МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Кудінов Федір Валерійович

«Алгоритми адаптивного автоматичного пошуку та збору наукової інформації / Algorithms for adaptive automatic search and gathering of scientific information»

спеціальність: 123 - Комп'ютерна інженерія
освітньо-професійна програма - Комп'ютерна інженерія
Кваліфікаційна робота

Виконав студент групи КІм-21
Ф.В. Кудінов
_____

Науковий керівник:
д.т.н., О.М. Березький
_____

Кваліфікаційну роботу допущено
до захисту:

"___" _____ 20___ р.

Завідувач кафедри
_____Л. О. Дубчак

Тернопіль – 2023

РЕЗЮМЕ

Кваліфікаційна робота на тему «Алгоритми адаптивного автоматичного пошуку та збору наукової інформації / Algorithms for adaptive automatic search and gathering of scientific information» зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «магістр» написана обсягом 103 сторінки і містить 38 ілюстрацій, 4 таблиці, 2 додатки та 52 джерела за переліком посилань.

Метою роботи є розроблення моделей пошуку інформації та адаптивної системи пошуку та збору наукової інформації.

Методи досліджень базуються на використанні теорії алгоритмів, методів обробки природної мови та об'єктно-орієнтованого програмування.

Розроблено булеву модель пошуку інформації, програмно реалізовано пошукову систему наукової інформації.

Результати роботи можуть бути використані при побудові інтелектуальних систем пошуку інформації.

Можливими напрямками подальших досліджень є : розробка алгоритмів пошуку інформації та побудова адаптивних систем її біометричних зображень.

КЛЮЧОВІ СЛОВА: МОДЕЛЬ, ПОШУК ІНФОРМАЦІЇ, АЛГОРИТМ, АДАПТИВНА ПОШУКОВА СИСТЕМА.

RESUME

The qualification work on the theme "Algorithms for adaptive automatic search and gathering of scientific information" from specialty 123 "Computer engineering" of the master's degree is written in the volume of 103 pages and contains 38 illustrations, 4 tables , 2 appendices and 52 sources according to the list of references.

The purpose of the work is to develop information search models and an adaptive system for searching and collecting scientific information.

Research methods are based on the use of the theory of algorithms, methods of natural language processing and object-oriented programming.

Information search models, a software search system for scientific information have been developed.

The results of the work can be used in the construction of intelligent information retrieval systems.

Possible trends of further research are: development of information search algorithms and construction of adaptive systems and biometric images.

KEY WORDS: MODEL, INFORMATION SEARCH, ALGORITHM, ADAPTIVE SEARCH SYSTEM.

ЗМІСТ

ВСТУП

Актуальність теми.

Для науковців створили наукометричні бази. Це дуже важливо для опублікування наукових праць науковців різних країн і різних галузей знань.

Існують відкриті джерела, які доступні всім та закриті джерела. Знайомство з роботами науковців гальмується закриттям публікацій в журналах. Це привело до створення штучних бар'єрів у обміні інформацією. Модератори відомих світових наукових баз на інформації науковців роблять неабиякий бізнес. У двадцять першому столітті це недопустимо. Обмін науковою інформацією повинен бути миттєвим, крім галузей державної безпеки та оборони країн. Інформація на даний час швидко старіє. Штучне стримування вільним обміном науковою інформацією приводить до старіння інформації та утримання темпу науково-технічного прогресу

Основною метою наукометричних баз є збір, організація та аналіз наукової інформації з метою вивчення та вимірювання наукового впливу, визначення актуальних тем досліджень, та підтримки прийняття рішень в галузі науки та досліджень.

Тому пошук інформації з відкритих джерел є актуальною задачею.

У даному дослідженні використано праці викладачів кафедри комп'ютерної інженерії Західноукраїнського національного університету із штучного інтелекту [1-10].

Мета і завдання дослідження. Метою роботи є розроблення моделей пошуку інформації та адаптивної системи пошуку та збору наукової інформації.

Об'єкт дослідження – процес пошуку інформації .

Предмет дослідження - алгоритми та моделі пошуку інформації.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- проаналізувати відомі наукометричні бази даних;
- проаналізувати пошукові системи наукометричних баз даних;
- проаналізувати моделі пошуку інформації;

– розробити булеву модель пошуку інформації;

– розробити адаптивну пошукову систему для пошуку та формування наукової інформації.

Методи досліджень базуються на використанні теорії алгоритмів, методів обробки природної мови та об’єктно-орієнтованого програмування.

Розроблено булеву модель пошуку інформації, програмно реалізовано пошукову систему наукової інформації.

Наукова новизна одержаних результатів. Розроблено булеву модель пошуку інформації.

Практичне значення отриманих результатів. Програмно реалізовано пошукову систему наукової інформації.

Публікації та апробація КР. За результатами кваліфікаційної роботи опубліковані двоє тез доповідей на VIII-й науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі», яка відбулася на кафедрі комп’ютерної інженерії Західноукраїнського національного університету, 5 грудня 2023 р., м. Тернопіль [11, 12].

Кваліфікаційна робота складається із трьох розділів, висновків, списку використаної літератури та додатків [13].

У першому розділі проведено аналіз наукометричних баз даних: великих сховищ наукових публікацій, цитувань тощо. Проаналізовано такі агрегатори повних текстів: Scopus, Web of Science, IEEE Xplore, Google Scholar тощо. Також проаналізовано методику та етапи пошуку інформації в наукометричних базах даних.

У другому розділі описані класичні моделі пошуку інформації.

У третьому розділі описано розроблену адаптивну систему пошуку та формування наукових публікацій.

У додатках приведено, довідку про використання результатів кваліфікаційної роботи, світлокопії виданих публікацій.

1 АНАЛІЗ НАУКОМЕТРИЧНИХ БАЗ ДАНИХ І МЕТОДІВ ПОШУКУ ІНФОРМАЦІЇ

1.1 Аналіз наукометричних баз даних

Наукометричні бази даних є важливою складовою сучасного наукового інформаційного середовища. Ці бази представляють собою великі сховища наукових публікацій, цитувань та інших даних, пов'язаних з науковими дослідженнями. Основною метою наукометричних баз є збір, організація та аналіз наукової інформації з метою вивчення та вимірювання наукового впливу, визначення актуальних тем досліджень, та підтримки прийняття рішень в галузі науки та досліджень.

