпечний та ефективний доступ до знань, дозволяючи бібліотеці надавати різноманітні інформаційні ресурси своїм користувачам у зручному електронному форматі.

1.4 Електронні колекції

Електронні колекції – це набори цифрових матеріалів, зібрані та організовані для зберігання, доступу і використання в межах електронної бібліотеки. Вони можуть включати електронні книги, журнальні статті, наукові дослідження, зображення, відео, аудіофайли, архівні документи та інші види цифрових даних.

Основна мета електронних колекцій – забезпечити користувачів легким і зручним доступом до широкого спектра інформаційних ресурсів в одному місці.

Процес створення електронних колекцій у бібліотеці включає кілька етапів:

1. Відбір та оцифрування матеріалів.

Спочатку визначаються типи та обсяги матеріалів, які необхідно включити до колекції. Це може бути певний набір тем або ресурси, пов’язані з конкретною дисципліною.

Паперові носії, які є частиною колекції, оцифровуються шляхом сканування або фотографування для переведення їх у цифровий формат. Під час оцифрування враховуються вимоги щодо якості зображень, зручності використання та технічної сумісності файлів.

2. Каталогізація та організація даних.

Оцифровані матеріали проходять процес каталогізації, під час якого їм присвоюються метадані – інформація про автора, дату публікації, тему, ключові слова тощо.

Метадані полегшують пошук матеріалів і дозволяють користувачам швидко знаходити потрібні ресурси, фільтруючи їх за різними критеріями.

3. Інтеграція до бібліотечної системи.

Готові колекції додаються до бази даних бібліотеки, після чого стають доступними для користувачів. Електронні колекції інтегруються з пошуковою системою бібліотеки, яка забезпечує доступ до матеріалів через інтерфейс користувача.

4. Забезпечення правового захисту та умов використання.

На етапі створення колекції враховуються авторські права на матеріали, щоб гарантувати легальний доступ та використання контенту. Для матеріалів, захищених авторським правом, передбачаються спеціальні умови доступу або ліцензійні угоди [4].

5. Використання електронних колекцій.

Електронні колекції забезпечують користувачів легким доступом до матеріалів у різноманітних форматах. Їх використовують для:

6. Навчання та досліджень.

Студенти, викладачі та дослідники використовують електронні колекції для отримання доступу до наукових праць, навчальних матеріалів та спеціалізованої літератури.

Колекції дозволяють швидко знаходити потрібні ресурси завдяки розширеним пошуковим можливостям, що значно спрощує проведення наукових досліджень.

7. Публічний доступ та збереження культурної спадщини.

Публічні бібліотеки використовують електронні колекції для зберігання та розповсюдження культурних та історичних матеріалів, забезпечуючи їхню доступність для широкої аудиторії.

Це сприяє збереженню рідкісних документів, архівів і рукописів, роблячи їх доступними в цифровому форматі.

8. Розширення доступності.

Користувачі можуть отримувати доступ до електронних колекцій з будь-якого місця та в будь-який час, що значно спрощує використання матеріалів для осіб з обмеженими можливостями чи для тих, хто знаходиться далеко від бібліотеки.

Електронні колекції забезпечують сучасний підхід до зберігання та розповсюдження інформації, дозволяючи бібліотекам виконувати свої освітні та культурні функції у цифрову епоху.

1.5 Аналіз існуючих розробок в області електронних бібліотек

Аналіз існуючих розробок в області електронних бібліотек виявляє, що ця галузь за останні десятиліття значно розвинулася, особливо завдяки впровадженню цифрових та інформаційних технологій. Сучасні електронні

бібліотеки поєднують у собі бази даних, системи управління контентом, інструменти для автоматизації та інтелектуальні пошукові алгоритми, які полегшують користувачам доступ до інформації.

Виділяють три типи електронних бібліотек, які орієнтовані на унікальні потреби зберігання та можливість доступу до інформації:

1. Наукові та університетські репозиторії орієнтовані на зберігання наукових статей, дисертацій, технічних звітів та інших академічних матеріалів. Прикладами є DSpace, EPrints (рис. 1.1), BASE та PubMed, які мають потужні функції пошуку та доступу до дослідницьких робіт.
2. Публічні цифрові бібліотеки пропонують широкий доступ до літератури, культурної спадщини та архівних матеріалів. Наприклад, Europeana та Національна цифрова бібліотека України забезпечують доступ до архівів та оцифрованих історичних документів.
3. Спеціалізовані бібліотеки орієнтовані на певні предметні області, наприклад, електронні бібліотеки з медичної літератури (як PubMed) або технічної літератури.

Рисунок 1.1 – Скріншот головної сторінки EPrints

EPrints – це відкрита платформа для створення та управління електронними архівами і репозиторіями наукових робіт, яка широко використовується в університетах та дослідницьких установах для зберігання наукових матеріалів. Основна мета EPrints – спростити доступ до наукових праць, технічних звітів, дисертацій та інших дослідницьких документів, забезпечуючи відкритий доступ до академічного контенту.

Характеристики та можливості EPrints:

1. Обробка запитів та пошук: EPrints підтримує індексацію та пошук по метаданих і повних текстах, що дозволяє користувачам легко знаходити необхідні документи. Хоча EPrints використовує базові методи пошуку, його можна інтегрувати з інструментами для обробки природної мови, щоб покращити розуміння пошукових запитів.

2. Відкритий доступ: платформа спеціалізується на підтримці відкритого доступу (Open Access), що дозволяє створювати публічні наукові репозиторії з вільним доступом до матеріалів, надаючи можливість користувачам у всьому світі знаходити та завантажувати наукові праці.

3. Гнучкість налаштувань: EPrints дозволяє легко налаштовувати інтерфейс, організацію даних і додавати нові функції для вдосконалення пошукових можливостей, таких як розширена фільтрація результатів, категоризація матеріалів та рекомендації для користувачів на основі їхніх інтересів.

4. Збереження наукових метаданих: система дозволяє детально зберігати метадані про кожну наукову роботу (автор, дата публікації, журнал, ключові слова), що допомагає у швидкому пошуку та організації робіт.

5. Інтеграція з іншими репозиторіями: EPrints підтримує стандарти обміну даними, такі як OAI-PMH, що дозволяє інтегруватися з іншими репозиторіями і полегшує доступ до великої кількості академічних джерел [5].

Використання в бібліотечній системі: EPrints є корисним інструментом для наукових бібліотек, університетських архівів та академічних установ, які прагнуть забезпечити відкритий доступ до наукових робіт, полегшити їхнє

збереження та покращити доступність для дослідників та студентів. Завдяки можливості налаштування, EPrints може бути адаптована для задоволення специфічних потреб кожної установи, наприклад, додавання інструментів для рекомендацій або обробки природної мови, що значно підвищує ефективність пошуку в межах архіву.

Europeana – це цифрова платформа, створена для забезпечення доступу до культурної спадщини Європи (рис. 1.2). Europeana пропонує інтерактивну онлайн-колекцію, яка містить мільйони цифрових об'єктів, таких як книги, картини, фільми, фотографії, музику, архівні документи тощо. Цей проект фінансується Європейським Союзом і об'єднує культурні установи по всій Європі для створення єдиної цифрової бібліотеки, що дозволяє користувачам з усього світу відкривати, досліджувати й використовувати європейську культурну спадщину [6].

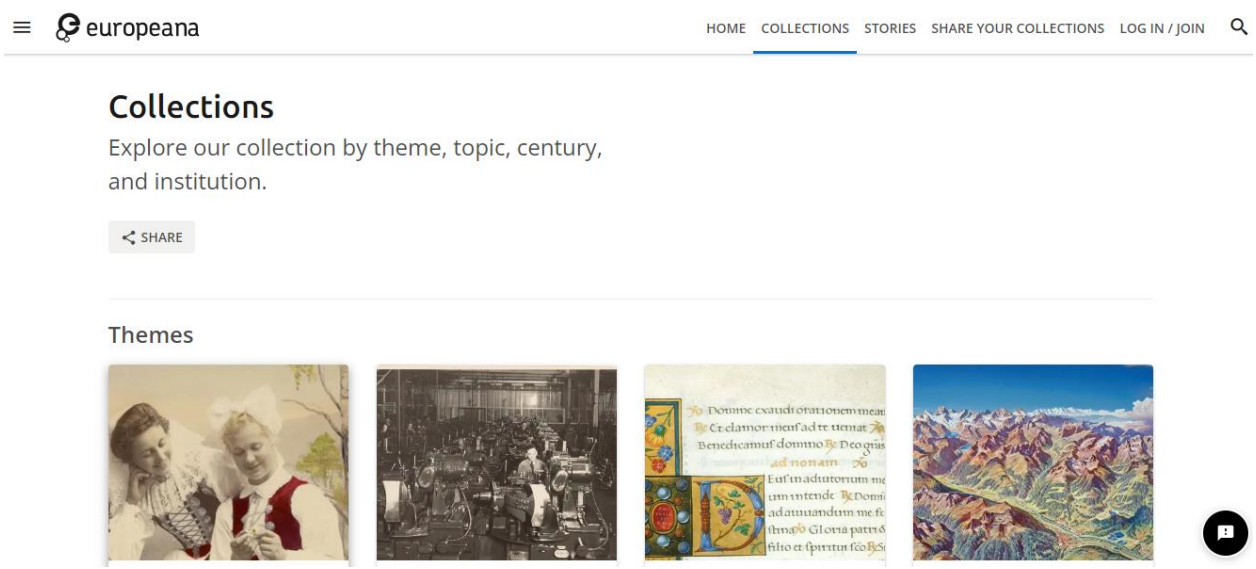

Рисунок 1.2 – Скріншот сторінки «Колекції» Europeana

Основні особливості Europeana:

1. Величезний масив контенту: Europeana містить понад 50 мільйонів об'єктів із понад 3 500 культурних установ, включаючи бібліотеки,

архіви, музеї, галереї та аудіовізуальні колекції. Контент платформи охоплює всі періоди та жанри, дозволяючи користувачам отримувати доступ до культурних об'єктів від Середньовіччя до сучасності.

2. Різноманіття форматів і типів медіа: Europeana підтримує різні типи медіа, такі як текст, зображення, аудіо та відео. Це дозволяє створювати повнішу картину культурної спадщини та забезпечує користувачам можливість працювати з різними видами контенту.
3. Інтерактивний пошук і фільтрація: Europeana забезпечує пошук за ключовими словами, категоріями, мовами, датами, географічними регіонами тощо. Платформа також використовує інструменти для фільтрації й сортування результатів, що полегшує доступ до потрібного контенту.
4. Залучення технологій семантичного пошуку: Europeana використовує метадані для покращення пошуку, дозволяючи користувачам знаходити більш релевантні результати. Семантичні технології допомагають забезпечити більш точні зв’язки між різними об’єктами, що покращує досвід користувача.
5. Відкритий доступ і можливість повторного використання: Багато з матеріалів, представлених у Europeana, знаходяться у відкритому доступі або мають ліцензії, які дозволяють їх повторне використання, що робить Europeana корисною для дослідників, освітніх установ, студентів та креативних індустрій [6].

Переваги Europeana:

- єдина точка доступу до культурної спадщини Європи: замість того, щоб звертатися до кожної бібліотеки або архіву окремо, користувачі можуть шукати матеріали з різних установ на одній платформі;
- інтеграція з освітніми проектами: Europeana активно підтримує освітні ініціативи, надаючи ресурси для навчальних цілей;

- висока якість контенту: багато матеріалів проходять ретельну перевірку, а метадані підтримуються на високому рівні;
- підтримка інноваційних проектів: Europeana регулярно бере участь у нових технологічних проектах і пропонує інтерфейси API для розробників, що дозволяє створювати інноваційні додатки на основі її даних.

Перед Europeana стоять декілька викликів, а саме:

1. Проблеми з авторським правом: через різні правові обмеження деякі матеріали можуть бути доступні лише з обмеженнями або взагалі недоступні.
2. Залежність від фінансування ЄС: проект фінансується Європейським Союзом, тому його стабільність може залежати від змін у політиці або бюджеті.
3. Проблеми з багатомовністю: незважаючи на багатомовний інтерфейс, не всі матеріали мають метадані на всіх мовах ЄС, що може обмежити їхню доступність для користувачів, які не володіють основними європейськими мовами.

Europeana є надзвичайно важливим проектом, який відкриває доступ до культурної спадщини Європи для глобальної аудиторії. Вона підтримує освітні, наукові та креативні ініціативи, зберігаючи та популяризуючи культурні цінності. Завдяки сучасним технологіям та інноваційним підходам Europeana створює нові можливості для вивчення та використання культурних матеріалів у цифровому форматі.