Наукометричні бази можуть включати наукові журнали, конференційні матеріали, патенти, тези докторських дисертацій та інші види наукової інформації. Основними характеристиками наукометричних баз є індексація, цитування, імпакт-фактори та інші метрики, які допомагають визначити значення та вплив наукових публікацій та досліджень.

Цитування вказує на те, наскільки часто стаття або дослідження були використані процитовані іншими науковцями. Дані про цитування дозволяють визначити впливовість і значення конкретних робіт у науковій спільноті. Індекс цитування статті можна перевірити за допомогою різних наукових баз даних та інструментів, таких як: Google Scholar, Web of Science, Scopus.

Індекс цитування [14] важливий для авторів, оскільки він може впливати на їхню репутацію та кар'єру. Для університетів та дослідницьких інститутів індекси цитування їхніх співробітників можуть впливати на рейтинг та фінансування. Хоча індекс цитування може бути корисним індикатором, він має обмеження. Наприклад, нові статті можуть мати низький індекс цитування через короткий час від їх публікації, а статті в менш популярних галузях можуть мати систематично нижчі індекси цитування незалежно від їх якості.

Імпакт-фактор [15] є метрикою, яка визначає впливовість наукового журналу. Він вимірює середню кількість цитат, отриманих журналом за певний

період, на кількість опублікованих статей у цьому журналі. Імпакт-фактор використовується для оцінки престижу та якості журналу, а також впливу публікацій, опублікованих у цьому журналі. Він розраховується як середнє число цитувань статей, опублікованих у журналі протягом певного періоду часу:

$$\text{Імпакт-фактор} = \frac{\text{Кількість цитувань статей, опублікованих у журналі за останні два роки}}{\text{Загальна кількість статей, опублікованих у журналі за останні два роки}}$$

До інших аналітичних інструментів можуть входити h-індекс, i10-індекс, а також аналіз цитувань та графи співавторств. Графи співавторства («Co-authorship graphs", "Collaboration networks") є важливим інструментом у соціометрії та бібліометрії, які використовуються для візуалізації та аналізу співпраці між науковцями. Ось основні риси графів співавторства:

1. Вузли (nodes) зазвичай представляють авторів наукових робіт.
2. Ребра (edges) показують зв'язки між авторами. Якщо два автори опублікували роботу разом, між їх вузлами буде ребро.
3. Структура графа (Graph structure") може відображати різні мережеві властивості, такі як кластеризація (групи авторів, які часто співпрацюють один з одним), центральність (автори, які є ключовими в мережі співавторства) та ізоляція (автори, які рідко співпрацюють з іншими).
4. Графи співавторства можуть використовуватися для аналізу мережевих властивостей, таких як ступінь централізації, щільність мережі та шляхи зв'язків між авторами.
5. Візуалізація графів співавторства може допомогти у визначенні ключових авторів у певній області, виявленні спільнот чи кластерів в мережі та розумінні динаміки наукової співпраці.
6. Графи співавторства використовуються в академічних дослідженнях для оцінки впливовості вчених, аналізу тенденцій співпраці у різних областях науки та вивчення еволюції наукових спільнот.

Бази даних та пошукові системи суттєво відрізняються одна від одної за охопленням та якістю отримання інформації. Важливо враховувати якості та обмеження баз даних та пошукових систем, особливо тих, які використовуються для систематичного пошуку записів, наприклад, в систематичних оглядах або метааналізах [16]. Різниця між базою даних і пошуковою системою (search engine) для цих складних систем витягнення документів є нечіткою.

Метадані в наукових базах даних включають інформацію, яка описує та категоризує наукові статті. Вони можуть включати:

– повну назву наукової роботи;

– імена та афіліації авторів статті;

– анотацію - Короткий опис вмісту статті;

– ключові слова, які визначають основні теми статті;

– назву журналу або конференції де була опублікована стаття.

– дату публікації;

– Унікальний цифровий ідентифікатор статті DOI (Digital Object Identifier);

– інформацію про цитування, тобто про інші роботи, які цитують цю статтю.

Індексація в наукових базах даних зазвичай виконується за цими метаданими, що дозволяє користувачам ефективно шукати та знаходити статті, які відповідають їхнім інтересам чи потребам дослідження.

Наукометричні бази можна класифікувати за різними ознаками, включаючи:

– тип джерел: деякі бази спеціалізуються на наукових журналах, інші на конференційних матеріалах, а треті можуть включати патенти, дисертації, технічні звіти та інші види наукової інформації;

– покриття: різні бази можуть охоплювати різні галузі науки та географічні регіони. Наприклад, деякі спеціалізуються на природничих науках, тоді як інші орієнтовані на гуманітарні дисципліни;

– тип доступу: деякі бази є відкритими і доступними безкоштовно для загального користувача, тоді як інші можуть бути комерційними та доступними за плату або підпискою.

Архітектура наукометричних баз може варіюватися в залежності від постачальника, але зазвичай вони мають такі основні компоненти:

–база даних зберігання наукової інформації, яке включає в себе публікації, цитати, метрики та інші дані;

–інтерфейс користувача дозволяє користувачам виконувати пошук, аналіз та інші дії з даними в базі;

–аналітичні інструменти представляють функціонал для аналізу та вимірювання наукового впливу, включаючи імпакт-фактори, h-індекси, графи цитувань і багато іншого.

1.2 Аналіз відомих баз даних наукових публікацій

Серед основних наукових баз даних, пошукових систем, агрегаторів повних текстів можна виділити такі, як Scopus, Web of Science, IEEE Xplore, Google Scholar, Internet Archive Scholar, CORE, CiteSeerX, Semantic Scholar, Europe PMC, PubMed Central (PMC), DBLP. Ці сервіси надають доступ до повних текстів та пошуку повного тексту, а також метаданих для об'єктів, повний текст яких недоступний [16]. По кількості представлених повних наукових текстів серед агрегаторів [16] лідерами є Internet Archive Scholar з 25 млн. записів, CORE з 9,8 млн. записів та CiteSeerX з 8,4 млн. Серед баз що надають пошук по метаданих лідерами є Google Scholar з 389 млн. записів, Dimensions з 302 млн. записів, AMiner з 365 млн. записів. Кількість записів на публікації в Web of Science оцінюється і 171 млн., Scopus 78 млн.