SpringerLink – це наукова платформа від видавництва Springer, яка надає доступ до широкого спектра академічного контенту, включаючи наукові журнали, книги, конференційні матеріали та довідкові роботи (рис. 1.3). SpringerLink активно використовується дослідниками, викладачами, студентами та науковцями з усього світу для пошуку та доступу до перевірених, рецензованих досліджень у багатьох наукових дисциплінах.

Платформа особливо популярна у сферах природничих і технічних наук, медицини, інженерії, комп'ютерних наук та суспільних наук [7].

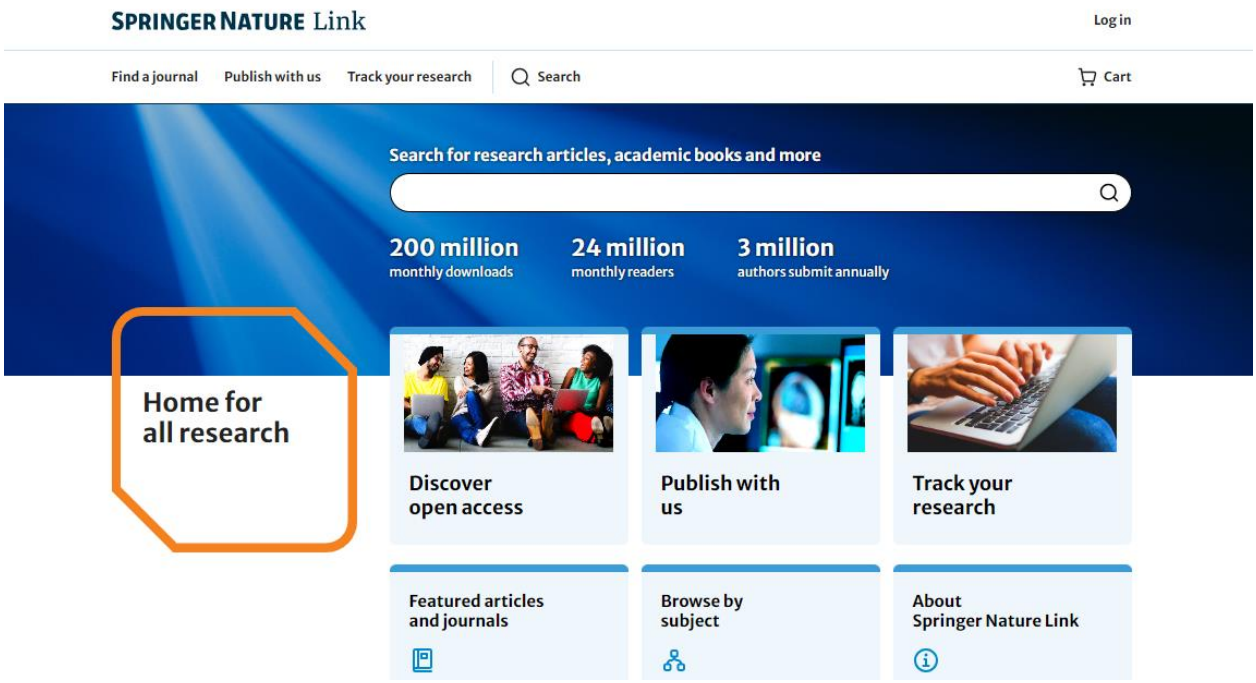

Рисунок 1.3 – Головна сторінка SpringerLink

Основні особливості SpringerLink:

1. Велика база академічного контенту.

Платформа надає доступ до більш ніж 12 мільйонів наукових статей, глав книг та матеріалів конференцій, покриваючи широкий спектр дисциплін. Це дає користувачам змогу отримати доступ до важливих наукових досягнень та передових досліджень.

2. Рецензований контент.

Усі матеріали на SpringerLink проходять ретельний процес рецензування, що забезпечує їх високу якість і наукову достовірність. Це робить платформу надійним джерелом інформації для академічних досліджень.

3. Розширені можливості пошуку.

Платформа дозволяє користувачам виконувати пошук за ключовими словами, авторами, назвами робіт, а також фільтрувати результати за дисципліною, типом контенту, датою публікації тощо. Це полегшує навігацію і доступ до релевантної інформації для наукових досліджень.

4. Інтерактивний інтерфейс з рекомендаціями.

На основі попередніх переглядів і пошукових запитів, SpringerLink пропонує користувачам релевантні рекомендації. Це допомагає знаходити пов’язані матеріали та розширює контекст досліджень.

5. Інтеграція з освітніми платформами та інституціями.

Багато університетів та наукових інституцій мають передплату на контент SpringerLink, що забезпечує зручний доступ до ресурсів студентам та науковцям без додаткових витрат.

До переваг SpringerLink слід віднести наступні:

1. Широкий доступ до наукової літератури: платформа охоплює різні дисципліни і має одне з найбільших сховищ рецензованого академічного контенту.
2. Постійні оновлення: завдяки регулярному додаванню нових публікацій користувачі мають доступ до останніх наукових досліджень та відкритів.
3. Високоякісний і надійний контент: видавництво Springer є одним із найвідоміших у світі, що гарантує наукову достовірність і рецензованість матеріалів.

Недоліки SpringerLink:

1. Платний доступ до багатьох матеріалів: хоча деякий контент є у відкритому доступі, значна частина публікацій платна, що може бути обмеженням для незалежних дослідників або установ без підписки.
2. Обмежені можливості для користувачів без акаунтів: користувачі без підписки можуть не мати повного доступу до всіх функцій, таких як завантаження PDF-версій статей або збереження в обраному.

SpringerLink є важливою платформою для наукових досліджень і академічного середовища, яка дозволяє отримати доступ до високоякісної та рецензованої літератури. Вона підтримує наукову спільноту, пропонуючи сучасні інструменти пошуку та індивідуалізовані рекомендації, що сприяє продуктивному навчальному процесу і дослідженням.

Greenstone – це система програмного забезпечення з відкритим кодом, розроблена для створення цифрових бібліотек. Вона була розроблена у співпраці з ЮНЕСКО та є одним із найбільш поширених інструментів для організації цифрових колекцій у бібліотеках, архівах, музеях та інших інформаційних установах. Greenstone підтримує створення, організацію та управління цифровими колекціями, що можуть містити різноманітні типи файлів (тексти, зображення, аудіо, відео) та працювати як на локальному комп’ютері, так і в Інтернеті [8].

Основні характеристики Greenstone:

1. Гнучкість у створенні колекційю.

Greenstone дозволяє створювати електронні колекції різних типів з широкими можливостями щодо метаданих, підтримуючи стандарти метаданих, такі як Dublin Core, MARC та інші. Це робить систему зручною для організації контенту на різних мовах та у різних форматах.

2. Потужні пошукові та індексаційні інструменти.

Платформа підтримує індексацію даних, що забезпечує швидкий і точний пошук. Користувачі можуть шукати документи за ключовими словами, фразами, категоріями, авторами, датами тощо. Індексація відбувається автоматично при завантаженні нових матеріалів.

3. Підтримка декількох форматів файлів.

Greenstone дозволяє зберігати й організовувати контент у різних форматах – текстові документи, PDF, HTML, зображення, аудіо- та відеофайли. Ця різноманітність форматів дозволяє створювати мультимедійні колекції, що є важливим для багатьох установ.

4. Підтримка багатомовності.

Інтерфейс користувача Greenstone перекладений на багато мов, що робить його доступним для користувачів з різних країн і дозволяє створювати багатомовні колекції.

5. Розширюваність та налаштовуваність.

Greenstone дозволяє адміністраторам налаштовувати інтерфейс і функціональність бібліотеки відповідно до потреб. Крім того, оскільки Greenstone є проєктом з відкритим кодом, його можна модифікувати для додавання нових функцій або інтеграції з іншими системами.

6. Веб-доступ та локальне розгортання.

Greenstone може бути встановлений локально або на сервері, що надає доступ до колекцій через Інтернет. Це дозволяє бібліотекам, музеям та іншим установам поширювати свої ресурси як у локальній мережі, так і через веб-інтерфейс.

Переваги Greenstone:

- безкоштовне використання завдяки ліцензії з відкритим кодом, що робить його привабливим для некомерційних організацій;
- гнучкість і багатофункціональність, яка дозволяє налаштувати систему під специфічні потреби установи;
- підтримка різних форматів метаданих і можливість створення мультимедійних колекцій;
- відкритий код дозволяє налаштовувати програмне забезпечення і інтегрувати його з іншими системами.

Недоліки Greenstone:

- технічна складність налаштування, яка може вимагати досвіду у програмуванні та знання адміністрування серверів;
- відсутність вбудованих інструментів для машинного навчання або вдосконаленого семантичного пошуку, що може обмежити можливості персоналізації або автоматизації в великих бібліотеках.

Greenstone є чудовим вибором для невеликих та середніх бібліотек, навчальних і наукових закладів, які хочуть організувати свої цифрові колекції з мінімальними витратами.

Якщо аналізувати функціональні можливості сучасних електронних бібліотек, то слід визначити наступні:

1. Пошук та фільтрація: розробки у сфері електронних бібліотек включають різноманітні методи пошуку, від базових пошукових систем до складних алгоритмів NLP. Такі системи, як Semantic Scholar і Google Scholar, використовують семантичний пошук для полегшення роботи з науковими запитами.
2. Оцифрування та каталогізація: цифрові бібліотеки мають інструменти для оцифрування паперових носіїв, метаданогою обробки та автоматичної класифікації, що допомагає зберігати матеріали та робити їх легкодоступними.
3. Забезпечення відкритого доступу: системи типу Open Access дозволяють створювати репозиторії з вільним доступом до матеріалів, що особливо актуально для наукових робіт.
4. Система рекомендацій: Завдяки аналізу поведінки користувачів, такі платформи як JSTOR або SpringerLink можуть рекомендувати схожі матеріали, що полегшує пошук корисних джерел.

Основні тенденції розвитку таких систем:

1. Впровадження штучного інтелекту та машинного навчання.

Для поліпшення пошуку, класифікації та персоналізації в електронних бібліотеках активно застосовуються алгоритми машинного навчання. Наприклад, Google Books використовує технології NLP, щоб розуміти контекст запитів і надавати релевантні результати.

2. Інтеграція з глобальними мережами та іншими ресурсами.

Багато бібліотек використовують стандарти для обміну даними, такі як OAI-PMH, що дозволяє об'єднувати дані з різних джерел, як це робить Europeana.

3. Мобільний доступ та хмарні технології.

Сучасні електронні бібліотеки надають доступ до колекцій із мобільних пристроїв, а зберігання та обробка даних все частіше здійснюються на хмарних сервісах для зручності та швидкого доступу.

Отже, слід зробити висновки щодо викликів та перспектив таких платформ:

- забезпечення авторського права: використання матеріалів у цифрових бібліотеках потребує дотримання авторських прав і ліцензійних умов, що іноді ускладнює їхнє поширення та обмежує доступ;
- підтримка багатомовності та локалізації: для глобальних електронних бібліотек актуальною є проблема забезпечення доступу до матеріалів різними мовами та підтримки локалізації для користувачів з різних країн;
- підтримка високої якості метаданих: повнота і точність метаданих впливають на точність пошуку та зручність користування бібліотекою.

Аналіз існуючих електронних бібліотек свідчить про їх значний розвиток завдяки новітнім технологіям, таким як машинне навчання, хмарні сервіси та інтеграція з глобальними базами даних. Такі розробки забезпечують ширший доступ до інформації, покращують пошук і зберігають цінні знання у зручному форматі, створюючи нові можливості для освітнього, наукового та культурного використання.

2 ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ ДЛЯ РЕАЛІЗАЦІЇ NLP У БІБЛІОТЕЧНИХ СИСТЕМАХ

2.1 Natural Language Processing

Обробка природної мови (Natural Language Processing, NLP) – це галузь штучного інтелекту, що займається взаємодією між комп'ютерами та людьми за допомогою природної мови. NLP включає методи та технології, які дозволяють комп'ютерам аналізувати, розуміти, генерувати й взаємодіяти з текстом або мовою в спосіб, схожий на людський.

Сучасне NLP поєднує знання лінгвістики, статистики, алгоритмів машинного навчання та глибинного навчання. NLP використовується для широкого спектру завдань, пов'язаних з мовою, включаючи відповіді на запитання, класифікацію тексту різними способами та спілкування з користувачами [9].

У контексті систем для бібліотек NLP відіграє ключову роль у створенні зручних та інтелектуальних механізмів пошуку, аналізу запитів, рекомендацій та персоналізації:

1. Пошук книг і матеріалів – NLP допомагає зробити пошуковий механізм системи більш адаптивним і розумним:

- розпізнавання наміру (Intent Recognition): наприклад, якщо користувач вводить "книги про машинне навчання для початківців", система розуміє, що потрібно запропонувати популярні вступні книги в цій галузі;
- синоніми та контекст: навіть якщо запит сформульований некоректно, наприклад "вчити Python", система розуміє, що користувач шукає ресурси для вивчення Python.

Перевага:

- забезпечення точних результатів навіть при розмитих або складних запитах;
- підтримка пошуку різними мовами.

2. Рекомендації – завдяки NLP бібліотечна система може:

- аналізувати історію пошукових запитів користувача;
- використовувати моделі семантичної схожості (наприклад, Word2Vec або BERT), щоб пропонувати книги, схожі за змістом або тематикою;
- враховувати відгуки й рейтинги інших користувачів.

Перевага: персоналізація рекомендацій відповідно до інтересів кожного користувача.

3. Автоматизація обробки тексту – NLP дозволяє автоматично аналізувати тексти, які зберігаються в бібліотеці:

- аналіз метаданих: витяг сутностей, таких як автор, рік публікації, жанр;
- класифікація текстів: розподіл книг за жанрами, темами, рівнем складності;
- анотації: автоматичне створення коротких описів книг.

Перевага: швидка обробка великих обсягів даних.

4. Робота з багатомовними даними – NLP забезпечує обробку даних на різних мовах:

- переклад запитів і текстів для користувачів з різних країн;
- використання моделей машинного перекладу (наприклад, Google Translate API, OpenNMT).

Перевага: доступність матеріалів користувачам з усього світу.

5. Інтерактивна взаємодія – NLP використовується для створення чат-ботів, які допомагають користувачам у пошуку книг чи відповідають на запитання. Наприклад: діалогова система приймає запит "Можете порадити сучасну фантастику для дітей?", бот розпізнає жанр ("сучасна фантастика") і аудиторію ("діти") та пропонує відповідні книги.

Перевага: підвищення зручності для користувачів завдяки інтерактивності.

6. Семантичний пошук – використання NLP дозволяє здійснювати семантичний пошук. Наприклад, якщо користувач шукає "книги про ШІ та майбутнє", система може знайти матеріали, які не містять точних ключових слів, але мають схожий зміст.

Приклад моделей:

- BERT: розуміння контексту запиту;
- TF-IDF: визначення ключових термінів у текстах.

7. Аналіз запитів користувачів – NLP допомагає аналізувати й групувати пошукові запити для:

- виявлення популярних тем;
- підтримки розширення колекції книг у відповідь на запити.

Переваги NLP для бібліотечних систем:

1. Підвищення точності пошуку.
2. Збільшення зручності використання.
3. Персоналізація та інтерактивність.
4. Здатність обробляти великі обсяги текстів та метаданих [9].

NLP дозволяє електронним бібліотекам еволюціонувати, пропонуючи не лише зручний доступ до ресурсів, але й створюючи унікальний досвід взаємодії для кожного користувача.

2.2 Моделі обробки тексту

Для створення інтелектуальних систем пошуку та рекомендацій на основі NLP у бібліотечних системах можна використовувати широкий спектр технологій і інструментів. Вони допомагають обробляти текстові запити, аналізувати дані та забезпечувати високий рівень персоналізації.

2.2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) – це мовна модель, структура машинного навчання з відкритим кодом для обробки

природної мови. BERT розроблено, щоб допомогти комп’ютерам зрозуміти значення неоднозначної мови в тексті, використовуючи навколишній текст для встановлення контексту. Структура BERT була попередньо навчена з використанням тексту з Вікіпедії та може бути налаштована за допомогою наборів даних запитань і відповідей.

Історично мовні моделі могли лише послідовно читати введений текст – зліва направо або справа наліво – але не могли робити те й інше одночасно. BERT відрізняється тим, що він призначений для читання в обох напрямках одночасно. Поява трансформаторних моделей увімкнула цю можливість, яка відома як двонаправленість.

Використовуючи двоспрямованість, BERT проходить попередню підготовку до двох різних, але пов’язаних завдань NLP: моделювання замаскованої мови (MLM) і передбачення наступного речення (NSP). Мета навчання MLM полягає в тому, щоб приховати слово в реченні, а потім змусити програму передбачити, яке слово було приховано на основі контексту прихованого слова. Мета навчання NSP полягає в тому, щоб програма передбачила, чи два наведені речення мають логічний послідовний зв’язок чи їхній зв’язок просто випадковий [10].

Мета будь-якої методики NLP – зрозуміти людську мову так, як вона вимовляється природним шляхом. У випадку BERT це означає передбачення слова в пустому місці. Для цього моделі зазвичай навчаються, використовуючи велике сховище спеціалізованих навчальних даних з мітками. У цьому процесі лінгвісти виконують трудомістке ручне маркування даних .

Однак BERT пройшов попередню підготовку, використовуючи лише колекцію немаркованого простого тексту, а саме всю англійську Вікіпедію та Корпус Брауна. Він продовжує навчатися за допомогою неконтрольованого навчання тексту без міток і вдосконалюється, навіть якщо його використовують у практичних програмах, таких як пошук Google.

Переваги BERT: підходить для семантичного пошуку, класифікації тексту та витягу сутностей. Приклад використання: аналіз складних запитів

користувача, як-от "книги для вивчення нейромереж у Python для початківців"[11].

2.2.2 Generative Pre-trained Transformer

Наступна модель – GPT (Generative Pre-trained Transformer). Це сімейство великих мовних моделей, розроблених компанією OpenAI. GPT базується на архітектурі трансформерів, що є основою сучасної обробки природної мови (NLP). Її унікальність полягає у здатності до генерації текстів, контекстного розуміння та виконання багатьох мовних завдань без спеціалізованого навчання.

Архітектура трансформера використовує механізм селф-атеншн (self-attention) для розуміння контексту слів у тексті. Архітектура оптимізована для паралельної обробки даних, що робить модель швидкою та ефективною.

GPT проходить два етапи навчання:

- Pre-training: модель вчиться на величезних текстових корпусах (наприклад, статтях, книгах, новинах) без нагляду, щоб зрозуміти структуру мови;
- Fine-tuning: адаптація до конкретних завдань з використанням меншого набору спеціалізованих даних.

Модель може виконувати широкий спектр завдань NLP:

- генерація тексту;
- резюмування;
- відповіді на запитання;
- переклад мов;
- семантичний аналіз;
- аналіз настрою тексту.

GPT здатна враховувати контекст у межах великих текстів завдяки використанню токенів і позиційного кодування. Генеративні можливості: модель може створювати логічно зв'язаний текст, який є граматично правильним і відповідним за змістом до контексту запиту.

Приклад використання: написання анотацій до книг або створення текстових відповідей на запити.

Роль GPT у бібліотечних системах:

1. Генерація рекомендацій.

GPT може аналізувати текст запитів і пропонувати користувачам книги або колекції на основі їхніх інтересів.

2. Автоматизація створення метаданих.

Генерація анотацій до книг, резюме статей або описів колекцій.

3. Діалогові системи.

Використання GPT для створення чат-ботів, які можуть відповідати на складні запити та допомагати користувачам знаходити необхідну інформацію.

4. Персоналізація.

Інтелектуальний аналіз історії пошуку або читацьких уподобань для надання персоналізованих рекомендацій.

5. Підтримка багатомовності.

GPT може обробляти запити різними мовами та перекладати контент для інтернаціональних користувачів.

GPT є потужним інструментом для реалізації функціональних можливостей бібліотечних систем, пов'язаних із обробкою текстів та взаємодією з користувачами. Завдяки своїй універсальності та контекстуальному розумінню, модель здатна забезпечити високу якість рекомендацій, автоматизацію та ефективний пошук інформації.

2.2.3 Word2Vec

Word2Vec – це алгоритм, розроблений командою Google для перетворення слів у числові вектори (представлення слів), що зберігають їхній семантичний зміст. Модель вперше була представлена у 2013 році і швидко стала популярною у задачах NLP.

На відміну від традиційного представлення тексту у вигляді «мішку слів» (bag-of-words), Word2Vec створює вектори слів, які відображають їхній контекст у тексті, а також відношення між словами. Це дозволяє використовувати математичні операції для аналізу семантичної близькості.

Основні характеристики Word2Vec:

1. Представлення слів у вигляді векторів:
   - алгоритм генерує числові вектори, де кожне слово представляється у багатовимірному просторі (зазвичай 100-300 вимірів);
   - близькі за значенням слова розташовані ближче одне до одного в цьому просторі.
2. Два основних підходи до навчання:
   - Continuous Bag of Words (CBOW) передбачає слово, виходячи з його контексту (оточуючих слів). Більш ефективний для великих наборів даних;
   - Skip-Gram передбачає контекст (оточуючі слова), виходячи з певного слова. Краще працює для рідкісних слів.
3. Використання великих текстових корпусів: Word2Vec вчиться на великих обсягах текстів (новини, статті, книги) для формування точних семантичних зв’язків між словами [12].

Переваги Word2Vec наступні:

1. Збереження семантики: вектори відображають не лише значення окремого слова, але й його зв'язки з іншими словами в контексті.
2. Обчислювальна ефективність: навчання Word2Vec є відносно швидким навіть на великих наборах даних.
3. Універсальність: модель може бути використана для різних завдань, таких як класифікація тексту, кластеризація, побудова рекомендацій.

До обмежень слід віднести наступні Word2Vec:

1. Ігнорує послідовність слів: алгоритм не враховує порядок слів, що може бути важливим у складних мовних конструкціях.

2. Залежність від розміру даних: для досягнення високої точності потрібні великі текстові корпуси.
3. Фіксований словниковий запас: нові слова, які не були у текстовому корпусі під час навчання, не можуть бути представлені векторами.

Роль Word2Vec у бібліотечних системах:

1. Пошук релевантної літератури: векторизація запитів користувачів і документів дозволяє знаходити книги, які максимально відповідають потребам.
2. Кластеризація: Word2Vec допомагає групувати книги або статті за тематикою, використовуючи близькість векторів.
3. Семантичний пошук: аналіз запитів, щоб знаходити документи з синонімічними або тематично спорідненими словами.
4. Рекомендаційні системи: пошук книг на основі схожих описів або інтересів користувачів.
5. Багатомовність: моделі Word2Vec для різних мов можуть допомагати в інтеграції бібліотечних систем для міжнародної аудиторії.

2.3 Методи обробки запитів

2.3.1 Named Entity Recognition

Named Entity Recognition (NER) – це завдання NLP, яке полягає у виявленні і класифікації певних сутностей (entities) у тексті. Ці сутності можуть бути іменами людей, назвами організацій, географічними місцями, датами, числами, назвами продуктів, подій тощо.

NER є одним із ключових компонентів NLP, який широко застосовується для розуміння тексту і витягування значущої інформації з неструктурованих даних.

Основні етапи роботи NER:

1. Токенізація: розбивка тексту на окремі слова або токени.

2. Визначення сутностей: модель визначає, які слова чи фрази є сутностями.
3. Класифікація сутностей: виявлені сутності класифікуються за наперед визначеними категоріями (наприклад, PER – особа, LOC – місце, ORG – організація).

Методи реалізації NER:

1. Правила та словники.

Використовуються регулярні вирази та наперед визначені списки ключових слів. Підходить для специфічних доменів, але обмежений у масштабованості та гнучкості.

2. Машинне навчання.

Використання алгоритмів, таких як Hidden Markov Models (HMM) та Conditional Random Fields (CRF). Ці підходи вимагають ручного створення ознак (features), наприклад, довжини слів, використання великих літер тощо.

3. Глибоке навчання.

Використання рекурентних нейронних мереж (RNN), наприклад LSTM, або трансформерів (наприклад, BERT). Забезпечує вищу точність завдяки здатності враховувати контекст і складні зв'язки між словами.

Роль NER у бібліотечних системах:

1. Автоматизація пошуку: виявлення назв книг, авторів, дат або тем у текстових запитах користувачів.
2. Аналіз контенту: ідентифікація ключових сутностей у текстах метаданих, описах або анотаціях книг.
3. Класифікація документів: використання сутностей для класифікації та впорядкування документів у каталозі бібліотеки.
4. Рекомендаційні системи: на основі виявлених сутностей (наприклад, автора чи жанру) надавати релевантні рекомендації.
5. Інтеграція з багатомовними системами: використання NER для підтримки пошуку у бібліотеках з контентом різними мовами [13].

2.3.2 Semantic Parsing

Семантичний аналіз (Semantic Parsing) – це процес перетворення тексту природної мови в структуровану форму, яка відображає його точне значення та логічну структуру. У задачах обробки природної мови семантичний аналіз допомагає розуміти зміст тексту, інтерпретувати запити користувачів і відповідати на них у контексті.