Scopus – це база даних анотацій та цитувань видавництва Elsevier, запущена в 2004 р. Scopus охоплює майже 36 377 видань [17]. Усі журнали, що входять до бази даних Scopus, щороку перевіряються на досить високу якість відповідно до чотирьох типів числових показників якості для кожної назви: h-індекс, CiteScore, SJR (SCImago Journal Rank) і SNIP (джерело нормалізованого впливу на статтю).

Scopus надає такі засоби пошуку і фільтрації результатів [18]:

1) пошук документів: безпосередньо з головної сторінки можна використати і детальні параметри пошуку;

2) пошук конкретного автора за іменем або за ідентифікатором відкритих досліджень та авторів (ORCID);

3) пошук за афіліацією. Можна визначити та оцінити наукові результати афілійованої організації, установи, що співпрацюють з нею, та провідних авторів.

4) розширений пошук. Сферу пошуку можна звузити за допомогою кодів полів (field codes), операторів близькості (proximity operators) та/або булевих операторів

Розширений пошук в Scopus використовує візуальні елементи. Можна використати булеві оператори AND, OR, NOT для одного чи декількох полів (рисунок 1.1). Результати ошуку можна фільтрувати за часом видання, типом доступу, полями предметною галуззю [19].

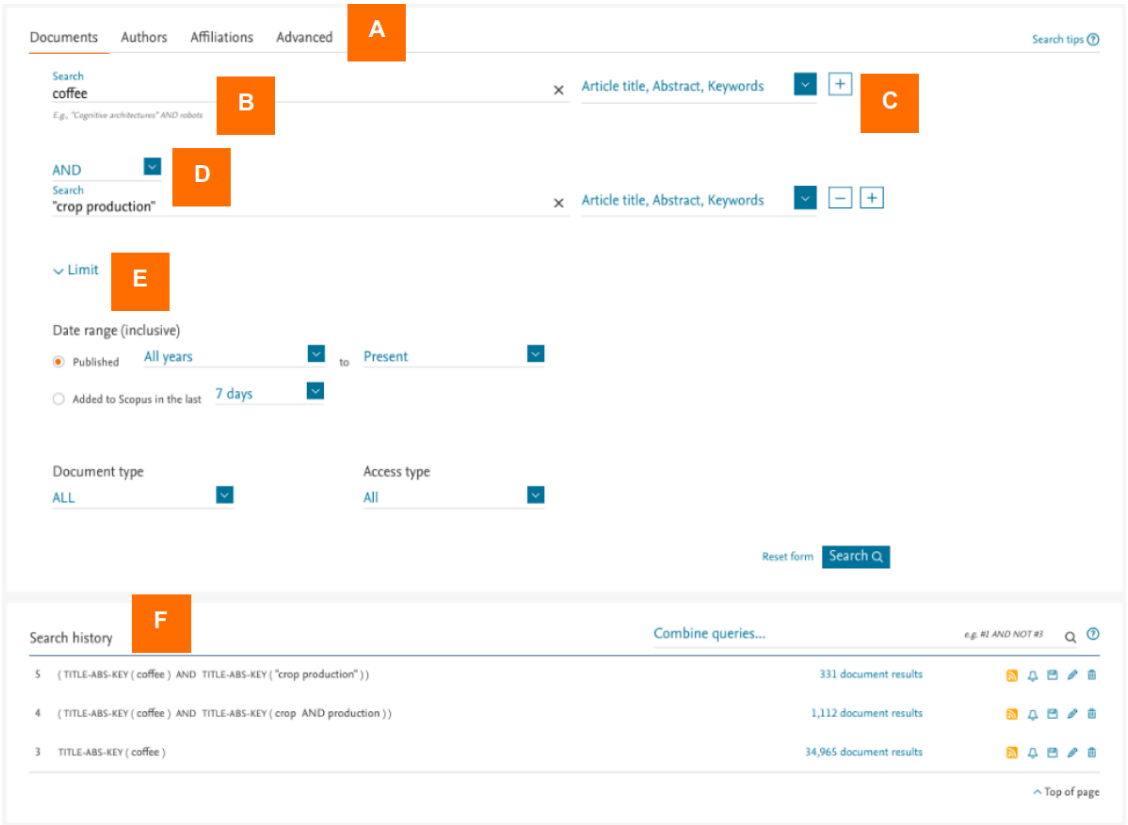


Рисунок 1.1 – Розширений пошук Scopus

Можна шукати в усіх базах даних Web of Science (Основна колекція) або у вибраних базах даних. Пошук за ключовим словом у таких полях, як тема (topic), назва, автор, назва публікації (publication name) та публікація(publication), резюме (abstract).

База даних Web of Science також дозволяє пошук за допомогою комбінації пошукових полів і булевих операторів (рисунок 1.2). На рисунку показано комбінований пошук по декількох частинах назви публікації та по періоду видання.