Семантичний аналіз спрямований на інтерпретацію тексту, тобто зрозуміти не лише окремі слова, але й їхнє значення у контексті. Побудову структурованого представлення: текст перетворюється на логічну форму, наприклад, дерево синтаксичного розбору чи граф знань. Автоматизацію розв'язання завдань: використання структурованої інформації для автоматичного виконання завдань, наприклад, пошуку релевантної інформації чи відповіді на запитання.

Процес семантичного аналізу полягає у наступних кроках:

1. Попередня обробка: токенізація, лематизація, частиномовний аналіз.
2. Ідентифікація сутностей та зв’язків: виявлення ключових сутностей і встановлення відношень між ними.
3. Побудова семантичного представлення: формалізація тексту у вигляді логічної структури.
4. Інтерпретація та виконання: використання семантичної структури для виконання дій, наприклад, відповіді на запитання чи пошуку даних.

Методи семантичного аналізу:

1. Правила на основі граматики.

Використання наперед заданих правил для перетворення тексту у формальні структури. Підходить для специфічних доменів, але складно масштабується.

2. Машинне навчання.

Алгоритми, такі як: CRF (Conditional Random Fields), SVM (Support Vector Machines). Використовуються для навчання моделей розпізнавання семантичних структур.

3. Глибоке навчання.

Seq2Seq моделі (послідовність у послідовність). Наприклад, LSTM або GRU для перетворення тексту у структуровані дані. Моделі трансформерів (Transformer): використання BERT, GPT для розуміння контексту тексту.

4. Зовнішні знання.

Інтеграція графів знань, таких як DBpedia, Wikidata, для уточнення значень сутностей та їх зв’язків.

Роль семантичного аналізу у бібліотечних системах:

1. Розуміння пошукових запитів.
2. Автоматична класифікація.
3. Рекомендаційні системи.
4. Збагачення даних.
5. Мультиформатний пошук.

3 ПРОЕКТУВАННЯ ПОШУКОВОЇ СИСТЕМИ

3.1 Опис функціональних вимог

Для опису варіантів використання роботи пошукової системи бібліотек з елементами машинного навчання слід розподілити варіанти використання для користувача та системи як окремих акторів. Такий підхід спрощує розуміння різних сценаріїв взаємодії й дозволяє чітко визначити ролі кожного з них у системі. Діаграма варіантів використання для користувача представлено на рис. 3.1.

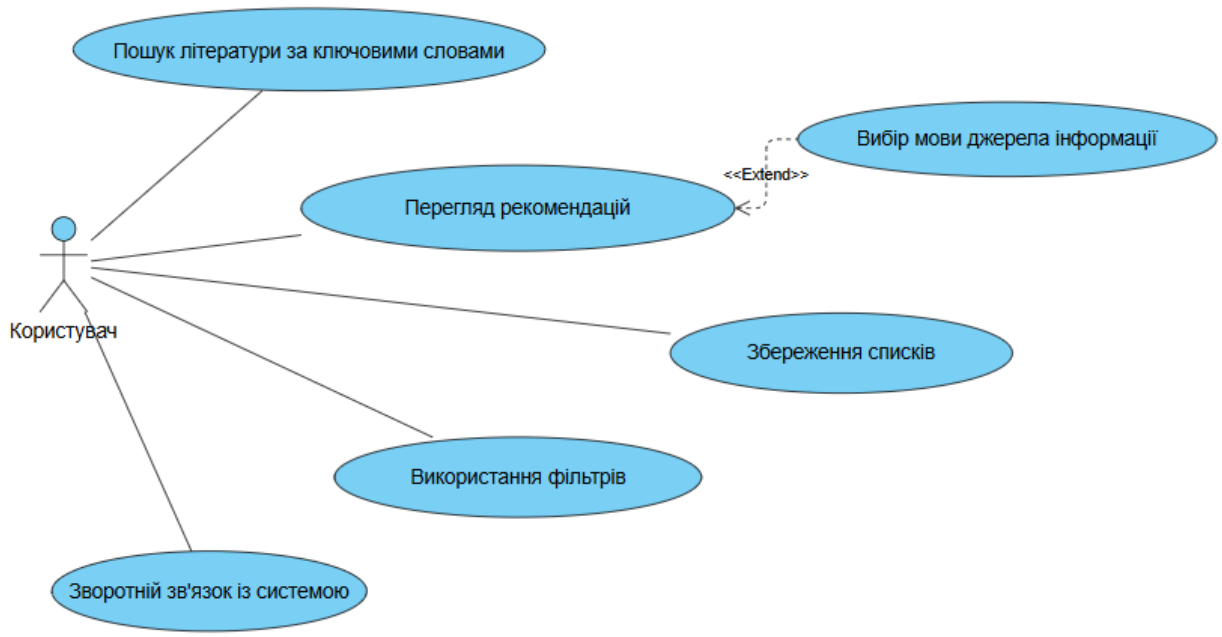

Рисунок 3.1 – Діаграма варіантів використання для актора-користувача системи

Далі наведено опис кожного варіанту використання більш детально.

ВВ №1 «Пошук літератури за ключовими словами»

Актор: Користувач.

Основний сценарій:

1. Користувач вводить запит (наприклад, "наукова фантастика").

2. Система показує список відповідних книг.
3. Користувач переглядає результати та обирає книгу.

Альтернативний сценарій: якщо запит нечіткий, система пропонує уточнення.

ВВ №2 «Перегляд рекомендацій»

Актор: Користувач.

Основний сценарій:

1. Користувач переходить до розділу "Рекомендації".
2. Система відображає персоналізований список книг на основі його історії пошуків і вподобань.
3. Користувач обирає книгу для перегляду деталей.

Альтернативний сценарій: якщо історії немає, система пропонує популярні книги.

ВВ №3 «Збереження списків»

Актор: Користувач.

Основний сценарій:

1. Користувач обирає кілька книг і натискає "Зберегти список".
2. Система додає книги до його персонального списку.
3. Користувач переглядає список пізніше.

Альтернативний сценарій: якщо користувач не авторизований, система пропонує увійти.

ВВ №4 «Використання фільтрів»

Актор: Користувач.

Основний сценарій:

1. Користувач виконує пошук.
2. Налаштовує фільтри (автор, рік, категорія тощо).
3. Система оновлює результати пошуку.

Альтернативний сценарій: система пропонує рекомендації, якщо знайдено мало результатів.

ВВ №5 «Зворотній зв'язок із системою»

Актор: Користувач.

Основний сценарій:

1. Користувач переглядає книгу і залишає відгук або оцінку.
2. Система використовує цей відгук для покращення рекомендацій.

Альтернативний сценарій: система висвічує помилку відправки повідомлення і просить повторити запит на відправку користувача.

ВВ №6 «Вибір мови джерела інформації».

Актор: Користувач.

Основний сценарій:

1. Користувач виконує пошук літератури за ключовими словами та вказує мову джерела.
2. Система відображає результати пошуку.

Альтернативний сценарій: якщо обране джерело недоступне, система пропонує джерела іншими мовами на задану тематику.

На рис. 3.2 представлено діаграму варіантів використання для актора-пошукової системи. Нижче представлено опис кожного варіанту використання для цієї діаграми.

ВВ №1 «Аналіз текстового запиту»

Актор: Система (модуль NLP).

Основний сценарій:

1. Користувач вводить текстовий запит.
2. Модуль NLP розпізнає ключові слова, сутності та їхні зв'язки.
3. Система готує релевантні результати для користувача.

Альтернативний сценарій: якщо запит неоднозначний, система запитує уточнення.

ВВ №2 «Підбір рекомендацій»

Актор: Система (рекомендаційний модуль).

Основний сценарій:

1. Система аналізує історію пошуків, збережені списки та оцінки користувача.
2. Визначає релевантні книги за допомогою кластеризації.
3. Створює список рекомендованих книг.

Альтернативний сценарій: якщо даних про користувача недостатньо, система показує популярні книги.

Рисунок 3.2 – Діаграма варіантів використання для актора-пошукової системи

ВВ №3 «Індексація нових матеріалів»

Актор: Система (модуль індексації).

Основний сценарій:

1. Нові дані надходять до бази (через API або файл).

2. Система індексує книги, витягує метадані (автор, категорія, ключові слова).
3. Інформація стає доступною для пошуку.

Альтернативний сценарій: якщо метадані неповні, система використовує NLP для їх доповнення.

ВВ №4 «Моніторинг релевантності»

Актор: Система.

Основний сценарій:

1. Система аналізує історію взаємодій користувачів із результатами пошуку.
2. Визначає, які запити були успішними (за часом взаємодії чи відгуками).
3. Регулює моделі пошуку для підвищення точності.

Альтернативний сценарій: система сигналізує адміністратору про потребу втручання, якщо виникають проблеми з релевантністю.

ВВ №5 «Генерація статистики»

Актор: Система (аналітичний модуль).

Основний сценарій:

1. Система збирає дані про використання (запити, рекомендації, взаємодії).
2. Аналізує тренди та популярність книг.
3. Створює звіти для адміністраторів.

Альтернативний сценарій: якщо дані недостатні, система пропонує додати нові джерела.

3.2 Нефункціональні вимоги

Наступні нефункціональні вимоги забезпечують високу якість роботи системи, її доступність для широкого кола користувачів та здатність адаптуватися до змін у бібліотечній сфері.

Нефункціональні вимоги до пошукової системи для бібліотеки:

1. Продуктивність.

Система повинна забезпечувати швидке виконання запитів, навіть при великій кількості користувачів. Час відповіді на пошук не повинен перевищувати 2 секунд для запитів із середньою складністю. Індексація нових матеріалів має відбуватися у фоновому режимі, не впливаючи на швидкість роботи користувачів.

2. Масштабованість.

Система повинна підтримувати додавання нових джерел інформації та зростання бази даних без значного зниження продуктивності. Можливість обробки запитів великої кількості одночасних користувачів (наприклад, 1000 активних сесій одночасно).

3. Надійність.

Система повинна бути стійкою до збоїв і забезпечувати безперервну роботу з доступністю не менше 99,9% часу. Автоматичне відновлення після збоїв або помилок у роботі.

4. Безпека.

Захист даних користувачів (історія пошуків, персональні рекомендації) за допомогою сучасних механізмів шифрування. Реалізація аутентифікації та авторизації користувачів для обмеження доступу до персоналізованої інформації. Відповідність системи вимогам до захисту даних, наприклад, GDPR (для європейських користувачів).

5. Зручність використання (юзабіліті).

Інтерфейс має бути інтуїтивно зрозумілим, з мінімальною кількістю кроків для виконання основних операцій. Адаптивний дизайн, який працює на різних пристроях (комп’ютерах, планшетах, смартфонах). Локалізація інтерфейсу (можливість обрати мову).

6. Сумісність.

Підтримка роботи з різними операційними системами (Windows, macOS, Linux) та браузерами (Chrome, Firefox, Safari, Edge). Інтеграція з зовнішніми

бібліотечними системами та базами даних (наприклад, Europeana, SpringerLink).

7. Підтримка масштабування.

Використання модульної архітектури для додавання нових функцій (наприклад, підтримка додаткових мов або форматів даних). Готовність до інтеграції з новими технологіями NLP та інструментами машинного навчання.

8. Модульність та розширюваність.

Система повинна підтримувати легке оновлення окремих модулів без зупинки всієї платформи. Простота додавання нових функцій, таких як аналіз емоційних тонів у відгуках користувачів або нові алгоритми рекомендацій.

9. Вимоги до ресурсів.

Ефективне використання обчислювальних ресурсів (процесор, пам’ять).

Система повинна працювати на сервері середньої конфігурації та підтримувати хмарне розгортання (AWS, Azure, Google Cloud).

10. Обробка помилок.

Система повинна повідомляти користувачів про помилки у ввічливій формі та пропонувати способи їх виправлення (наприклад, уточнення запиту). Логи помилок повинні автоматично зберігатися для аналізу адміністратором.

11. Підтримка різних форматів.

Можливість обробки та відображення даних у різних форматах (текст, зображення, аудіо, відео). Підтримка метаданих для кращої категоризації та пошуку матеріалів.

3.3 Проектування архітектури

При проектуванні архітектури пошукової системи для бібліотеки з елементами NLP необхідно враховувати такі фактори: масштабованість, інтеграція з іншими системами, підтримка сучасних технологій та забезпечення швидкої роботи.

В такому випадку мікросервісна архітектура ідеально підходить:

- кожна частина системи (пошук, NLP-аналіз, рекомендації, авторизація) може бути розроблена, розгорнута та масштабована окремо;
- легкість інтеграції нових компонентів (наприклад, додавання нових мов для NLP або інтеграція з іншою бібліотекою);
- висока надійність – збій одного модуля не порушує роботу всієї системи.