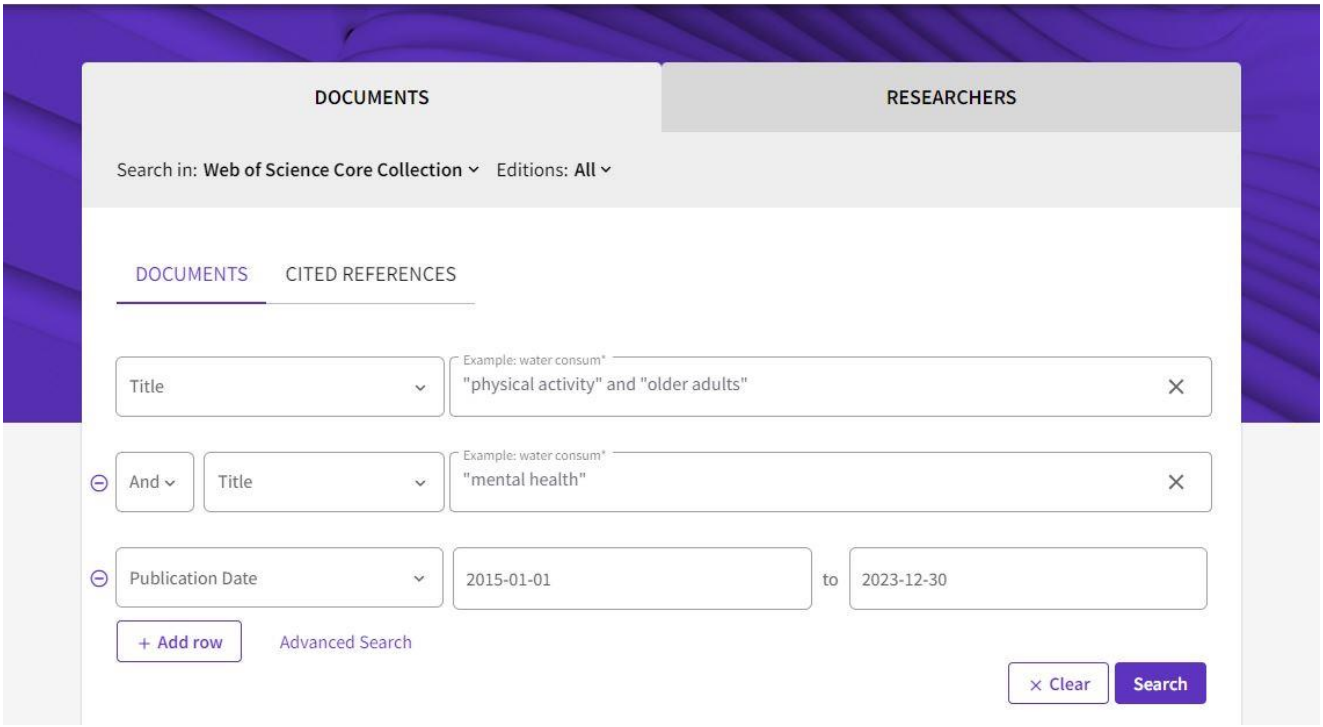


Рисунок 1.2 – Розширений пошук Web of Science

Можна використати оператори близькості, щоб обмежити результати пошуку записами, в яких задані пошукові терміни знаходяться в межах заданої кількості слів один від одного:

1) NEAR/# знаходить записи, в яких пошукові терміни знаходяться в межах # слів один від одного. NEAR без /# за замовчуванням обмежує пошук до 15 слів.

2) SAME використовується лише в пошуку за адресою (Address) і обмежує результати пошуку термінами, розташованими за тією самою адресою в записі.

Результати пошуку показані на рисунку 1.3.

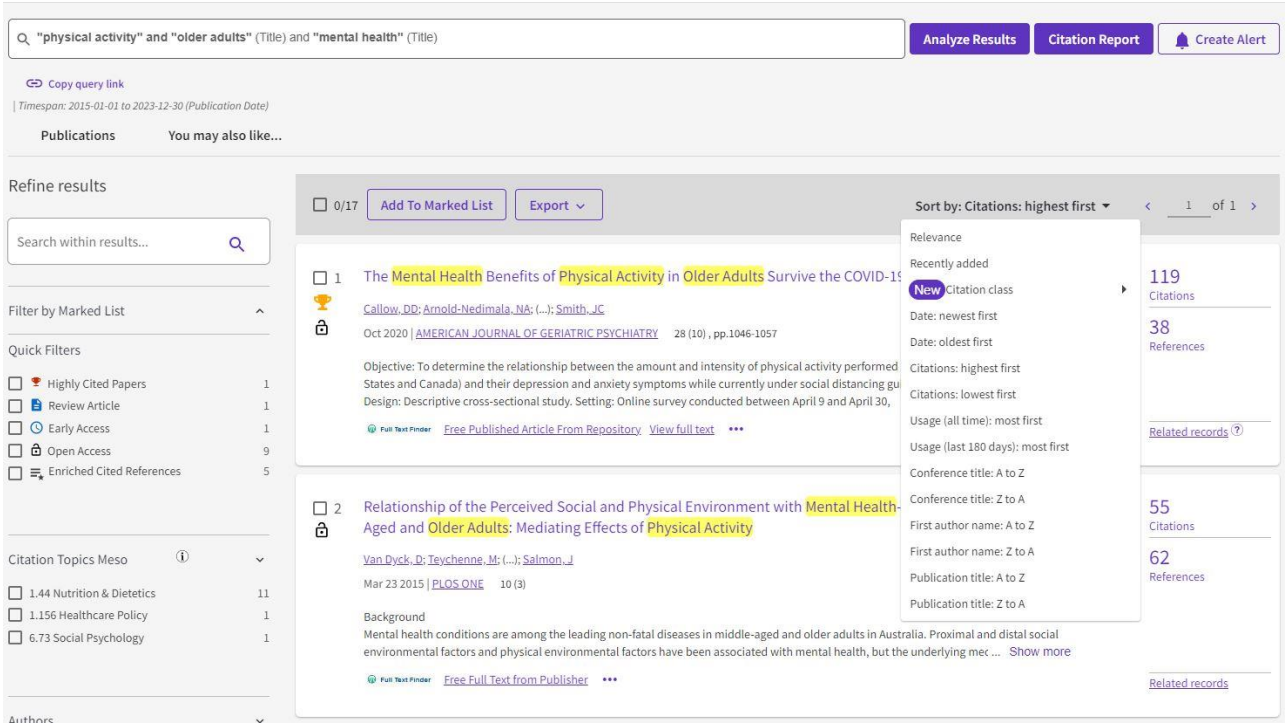


Рисунок 1.3 – Результати пошуку Web of Science

Спільні риси платформ:

1. Усі платформи надають доступ до широкого спектру наукових матеріалів, таких як журнальні статті, конференційні матеріали, книги, дисертації тощо.

2. Усі платформи пропонують розширені функції пошуку, які дозволяють користувачам шукати літературу за різними критеріями (ключові слова, автори, назви, рік публікації тощо).

3. Всі ці ресурси орієнтовані на академічну аудиторію, включаючи дослідників, студентів, викладачів та інших фахівців.

Відмінні риси:

1. Scopus та Web of Science:
   - вважаються одними з найавторитетніших наукометричних баз;
   - надають детальні бібліометричні аналізи та індекси цитувань;
   - вимагають підписки для повного доступу.
2. IEEE Xplore:

– спеціалізується на літературі з електротехніки, комп'ютерних наук та електроніки;

– доступ до публікацій IEEE та інших технічних організацій.

3. Google Scholar:

– безкоштовний та зручний у використанні;

– охоплює широкий спектр дисциплін, але часто не включає детальні бібліометричні дані.

4. Internet Archive Scholar, CORE, CiteSeerX:

– вільно доступні ресурси;

– надають доступ до великої кількості відкритих та безкоштовних наукових ресурсів.

5. Semantic Scholar:

– використовує штучний інтелект для поліпшення пошукових можливостей;

– додаткові інструменти для аналізу трендів та зв'язків між публікаціями.