На діаграмі представлена архітектура пошукової системи для бібліотеки, що включає основні компоненти та їх взаємозв'язки (рис. 3.3).

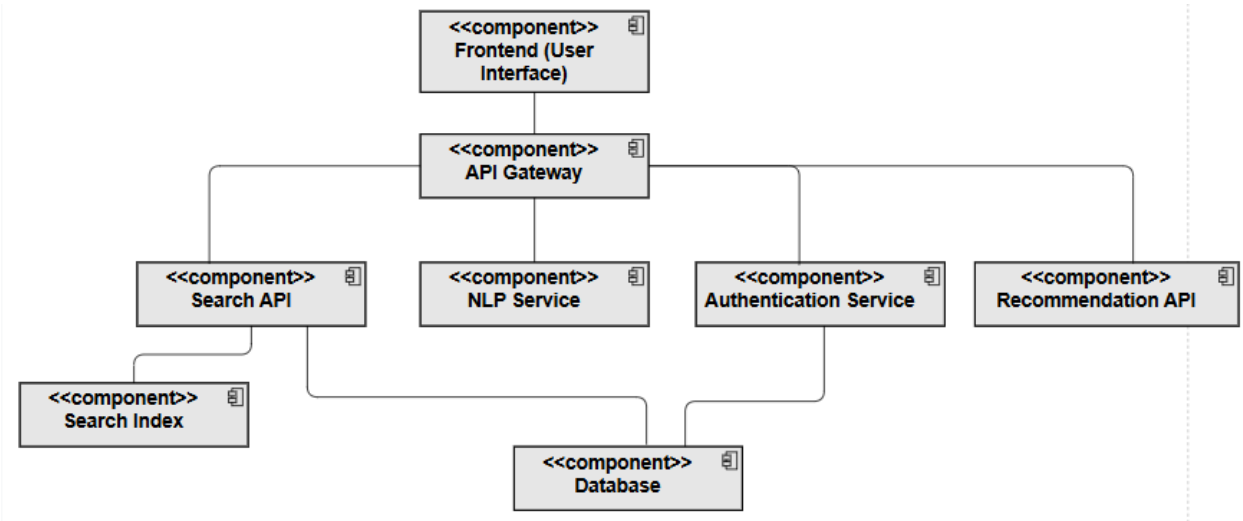

Рисунок 3.3 – Діаграма компонентів системи

Компоненти та їх функції:

1. Frontend (User Interface): інтерфейс для користувачів, через який вони здійснюють пошук, отримують рекомендації та взаємодіють із системою. Надсилає запити до API Gateway.
2. API Gateway є посередником між Frontend і мікросервісами. Розподіляє запити до відповідних сервісів: пошукового, NLP, рекомендацій чи автентифікації.

3. Search API обробляє запити на пошук книг чи колекцій. Взаємодіє з Elasticsearch для швидкого текстового пошуку. Крім цього, він отримує метадані книг із PostgreSQL.
4. NLP Service аналізує природну мову користувача, розпізнає сутності, контекст та формує структуровані запити для пошуку чи рекомендацій.
5. Recommendation API генерує персоналізовані рекомендації на основі даних користувача (історія пошуку, уподобання). Він використовує дані з NLP Service і зберігає результати в базі даних.
6. Authentication Service керує доступом і правами користувачів. Підтримує аутентифікацію та зберігає відповідну інформацію в базі даних.
7. Database зберігає метадані книг, користувачів, історію пошуку та рекомендацій. Взаємодіє із Search API, Recommendation API та Authentication Service.
8. Search Index (Elasticsearch) забезпечує швидкий пошук текстової інформації (назви книг, ключові слова). Інтегрується із Search API.

Взаємозв'язки на діаграмі компонентів наступні:

- Frontend надсилає всі запити через API Gateway, який розподіляє їх між мікросервісами;
- Search API взаємодіє з Elasticsearch для пошуку тексту та з PostgreSQL для отримання додаткових даних;
- NLP Service обробляє текст користувача та передає результати до Recommendation API;
- Recommendation API формує рекомендації на основі даних з NLP і бази даних;
- Authentication Service забезпечує безпеку доступу до системи.

Ця структура гарантує масштабованість, продуктивність і зручність у підтримці системи.

3.4 Проектування структури БД

Для коректної роботи пошукової системи бібліотечного фонду слід передбачити можливість зберігання інформації щодо книг та їх авторів, сформованих колекцій по ключовим словам, дані користувачів та інше. Для бази даних було спроектовано 14 таблиць.

Таблиця «Users» (Користувачі) – необхідна для зберігання інформацію про користувачів системи.

Поля:

- user_id (PK, int, автоінкремент) – унікальний ідентифікатор користувача;
- username (varchar, унікальний) – ім’я користувача;
- email (varchar, унікальний) – електронна пошта користувача;
- password_hash (varchar) – хеш пароля;
- role (varchar) – роль користувача (наприклад, "читач", "адміністратор");
- created_at (timestamp) – дата реєстрації.

Таблиця «Books» (Книги) – її призначення зберігання інформацію про книги, доступні у бібліотеці.

Поля:

- book_id (PK, int, автоінкремент) – унікальний ідентифікатор книги;
- title (varchar) – назва книги;
- author (varchar) – автор книги;
- publisher (varchar) – видавництво;
- published_year (year) – рік видання;
- genre (varchar) – жанр книги;
- language (varchar) – мова книги;
- summary (text) – короткий опис книги;

- isbn (varchar, унікальний) – унікальний ідентифікатор книги (ISBN);
- created_at (timestamp) – дата додавання книги.

Таблиця «SearchQueries» (Пошукові запити) зберігає історію пошукових запитів користувачів.

Поля:

- query_id (PK, int, автоінкремент) – унікальний ідентифікатор запиту;
- user_id (FK) – користувач, який виконав запит;
- query_text (text) – текст пошукового запиту;
- results_count (int) – кількість знайдених результатів;
- executed_at (timestamp) – час виконання запиту.

Таблиця «Recommendations» (Рекомендації) зберігає результати персоналізованих рекомендацій.

Поля:

- recommendation_id (PK, int, автоінкремент) – унікальний ідентифікатор рекомендації;
- user_id (FK) – користувач, для якого створено рекомендацію;
- book_id (FK) – книга, яку рекомендовано;
- created_at (timestamp) – дата створення рекомендації.

Таблиця «Logs» (Журнал подій) зберігає події для моніторингу та аналізу роботи системи.

Поля:

- log_id (PK, int, автоінкремент) – унікальний ідентифікатор запису;
- event_type (varchar) – тип події (пошук, автентифікація, помилка тощо);
- user_id (FK, NULLABLE) – користувач, пов’язаний із подією.
- details (text) – деталі події;
- timestamp (timestamp) – час створення запису.

Таблиця «Keywords» (Ключові слова) для ефективного аналізу запитів і їх індексації.

Поля:

- keyword_id (PK, int) – унікальний ідентифікатор ключового слова;
- keyword (varchar) – ключове слово або фраза;
- book_id (FK) – прив’язка до книги, пов'язаної з ключовим словом.

Таблиця «Tags» (Теги) необхідна для маркування книг і колекцій.

Поля:

- tag_id (PK, int) – унікальний ідентифікатор тегу;
- name (varchar) – назва тегу;
- description (text) – опис тегу.

Таблиця «BookTags» (Зв'язок книг і тегів):

- book_id (FK) – прив’язка до книги;
- tag_id (FK) – прив’язка до тегу.

Таблиця «Collections» (Колекції) для об'єднання книг за певною тематикою або критеріями.

Поля:

- collection_id (PK, int) – унікальний ідентифікатор колекції;
- name (varchar) – назва колекції;
- description (text) – опис колекції;
- created_at (timestamp) – дата створення.

Таблиця «CollectionBooks» (Зв'язок книг і колекцій):

- collection_id (FK) – прив’язка до колекції;
- book_id (FK) – прив’язка до книги.

Таблиця «Ratings» (Оцінки користувачів) для зберігання рейтингових оцінок книг.

Поля:

- rating_id (PK, int) – унікальний ідентифікатор рейтингу;
- user_id (FK) – користувач, який поставив оцінку;
- book_id (FK) – книга, яка отримала оцінку;

- rating (int) – числова оцінка (наприклад, від 1 до 5);
- review (text) – відгук користувача;
- created_at (timestamp) – дата оцінювання.

Таблиця «Authors» (Автори) для зберігання даних про авторів книг.

Поля:

- author_id (PK, int) – унікальний ідентифікатор автора;
- name (varchar) – повне ім’я автора;
- biography (text) – коротка біографія автора;
- birth_date (date) – дата народження;
- death_date (date, nullable) – дата смерті (якщо застосовно).

Таблиця «BookAuthors» (Зв'язок книг і авторів):

- book_id (FK) – прив’язка до книги;
- author_id (FK) – прив’язка до автора.

Таблиця «Sessions» (Сесії користувачів) для зберігання активних сесій користувачів, що необхідно для авторизації.

Поля:

- session_id (PK, varchar) – унікальний ідентифікатор сесії;
- user_id (FK) – ідентифікатор користувача;
- ip_address (varchar) – IP-адреса користувача;
- user_agent (varchar) – дані про пристрій користувача;
- created_at (timestamp) – час створення сесії;
- expires_at (timestamp) – час закінчення сесії.

Інтеграція додаткових таблиць із основними виконується наступним чином: ключові слова та теги дозволяють покращити пошукову індексацію та рекомендації. Колекції та рейтинги покращують взаємодію з користувачем, додаючи елементи персоналізації. Автори надають більше інформації про контент бібліотеки. Сесії підвищують безпеку системи та забезпечують безперебійний досвід для користувачів.

Ці таблиці дозволяють розширити функціональність системи та підвищити її ефективність, враховуючи різні сценарії використання та вимоги.

Діаграму «сутність-зв’язок» представлено на рис. 3.4.

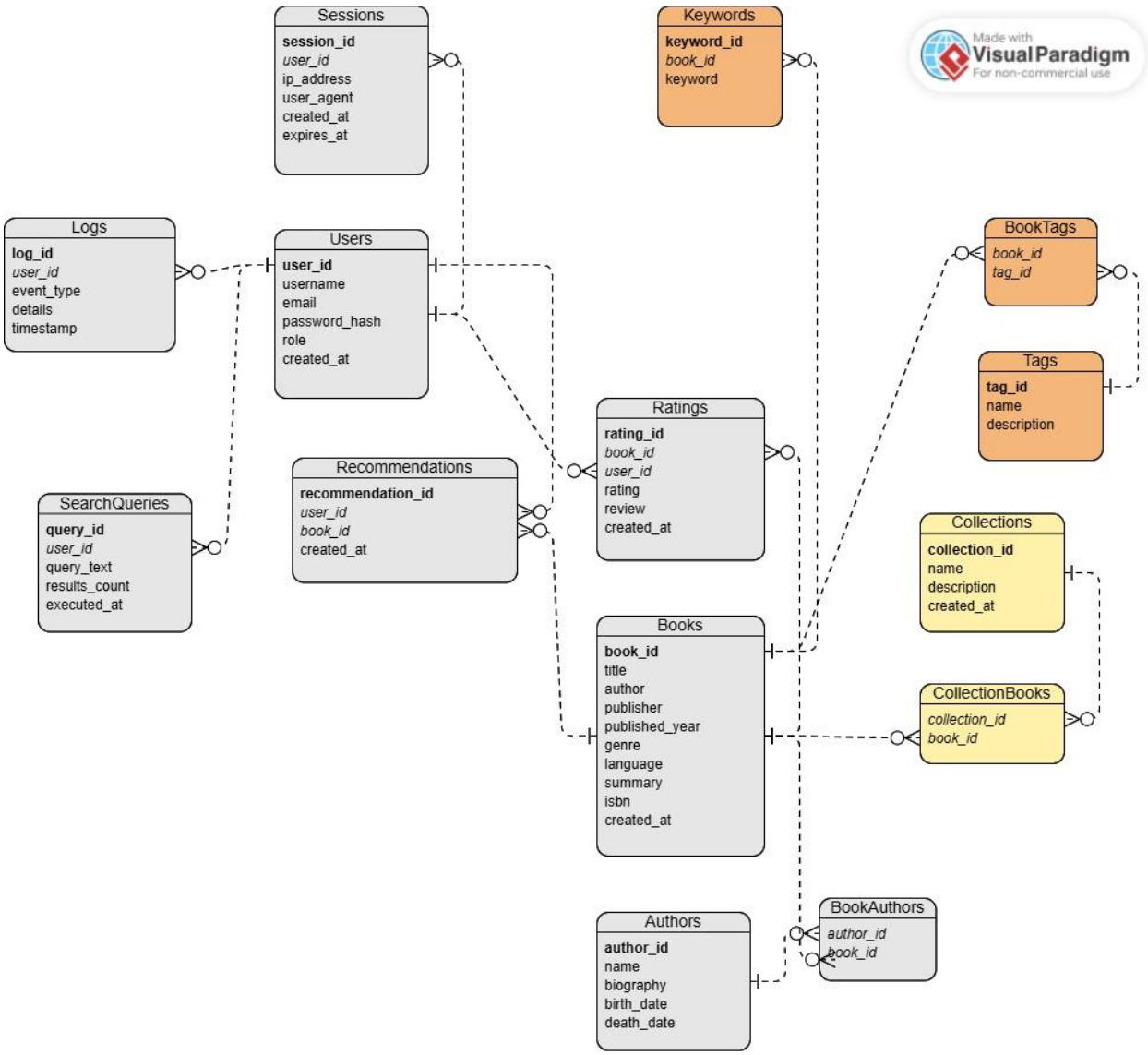

Рисунок 3.4 – ERD

3.5 Алгоритми роботи пошукової системи

Опис покрокового алгоритму роботи пошукової системи розділено на окремі сценарії відповідно до ключових компонентів: Frontend, Search API, NLP-сервіс, Recommendation API, Authentication Service. Це допомагає більш доступно відобразити процеси кожного компонента пошукової системи.

На діаграмі послідовності дії (рис. 3.5) представлено повний алгоритм роботи пошукової системи.

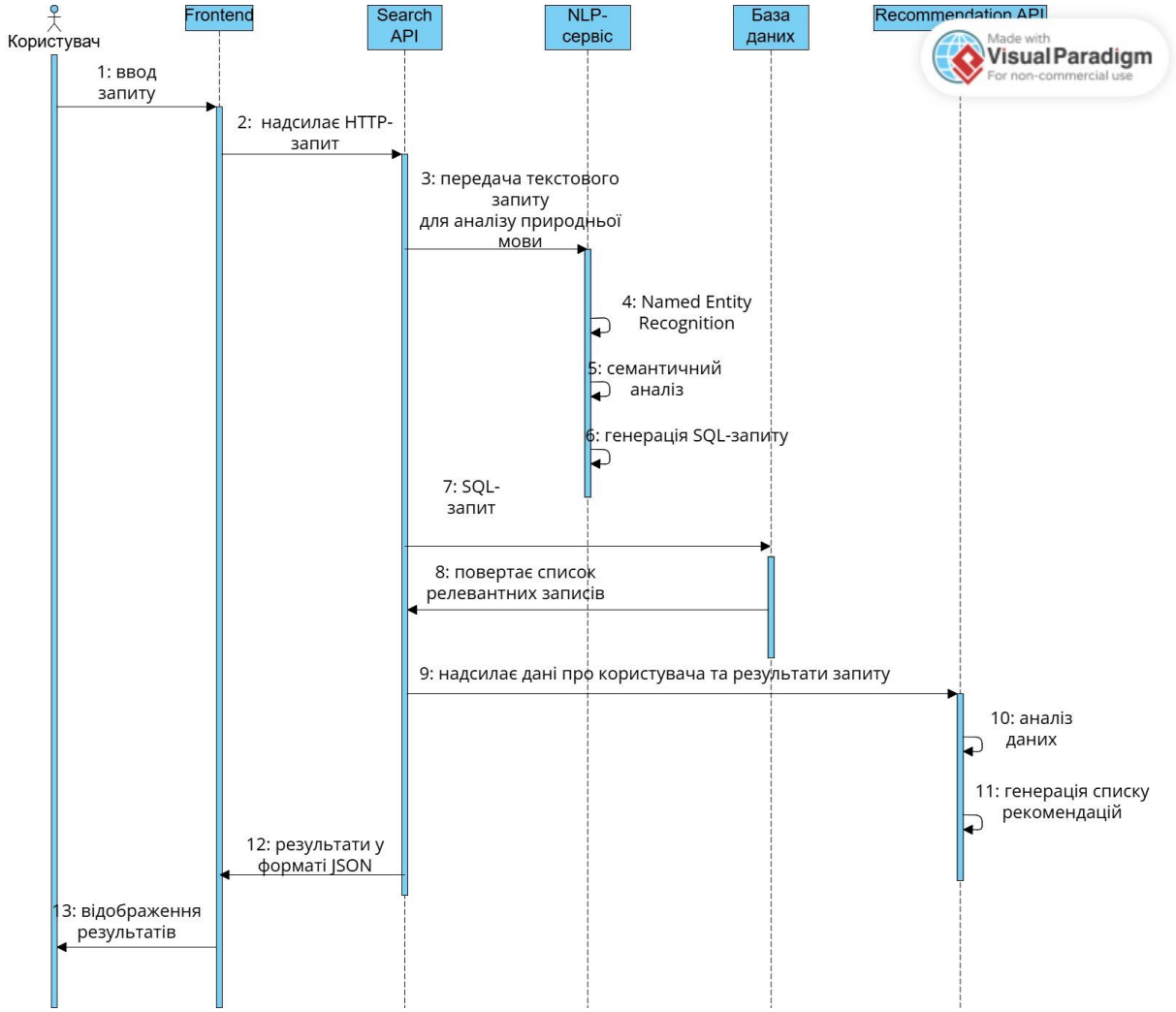

Рисунок 3.5 – Діаграма послідовності дії роботи пошукової системи

Розглянемо кожний крок більш детально.

Крок №1 «Frontend: Подача запиту користувачем».

1. Введення запиту:
   - користувач вводить текстовий запит у пошуковий рядок на веб-інтерфейсі;
   - опціонально користувач може вибрати фільтри (категорії, автори, рік публікації тощо).

2. Передача запиту: запит разом із фільтрами (якщо застосовано) відправляється на Search API через HTTP-запит.
3. Отримання результатів:
   - система отримує список книг, колекцій або рекомендацій;
   - результати відображаються у зручному вигляді на сторінці: із заголовками, авторами, короткими описами та кнопкою для більш детального перегляду.

Крок №2 «Search API: Обробка пошукового запиту».

1. Отримання запиту: API приймає HTTP-запит від Frontend із текстом запиту та параметрами.
2. Попередня обробка:
   - перевіряється формат та валідність запиту;
   - якщо потрібно, викликається Authentication Service для перевірки прав доступу користувача.
3. Звернення до NLP-сервісу: текстовий запит передається до NLP-сервісу для обробки природної мови (розпізнавання сутностей, семантичний аналіз тощо).
4. Пошук у базі даних:
   - на основі обробленого запиту генерується SQL-запит до бази даних;
   - результати повертаються у вигляді списку релевантних записів.
5. Обробка результатів:
   - якщо активна персоналізація, API викликає Recommendation API для додавання рекомендацій;
   - дані структуруються у зручному форматі JSON і повертаються на Frontend.

Крок №3 «NLP-сервіс: Обробка природної мови».

1. Аналіз запиту:
   - використовується модель BERT для аналізу тексту;

- виконується Named Entity Recognition (NER) для визначення ключових сутностей (імена авторів, жанри, теми тощо).

2. Семантичний аналіз:
   - визначається значення запиту за допомогою семантичного парсингу;
   - система може виправляти орфографічні помилки або уточнювати запит.
3. Генерація пошукового запиту: формується SQL-запит до бази даних, що відповідає змісту запиту користувача.
4. Повернення результатів до Search API.

Крок №4 «Recommendation API: Формування рекомендацій»

1. Отримання контексту: система приймає контекст пошуку (текстовий запит, результати пошуку, профіль користувача).
2. Персоналізований аналіз:
   - використовуються дані про попередні запити, оцінки, історію переглядів;
   - генерується список рекомендацій на основі схожості контенту (колекції, книги) з тим, що цікавить користувача.
3. Повернення рекомендацій: персоналізований список повертається до Search API.

Крок №5 «Authentication Service: Перевірка доступу».

1. Авторизація:
   - перевіряється токен доступу користувача;
   - визначаються права доступу (користувач, адміністратор).
2. Перевірка дій: оцінюється, чи має користувач доступ до запитаних даних (наприклад, преміум-контенту).
3. Повернення статусу:
   - у разі успіху надається доступ до запитаної функції.
   - у разі помилки відправляється відповідне повідомлення.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ПОШУКОВОЇ СИСТЕМИ

4.1 Вибір технологій розробки для Frontend і Backend

Ці інструменти були обрані завдяки їх сумісності, підтримці інтерактивного користувацького досвіду, швидкості обробки запитів і здатності працювати зі структурованими і неструктурованими даними, що критично для платформи із залученням користувачів та інформуванням їх у реальному часі.

React.js (для фронтенд) є одним із найпопулярніших фреймворків для створення динамічних веб-інтерфейсів завдяки його можливості компонування, швидкого рендерингу та великої кількості бібліотек. React.js дозволяє створювати інтерактивний інтерфейс, де компоненти оновлюються тільки за необхідності, що сприяє покращенню продуктивності. Інструмент добре інтегрується з іншими бібліотеками, наприклад, для управління станом або анімації, і особливо зручний для проектів, де передбачено розширення функціоналу.

React підходить для цього проекту з декілької причин:

1. Динамічний інтерфейс: React чудово справляється з динамічними компонентами, які швидко оновлюються, як у пошуковій системі.
2. Легка інтеграція API: React легко інтегрується з REST API або GraphQL, що підходить для отримання даних із пошукової системи та бази даних.
3. Масштабованість: компонентна структура дозволяє розширювати додаток без проблем.
4. Екосистема: React має великий набір бібліотек і інструментів (React Router, Redux, Context API), що робить розробку гнучкішою [14].

Для реалізації бекенду обрано Python та Django. Python – це високорівнева мова програмування, відома своєю простотою та багатою екосистемою. Django – це фреймворк для веб-розробки на Python, який забезпечує швидку розробку та масштабованість.

Головні переваги Python:

1. Зрозумілий синтаксис: Python легко читати та писати, що знижує час розробки.
2. Багатий набір бібліотек: Python має потужні бібліотеки для машинного навчання (TensorFlow, PyTorch, Scikit-learn), обробки тексту (NLTK, spaCy) та роботи з базами даних.
3. Масштабованість та універсальність: Python підходить як для швидких прототипів, так і для великих проектів.

До переваг Django слід віднести наступні:

1. «Batteries Included» (усе включено): Django надає вбудовані інструменти для створення повноцінного веб-додатка:
2. Аутентифікація користувачів.
3. ORM (Object-Relational Mapping) для роботи з базами даних.
4. Адміністративна панель.
5. Безпека: Django пропонує вбудовані захисти від таких атак, як CSRF, SQL-ін’єкції, XSS.
6. Масштабованість: завдяки своєму продуманому дизайну, Django дозволяє створювати масштабовані веб-додатки.
7. Підтримка REST API: Django REST Framework (DRF) полегшує створення RESTful API для взаємодії з клієнтським інтерфейсом та іншими компонентами системи.

4.2 Вибір СУБД

PostgreSQL – це потужна об’єктно-реляційна база даних з відкритим вихідним кодом, активна розробка якої триває понад 35 років, завдяки чому вона заслужила міцну репутацію надійності, надійності функцій і продуктивності.

PostgreSQL як реляційна база даних підтримує складні запити, що підходить для зберігання даних, пов’язаних між собою, таких як користувачі,

автори, рукописи та відгуки. PostgreSQL відзначається стабільністю, безпекою, великим функціоналом для масштабованих систем і можливістю працювати з великими обсягами даних. Завдяки своїй надійності, ефективності та підтримці розширених функцій, PostgreSQL активно використовується в корпоративних рішеннях, наукових дослідженнях та розробці веб-додатків [15].

Основні особливості PostgreSQL:

1. Сумісність з SQL-стандартом.

PostgreSQL відповідає сучасним стандартам SQL і має підтримку розширень, що дозволяють використовувати додаткові мови запитів та функції для складної аналітики.

2. Масштабованість та ефективність.

Підтримує як горизонтальне, так і вертикальне масштабування, що дозволяє обробляти великі обсяги даних. Його структура оптимізована для роботи з індексами, що прискорює операції читання та запису.

3. Розширення та користувацькі типи даних.

PostgreSQL дозволяє створювати власні типи даних, зберігати JSON, працювати з географічними об'єктами (через PostGIS) і забезпечує повну підтримку індексації для таких типів.

4. Система транзакцій: реалізація ACID-принципів гарантує надійність транзакцій, а механізм MVCC (Multi-Version Concurrency Control) дозволяє одночасно працювати з базою без блокувань і з мінімальними затримками.

5. Безпека та управління доступом.