6. Europe PMC, PubMed Central (PMC):

– є американським архівом безкоштовних повних текстів медичних та біологічних журнальних статей.

7. DBLP:

– фокусується на літературі з комп'ютерних наук;

– надає індексацію конференцій, журналів, та інших публікацій у галузі IT.

Агрегатор Internet Archive Scholar [20] містить понад 35 мільйонів наукових статей та інших наукових документів. Колекція охоплює цифрові копії журналів XVIII століття та найновіші матеріали конференцій з вільним доступом. Метадані беруться з fatcat.wiki, відкритого каталогу, який користувачі можуть редагувати, і який містить інформацію про наукові роботи.

Google Scholar пропонує широкий спектр академічних ресурсів з різних дисциплін.

Основні функції наукові бази даних:

– доступ до академічних журналів та статей;

– пошук за ключовими словами та авторами;

– повнотекстовий пошук та доступ (або безкоштовний доступдо резюме, чи декількох перших сторінок);

– метадані та індексація: надання метаданих, які допомагають в ідентифікації та класифікації статей.

IEEE Xplore Digital Library це це наукова база даних для пошуку та доступу до журнальних статей, матеріалів конференцій, технічних стандартів та супутніх матеріалів з інформатики, електротехніки та електроніки, а також суміжних галузей. Чи не основним елементом головної сторінки є стрічка пошуку (рисунок 1.4).

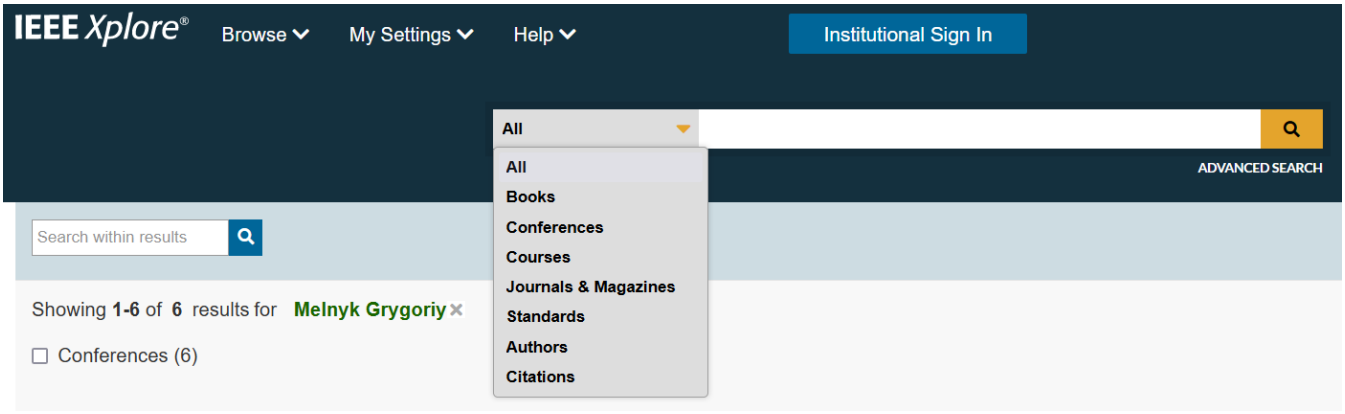
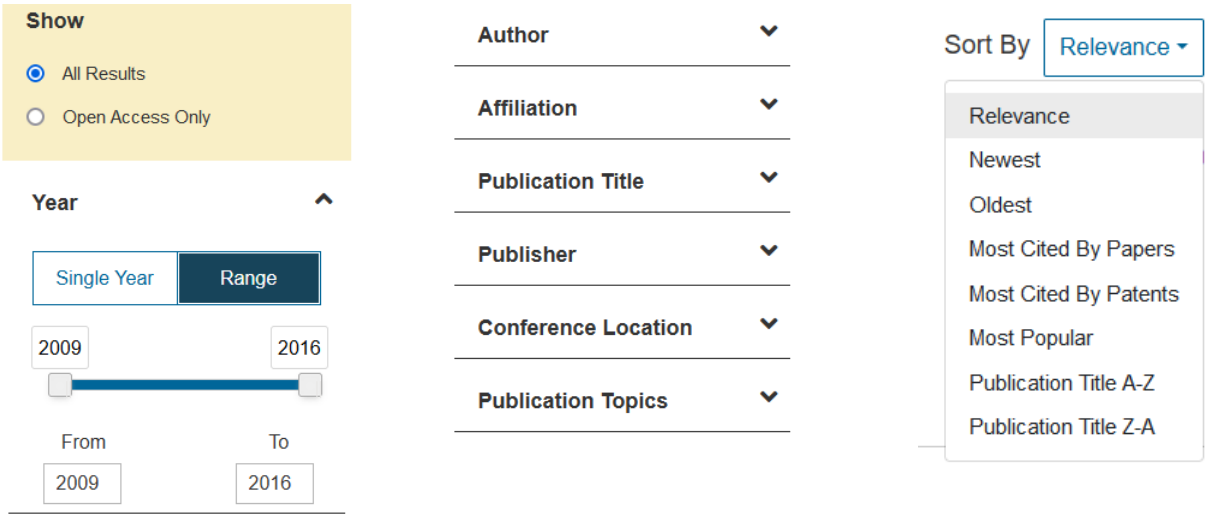


Рисунок 1.4 – Пошук в IEEE Xplore

Результатом пошуку є сторінка результатів – детальна інформація про знайдені публікації з посиланнями на всі елементи самої публікації. Фільтрація та сортування результатів пошуку на сторінці показана на рисунку 1.5.



Фільтрація Фільтрація Сортування

Рисунок 1.5 – Фільтрація видачі платформи IEEE Xplore

Короткий опис полів даних IEEE Explore. Поля даних ідентифікують певні частини запису документа. Обмеживши пошук певним полем (або метаданими), можна скоротити час, необхідний для обробки пошуку, і отримати більш точні результати. В таблиці 1.1 наведено поля даних, які можна використовувати в розширеному пошуку та/або пошуку за командами.