Підтримує аутентифікацію через SSL, сертифікати, а також пропонує можливості шифрування даних та доступу на основі ролей.

6. Інтеграція з іншими мовами та системами.

PostgreSQL підтримує такі мови програмування, як Python, Java, C, та інші, що дозволяє інтегрувати його в різноманітні екосистеми, а також використовувати API для взаємодії з різними додатками.

7. Масивне співтовариство та підтримка: завдяки активному співтовариству, існує багато розширень і бібліотек, які підвищують функціональність та допомагають у вирішенні різних задач [12].

Завдяки цьому PostgreSQL є універсальним рішенням для багатьох типів проектів, від простих додатків до систем високої доступності, які обробляють великі обсяги даних.

Використання JWT (JSON Web Token) корисне для аутентифікації в онлайн-сервісі, оскільки він дозволяє зручно передавати зашифровані дані користувача без необхідності постійного зберігання сесій на сервері.

4.3 Інструменти реалізації NLP

4.4.1 Бібліотека для обробки тексту та синтаксичного аналізу

spaCy – це сучасна бібліотека Python для обробки природної мови (NLP), яка надає швидкі, надійні та зручні інструменти для роботи з текстами. Вона орієнтована на виробниче використання, підтримує багатомовність і має високу продуктивність [15].

Основні функції та можливості spaCy:

1. Токенізація:
   - розбиває текст на окремі слова або речення (токени);
   - підтримує спеціальні токенайзери для різних мов.
2. Лематизація:
   - приводить слова до базової форми (наприклад, "running" -> "run");
   - дозволяє зменшити варіативність тексту для пошуку та аналізу.
3. Частини мови (POS-тегінг): розпізнає граматичні ролі слів (іменник, дієслово, прикметник тощо).
4. Розпізнавання сутностей (NER): виділяє важливі сутності, такі як дати, імена, локації, організації. Наприклад, у запиті «Книги Джорджа Орвелла про майбутнє» NER визначить «Джордж Орвелл2 як особу.

5. Синтаксичний аналіз (Dependency Parsing) включає сттворення дерева залежностей між словами для визначення їх ролей у реченні.
6. Семантичний аналіз включає векторизацію слів через попередньо навчені моделі.
7. Мовна підтримка містить попередньо навчені моделі для понад 20 мов.
8. Інтеграція з іншими інструментами: підтримує кастомізацію моделей та інтеграцію з TensorFlow, PyTorch, та бібліотекою Hugging Face.

4.4.2 Бібліотека для моделювання семантичного пошуку

Sentence-BERT – це модифікація моделі BERT (Bidirectional Encoder Representations from Transformers), створена для ефективного обчислення семантичної близькості між реченнями. На відміну від стандартного BERT, який не підходить для обчислення схожості через значні обчислювальні витрати, Sentence-BERT значно прискорює цей процес [15].

SBERT додає шар об'єднання (pooling layer) поверх BERT, щоб отримувати вектори для повних речень. Ці вектори можна використовувати для:

- порівняння семантичної схожості між реченнями;
- кластеризації текстів;
- пошуку найбільш релевантного тексту з бази даних.

SBERT використовує модифікований BERT і додаткові нейронні шари для генерації реченнєвих векторів. Вектори мають фіксовану розмірність і можуть використовуватись у задачах порівняння через евклідову відстань або косинусну схожість.

SBERT навчається на задачах типу "natural language inference" (NLI), таких як визначення, чи є два речення схожими за змістом. В результаті модель отримує здатність порівнювати семантичну близькість текстів.

Основні функції Sentence-BERT:

1. Обчислення семантичної схожості: модель ефективно порівнює значення речень, навіть якщо слова у реченнях відрізняються.
2. Пошук за змістом: SBERT може використовуватися для пошуку релевантних текстів у великій базі даних на основі змісту запиту.
3. Кластеризація тексту: завдяки компактним векторним представленням текстів SBERT полегшує групування схожих документів.
4. Прискорення обчислень: на відміну від стандартного BERT, який потребує багаторазової обробки текстів, SBERT дозволяє попередньо обчислити вектори для текстів і зберігати їх для швидких порівнянь.

Обрати SBERT для бібліотечної системи слід з наступних причин:

- покращення пошуку: SBERT дозволяє реалізувати пошук, орієнтований на значення запиту, а не лише точний збіг слів;
- швидкість: підходить для обробки запитів у великих базах даних, оскільки вектори речень можна обчислити заздалегідь;
- інтеграція з існуючими базами: SBERT генерує вектори, які можна використовувати з інструментами пошуку (наприклад, Elasticsearch, Faiss).

4.4.3 Бібліотека для рекомендаційних систем

ElasticSearch – це потужна система для пошуку та аналітики, що базується на відкритому коді. Вона побудована на основі Lucene і дозволяє ефективно індексувати та здійснювати пошук у великих обсягах структурованих і неструктурованих даних. Особливості ElasticSearch:

1. Швидкість пошуку та індексації: підтримує індексацію даних у реальному часі та швидко обробляє пошукові запити завдяки розподіленій архітектурі.

2. Масштабованість: підходить для обробки великих даних завдяки можливості масштабування шляхом додавання вузлів у кластер.
3. Повнотекстовий пошук:
   - пошук за релевантністю (relevance-based search);
   - морфологічний пошук;
   - синоніми, стоп-слова та стемінг.
4. Інтеграція: легко інтегрується з різними інструментами для аналітики (наприклад, Kibana), підтримує API для Python, Java, JavaScript, Node.js та інших мов програмування.
5. Гнучкість: працює з даними будь-якого формату: JSON, XML, текстові документи.

4.5 Програмна реалізація основних функцій

Аналізом запитів займається spaCy, тож на першому кроці слід підклбчити цю бібліотеку до файлу. За замовчуванням використовуємо модель англійської мови:

```
import spacy
nlp = spacy.load("en_core_web_sm")
```

Наступний крок, це сама обробка тестового текстового запиту:

```
query = "Find books by George Orwell about dystopia"
doc = nlp(query)
```

Далі йде розпізнавання сутностей, частини мови та лематизація:

```
    print("Named Entities:", [(ent.text, ent.label_) for ent in
doc.ents])
    print("POS Tags:", [(token.text, token.pos_) for token in
doc])
    print("Lemmas:", [token.lemma_ for token in doc])
```

Результати свідчать про коректну роботу цього модуля:

```
Named Entities: [('George Orwell', 'PERSON')]
POS     Tags:[('Find','VERB'),('books','NOUN'),('by','ADP'),
...]
Lemmas: ['find', 'book', 'by', 'George', 'Orwell', 'about',
'dystopia']
```

Для впровадження пошуку та рекомендаційної системи у бібліотечних проектах використовується Sentence-BERT. Спочатку необхідно завантажити модель та протестувати її на прикладі запиту користувача:

```
from sentence_transformers import SentenceTransformer, util
model = SentenceTransformer('all-MiniLM-L6-v2')
query = "Find books about Shakespeare"
```

Після цього, система формує колекцію документів:

```
documents = [
    "A biography of William Shakespeare",
    "Historical analysis of 16th-century literature",
    "Guide to Shakespearean plays and poems",
    "Introduction to English literature"
]
```

Наступний крок: генерація векторів та пошук найбільш схожих текстів:

```
query_embedding = model.encode(query,
convert_to_tensor=True)
doc_embeddings = model.encode(documents,
convert_to_tensor=True)
cosine_scores = util.cos_sim(query_embedding,
doc_embeddings)

# Виведення результатів
for i, score in enumerate(cosine_scores[0]):
    print(f"Document: {documents[i]}, Similarity:
{score.item():.4f}")
```

5 ТЕСТУВАННЯ РОБОТИ ПОШУКОВОЇ СИСТЕМИ

5.1 Функціональне тестування

Функціональне тестування охоплює різні види тестування, спрямовані на перевірку правильності роботи функцій, описаних у вимогах до системи. Мета такого тестування – переконатися, що система виконує всі передбачені функції й відповідає очікуванням користувачів. Тестування в даному випадку буде грунтуватися на варіантах використання системи (описаних у другому розділі роботи).

Тест-кейс – це документ, що містить послідовність дій, які виконує тестувальник, щоб перевірити певну функціональність або властивість програмного забезпечення. Тест-кейс включає в себе: назву, передумови, які мають бути виконані перед запуском тесту, кроки виконання, які необхідно виконати для проведення тестування та очікувані результати.

За варіантами використання було складено 11 тест-кейсів (ТС – test-cases), які охвачують основні функціональні вимоги до пошукової системи електронної бібліотеки:

- Тест-кейси для функціонального тестування пошукової системи (ТС01-ТС04);
- Додаткові функціональні тест-кейси (ТС05-ТС08);
- Тест-кейси для UX/UI (ТС09);
- Тест-кейси для обробки помилок (ТС10-ТС11).

ТС01 «Виконання пошуку книг за ключовими словами»

Передумови:

1. Користувач зареєстрований та увійшов до системи.
2. У базі даних містяться книги з відповідними ключовими словами.

Основні кроки:

1. Відкрити головну сторінку системи.
2. У поле пошуку ввести ключові слова, наприклад, "математика, алгоритми".

3. Натиснути кнопку "Пошук".

Очікуваний результат:

1. Система повертає список книг, релевантних до введених ключових слів.
2. Відображаються назви книг, автори, жанри та дати публікації.

ТС02 «Фільтрація результатів пошуку»

Передумови:

1. У базі даних містяться книги з різними жанрами («роман», «наукова фантастика», «навчальна література»).
2. Користувач виконав пошук книг.

Основні кроки:

1. Виконати пошук за ключовими словами.
2. У списку результатів пошуку вибрати жанр "наукова фантастика" у фільтрах.

Очікуваний результат: результати оновлюються, відображаються лише книги відповідного жанру.

ТС03 «Отримання рекомендацій на основі історії пошуку»

Передумови:

1. Користувач виконав декілька пошукових запитів.
2. Рекомендаційна система активна.

Основні кроки:

1. Виконати кілька пошукових запитів.
2. Перейти до розділу "Рекомендації".

Очікуваний результат: система пропонує список книг, релевантних історії пошуку користувача.

ТС04 «Вибір джерела інформації»

Передумови:

1. У системі зареєстровано декілька мов представленої літератури (українська, англійська).
2. Користувач зайшов до пошукової системи.

Основні кроки:

1. У фільтрі пошуку обрати потрібну мову джерела інформації.
2. Підтвердити параметри пошуку.

Очікуваний результат: система відображає перелік літератури обраною мовою та яка відповідає ключовому слову запиту. У разі відсутності такого джерела, система висвічує повідомлення щодо рекомендації обрати іншу мову.

ТС05 «Перевірка пошуку за автором»

Передумови: у базі даних є книги певного автора.

Основні кроки:

1. У полі пошуку ввести ім'я автора.
2. Натиснути кнопку «Пошук».

Очікуваний результат: відображається список книг цього автора.

ТС06 «Перевірка коректності обробки запитів за допомогою NLP»

Передумови:

1. У системі активовано модуль обробки природної мови (NLP).
2. Користувач увів запит природною мовою.

Основні кроки:

1. У полі пошуку ввести запит типу: «Книги про програмування для початківців».
2. Натиснути кнопку «Пошук».

Очікуваний результат: система коректно обробляє запит та відображає релевантні книги.

ТС07 «Обробка некоректного пошукового запиту».

Передумова: користувач увійшов до системи.

Основні кроки:

1. У полі пошуку ввести некоректний або незрозумілий запит ("!@^&34*).
2. Натиснути кнопку «Пошук».

Очікуваний результат: система відображає повідомлення: «Запит не містить зрозумілих ключових слів» або пропонує допомогу у формулюванні запиту.

ТС08 «Перевірка функції сортування результатів пошуку».

Передумови: виконано пошук, і результати містять кілька книг.

Основні кроки:

1. Виконати пошук за ключовими словами.
2. Вибрати параметр сортування, наприклад, «За датою публікації» або «За популярністю».

Очікуваний результат: результати пошуку сортуються відповідно до вибраного критерію.

ТС09 «Тестування інтуїтивності інтерфейсу пошуку».

Передумова: користувач увійшов до системи.

Основні кроки:

1. Виконати пошук без додаткових інструкцій.
2. Перевірити, чи зрозумілий інтерфейс фільтрів і сортування.

Очікуваний результат: користувач може легко знайти потрібну інформацію, навіть якщо вперше використовує систему.

ТС10 «Обробка помилки при недоступності NLP-сервісу».

Передумова: NLP-сервіс тимчасово недоступний.

Основний крок: Виконати пошуковий запит.

Очікуваний результат: система відображає повідомлення: 2Вибачте, обробка запиту зараз недоступна. Спробуйте пізніше».

ТС11 «Відображення повідомлення при відсутності результатів пошуку».

Передумова: база даних не містить книг, які відповідають запиту.

Основні кроки: у полі пошуку ввести запит, якого немає у базі даних, наприклад, «книга, якої не існує».

Очікуваний результат: система відображає повідомлення: «Нічого не знайдено. Спробуйте змінити запит.»

5.2 Експериментальне тестування

Експериментальне тестування пошукової системи передбачає перевірку її роботи на реальних даних та сценаріїв використання. Це дозволяє оцінити ефективність, зручність, точність пошуку та поведінку системи під навантаженням.

Стартова сторінка електронної бібліотеки представлено на рис. 5.1. У хедері розміщено вкладки «Рекомендації», «Список побажань». Після авторизації ім’я користувача відображається у верхньому лівому куту.

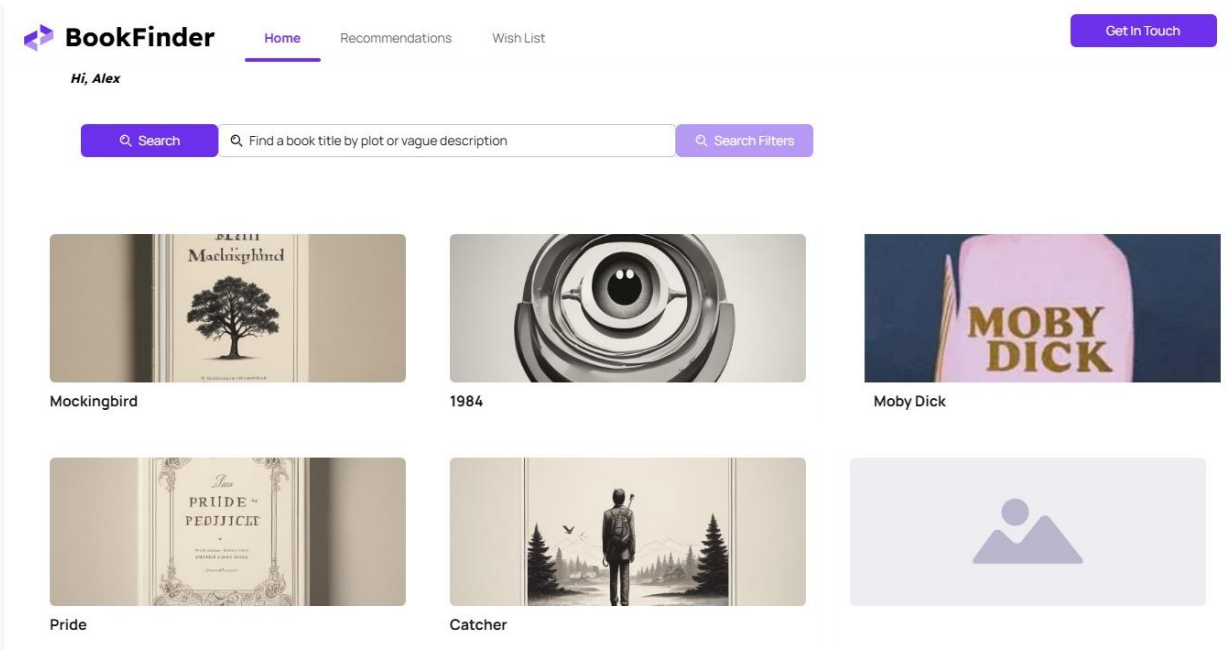

Рисунок 5.1 – Стартова сторінка електронної бібліотеки

Одна з основних задач даного ресурсу – це пошук інформації. Користувач може ввести у поля назву книги або описати тематику, до якої належить книга (чи інше джерело інформації). Як результат роботи пошуку, після натискання кнопки «Search» система повинна вивести список наявної літератури за вказаним описом.

Для прикладу було введено у пошукову строку запит «Web application development». Результат пошуку представлено на рис. 5.2.

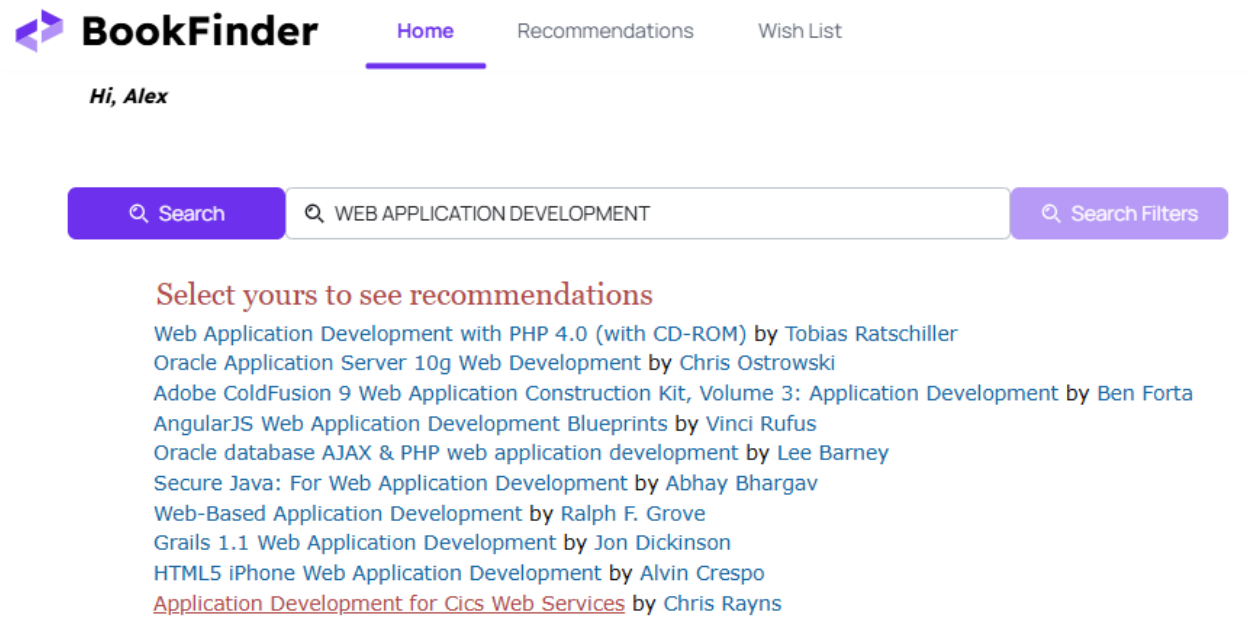

Рисунок 5.1 – Скріншот результату виводу інформації за пошуковим запитом

Кожнй запис представляє посиланнння на електроний варіант джерела інформації. Якщо користувачу необхідно відсортувати джерела за роком виданння, чи призвіщем автора, слід використовувати кнопку «Search Filters» (рис. 5.3).

Рисунок 5.3 – Перелік фільтрів для сортування джерел інформації

На приклад, користувача цікавить автор Simon Stobart, його ім’я слід ввести у відповідне поле фільтру. Також користувач може обрати якою мовою повинні бути джерела. Після підтвердження пошукова система виводить

перелік книг цього автора, які є в її базі даних (рис. 5.4 а). З урахуванням того, що система працює з NLP, для тестування слід ввести ім’я автора, наприклад, українською абеткою та з маленькими помилками. Результат роботи системи у такому випадку представлено на рис. 5.4 б.

а)

б)

Рисунок 5.4 – Тестування роботи пошукових фільтрів

Остання перевірка – це випадок, коли користувач вводить незрозумілий для пошукової системи запит (набор символів). Система повинна висвітити відповідне повідомлення (рис. 5.5).

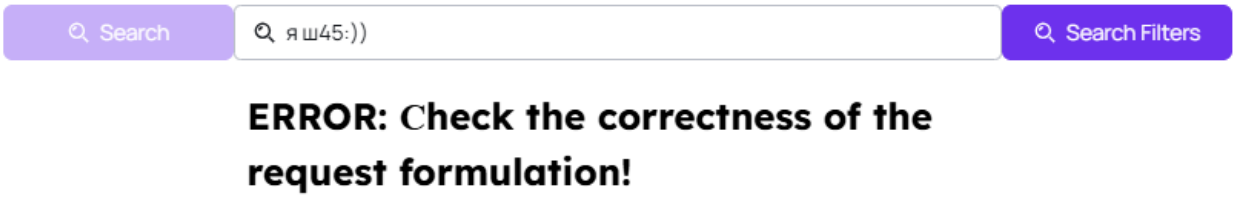

Рисунок 5.5 – Приклад реакції системи на некоректний запит

ВИСНОВКИ

Обробка природної мови (NLP) є ключовим компонентом сучасних систем, які створюють персоналізовані рекомендації, здійснюють пошук і організовують інформацію, включаючи книги й колекції в електронних бібліотеках. Використання NLP дозволяє системі аналізувати текстові запити користувачів, розуміти їхні наміри, контекст, і навіть передбачати їхні потреби на основі історії взаємодій.

Під час розробки та оптимізації роботи пошукової системи бібліотеки з елементами машинного навчання було досліджено предметну область та існуючі програмні рішення. Проаналізовані алгоритми роботи таких пошукових систем, технології та інструменти для реалізації NLP у бібліотечних системах. Обрані інструментальні засоби розробки, проведене проектування архітектури та основних алгоритмів роботи її компонентів. На останньому етапі робот проведене тестування пошукової системи, результати якого свідчать про досягнення поставленої мети.

Електронні бібліотеки дозволяють користувачам отримувати доступ до широкого спектру матеріалів з будь-якого місця та в будь-який час, маючи лише бібліотечну картку та підключення до Інтернету. Ця цілодобова доступність відповідає різноманітним графікам користувачів бібліотеки, роблячи матеріали для читання доступними, коли вони потрібні.

Трансформація бібліотек за допомогою цифрового дизайну та штучного інтелекту полягає не лише в тому, щоб йти в ногу з технологічним прогресом; йдеться про переосмислення того, якими можуть бути бібліотеки в епоху цифрових технологій. Оскільки бібліотеки продовжують адаптуватися та розвиватися, вони знову підтверджують свою роль незамінних ресурсів громади, пропонуючи не лише доступ до інформації, але й шляхи до знань, навчання та зростання. Бібліотеки продовжуватимуть впроваджувати інновації, надихати та освітлювати шлях у майбутнє.

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. The Impact of Emerging Technologies on Libraries. URL: https://www.collectionhq.com/the-impact-of-emerging-technologies-on-libraries/#:~:text=In%20recent%20years%2C%20many%20libraries,is%20accessible%20in%20their%20branch (дата звернення 15.07.2024)

2. The Future of Libraries: Navigating Digital Transformations. URL: https://princh.com/blog-the-future-of-libraries-navigating-digital-transformations/ (дата звернення 15.07.2024)

3. How digital resources at libraries can create a better experience for patrons. URL: https://blog.pressreader.com/libraries-institutions/how-digital-resources-at-libraries-can-create-a-better-experience-for-patrons (дата звернення 15.07.2024)

4. Що таке електронна бібліотека? Її основні задачі та особливості. URL: https://elect-library.blogspot.com/2017/03/blog-post_10.html (дата звернення 22.08.2024)

5. About EPrints – Digital Library NAES of Ukraine. URL: https://lib.iitta.gov.ua/eprints/ (дата звернення 22.08.2024)

6. The European Library is now in Europeana. URL: https://www.europeana.eu/en/TEL(дата звернення 22.08.2024)

7. Digital Library. URL: https://link.springer.com/chapter/10.1007/978-3-662-05300-3_11 (дата звернення 06.09.2024)

8. Greenstone: Open-Source Digital Library Software. URL: https://www.dlib.org/dlib/october01/witten/10witten.html (дата звернення 06.09.2024)

9. What Is NLP (Natural Language Processing)? URL: https://www.ibm.com/topics/natural-language-processing (дата звернення 22.09.2024)

10. Detailed view of BERT. URL: https://pub.aimind.so/detailed-view-of-bert-ecc98d50a8fd (дата звернення 06.10.2024)

11.BERT language model. What is BERT? URL: https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model (дата звернення 06.10.2024)

12.Top 5 Pre-trained Word Embeddings. URL: https://patil-aakanksha.medium.com/top-5-pre-trained-word-embeddings-20de114bc26 (дата звернення 15.10.2024)

13.What Is Named Entity Recognition (NER) and How It Works? URL: https://www.altexsoft.com/blog/named-entity-recognition/ (дата звернення 24.10.2024)

14.React – A JavaScript library for building user interfaces. URL: https://legacy.reactjs.org/ (дата звернення 02.11.2024)

15.PostgreSQL: The world's most advanced open source database. URL: https://www.postgresql.org/ (дата звернення 02.11.2024)