Таблиця 1.1 – Поля метаданих наукових публікацій IEEE Explore

Ім’я поля	Визначення
Резюме Abstract	Коротке резюме або виклад змісту журнальної статті, доповіді на конференції, стандарту, книги, розділу книги або курсу.
Accession Number	Порядковий номер, який присвоюється кожному запису або тому при додаванні до бази даних.
Article Number	Унікальний номер запису, присвоєний статті.
Article Page Number	Позиція журнальної статті в журналі, що пропонується, наприклад, номер сторінки у випуску журналу.
Author Affiliation	Інституційна приналежність авторів, зазначена в документах (університет, державна установа, корпорація тощо).
Author Keywords	Терміни, надані автором, які описують теми або предмети документа.
Author ORCID	ORCID (Open Researcher and Contributor ID) – це некомерційна організація, яка допомагає унікально ідентифікувати дослідників і авторів, щоб залишатися на зв'язку з їхнім внеском і зв'язками.
Authors	Ім'я автора або авторів, зазначених у документі.

Продовження таблиці 1.1

Document Title	Назва окремого документа (журнальна стаття, доповідь на конференції, стандарт, розділ книги або курс).
DOI	Цифровий ідентифікатор об'єкта. Унікальний рядок символів для ідентифікації окремого об'єкта, наприклад, журнальної статті або доповіді на конференції.
Full Text & Metadata	Повний текст – це текст роботи, статті, стандарту тощо. Метадані – це детальна інформація, яка описує повний текст, наприклад, імена авторів, дата публікації та DOI.
Funding Agency	Назва або ідентифікатор організації, яка надала фінансування або грант на наукове дослідження.
IEEE Terms	Ключові слова, присвоєні статтям журналів IEEE та доповідям конференцій з контрольованого словника, створеного IEEE.
Index Terms	Комбіноване поле, яке дозволяє користувачам здійснювати пошук за ключовими словами автора (Author Keywords), термінами IEEE (IEEE Terms) та сітковими термінами (Mesh Terms).
ISBN	Міжнародний стандартний номер книги (International Standard Book Number). Номер, який використовується для унікальної ідентифікації книги або неперіодичного видання
ISSN	Міжнародний стандартний серійний номер (International Standard Serial Number). 8-значний номер, який використовується для унікальної ідентифікації періодичного видання (журналу або серії).
Issue	Номер випуску журналу, в якому була опублікована стаття.
Metadata	Включає анотацію, індексні терміни та дані бібліографічного посилання (наприклад, назву документа, назву публікації, автора тощо).
MeSH Terms	Медичні предметні рубрики, визначені Національною медичною бібліотекою (NLM). Терміни MeSH інтегровані в IEEE Xplore для 14 біомедичних видань IEEE.
Publication Number	Унікальний номер запису, присвоєний публікації.
Publisher	Видавничі організації, які мають контент у цифровій бібліотеці IEEE Xplore.
Publication Title	Назва публікації (журнал, конференція, книга).
Standard Number	Позначення стандарту (наприклад, IEEE 802.11u-2011). Позначення стандартів присвоюються Адміністратором Комітету з нових стандартів Ради зі стандартів IEEE-SA (NesCom).
Standard Dictionary Terms	Терміни, включені до глосарію стандарту та словника стандартів IEEE.
Standards ICS Terms	Таксономія термінів ICS [22] (Міжнародна класифікація стандартів) (International Classification for Standards)

Платформа забезпечує функцію розширеного пошуку по вказаних полях з допомогою логічних операторів. Можна використовувати як візуальний інтерфейс (рисунок 1.6) так і запис команд «Command search». Наприклад, спробуємо знайти статті пов’язані з IoT в збірнику чи журналі по темі штучного інтелекту (рисунок 1.7).

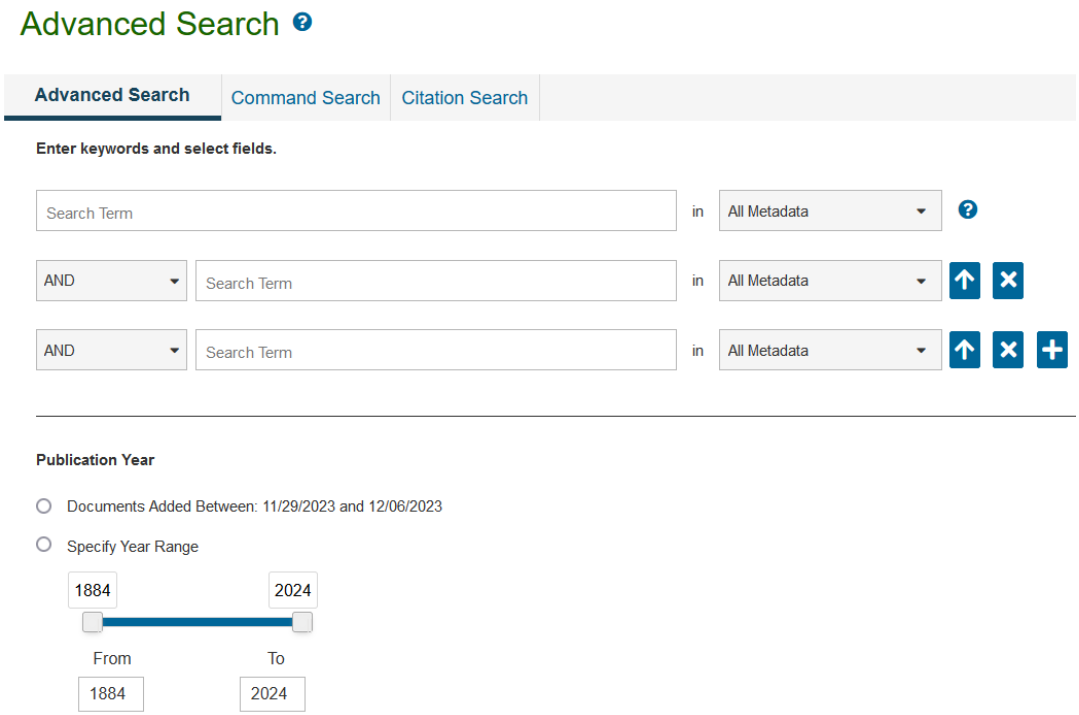


Рисунок 1.6 – Розширений пошук IEEE Xplore

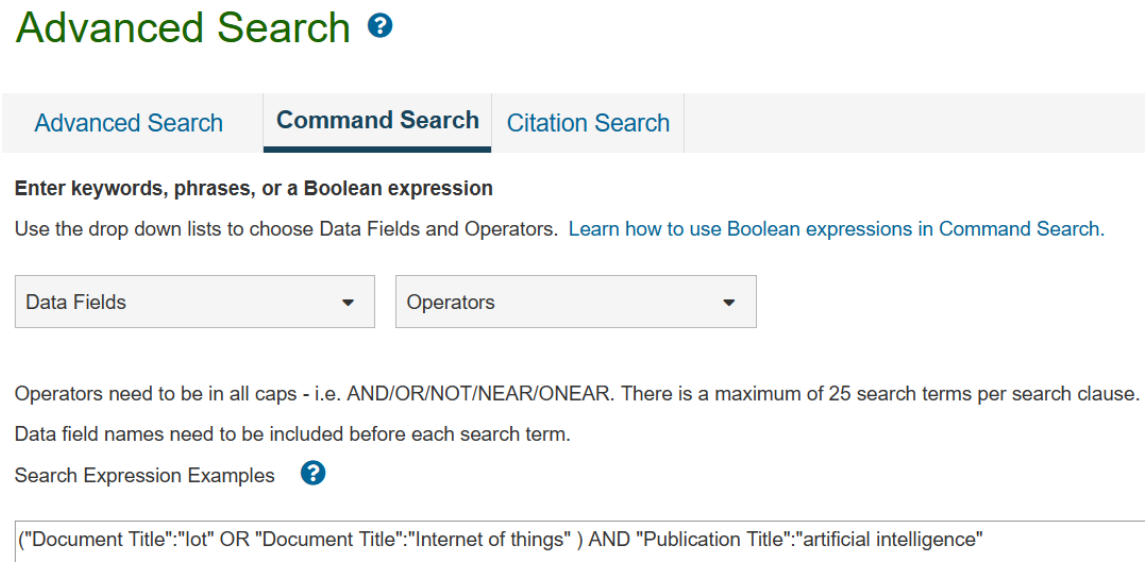


Рисунок 1.7 – Командний пошук IEEE Xplore

Результат структурного пошуку зображено на рисунку 1.8.

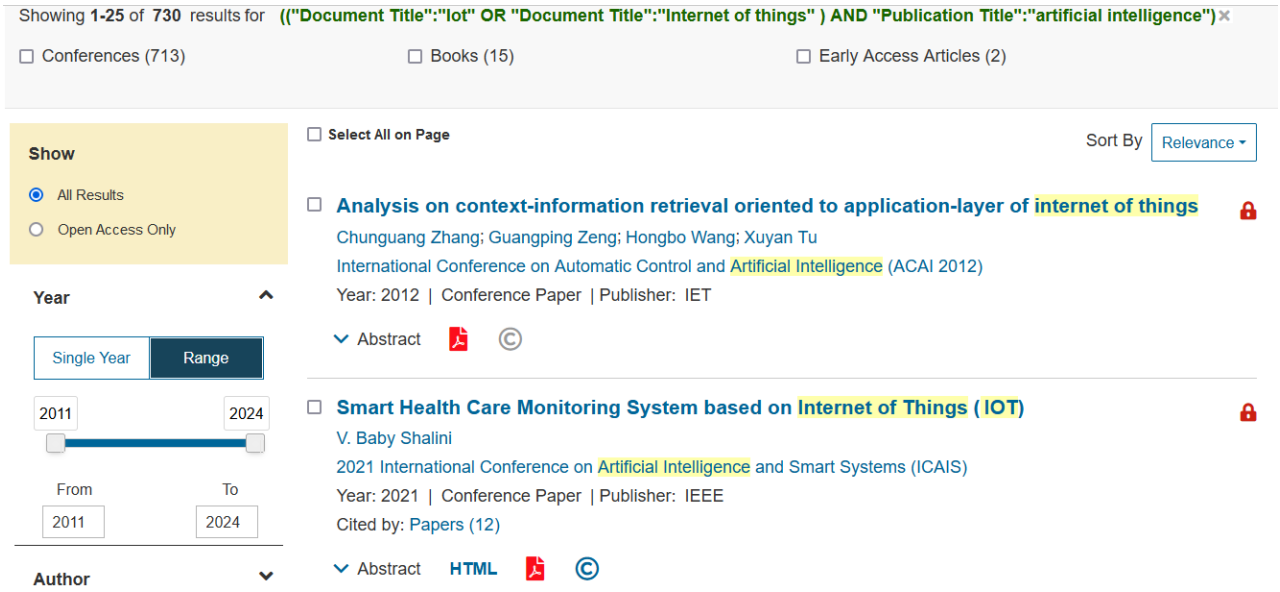


Рисунок 1.8 – Результат структурного пошуку

Оператори пошуку – це елементи, які виражають зв'язки між пошуковими термінами або пошуковими виразами, або які іншим чином модифікують запит. У структурованому розширеному пошуку можна використовувати логічні оператори AND, OR і NOT. У командному пошуку можна використовувати також оператори пошуку слів по їх позиції в тексті: NEAR (для невпорядкованого пошуку по близькості) і ONEAR (для впорядкованого пошуку по близькості) [21].

Semantic Scholar — це безкоштовна, інтелектуальна пошукова система для наукових публікацій, розроблена Allen Institute for AI. Її головна мета — поліпшити спосіб, яким дослідники знаходять релевантні інформаційні матеріали для своїх робіт. Semantic Scholar використовує передові технології ШІ та машинного навчання для аналізу наукових робіт, що дозволяє виявити важливі цитати, зображення, графіки, авторів та ключові поняття. Semantic Scholar допомагає візуалізувати зв'язки між різними науковими роботами, показуючи, як публікації впливають одна на одну через цитування та спільні теми дослідження (рисунок 1.9).

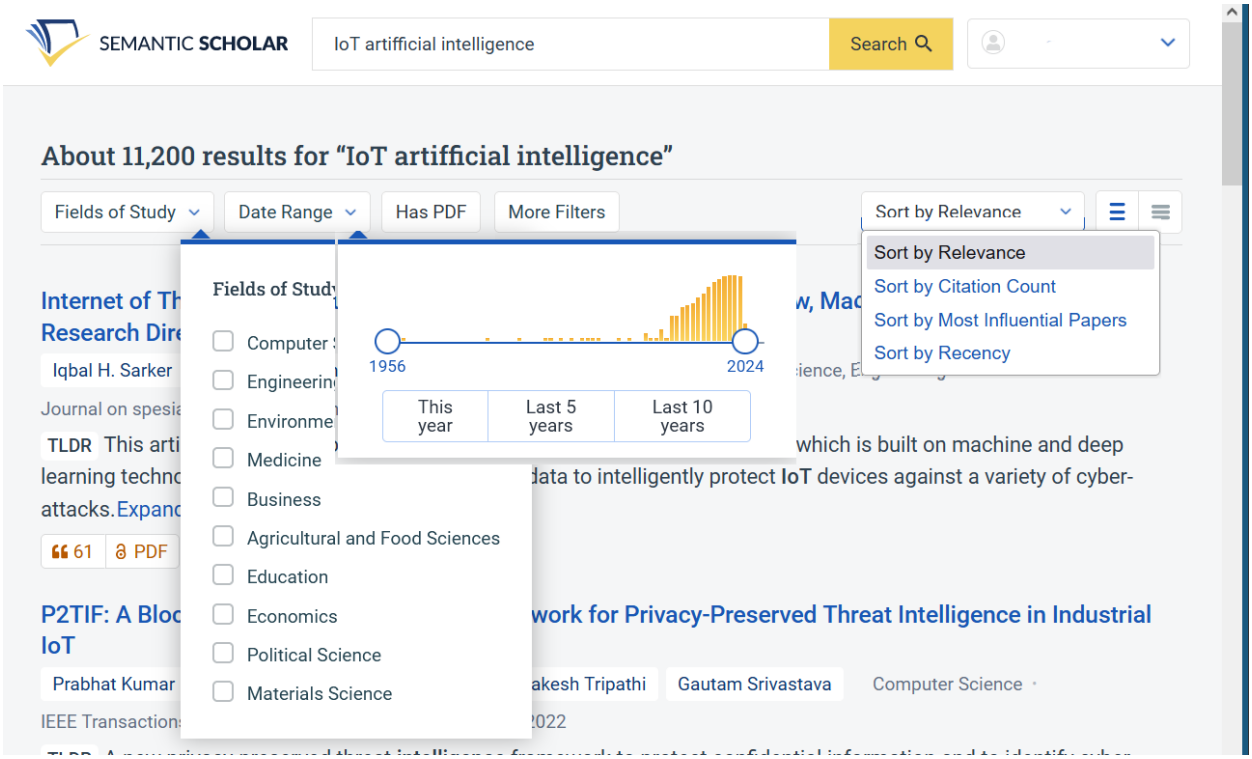


Рисунок 1.9 – Вікно пошуку Semantic Scholar

Для кожного знайденого джерела Semantic scholar пропонує надзвичайно інформативну сторінку результату (рисунок 1.10). Зокрема крім обов'язкових полів, наведено і посилання на документи зі списку джерел і посилання на статті що цитують дану і список «пов'язаних» статей за даною тематикою. Можна одразу завантажити статтю або створити бібліографічний запис.

Пошуковик, використовує розширені методи аналізу даних для категоризації цитувань на основі їхньої ролі чи значення в науковому дослідженні. Категорія високого впливову цитати (Highly Influential citations) [23] показує наскільки важливою стала дана стаття для подальших досліджень.

Цитування Background Citations (Фонова Цитата) відноситься до тих, що забезпечують контекст чи фонову інформацію для дослідження. Такі цитати часто включають посилання на роботи, які описують загальну область дослідження, теоретичні основи, або попередні дослідження, які формують основу для поточного дослідження.

Methods Citations (Цитування Методології) стосуються конкретних методів, технік або процедур, які були використані в дослідженні. Вони можуть

включати посилання на роботи, де описані експериментальні методики, аналітичні підходи, алгоритми, інструменти тощо.

Рисунок 1.10 – Сторінка результату

Цитування Results Citations (Цитати Результатів) включають посилання на роботи, що мають прямий зв'язок із результатами даного дослідження. Наприклад, попередні дослідження, які підтверджують або спростовують результати поточного дослідження, або дослідження, які демонструють подібні висновки.

Функція розширеного пошуку в Google Scholar (Google Академія) є хорошим інструментом для детального та спеціалізованого пошуку наукових матеріалів. Основні її функції:

1. «Пошук за точними фразами» дозволяє шукати точні фрази, використовуючи лапки. те.

2. Функція «Виключення слів» дозволяє виключити певні слова з пошукового запиту, для звуження результатів.

3. «Пошук за автором» дозволяє шукати роботи конкретних авторів.

4. «Пошук у певних журналах» дозволяє обмежити пошук публікаціями з конкретних журналів.

5. «Дата публікації» дозволяє вибрати певний період часу, протягом якого були опубліковані дослідження, що вас цікавлять.

6. «Пошук в назві чи тексті» дозволяє обмежувати пошук окремими частинами публікації.

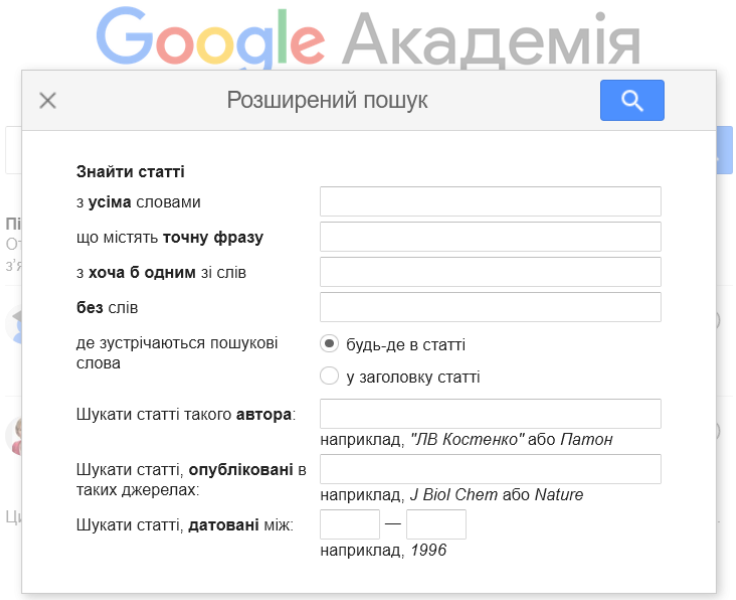


Рисунок 1.11 – Пошук в Гугл Академії

Сторінка результатів пошуку дозволяє проводити сортування за релевантністю або датою, виділяти оглядові статті та патенти (рисунок 1.12).

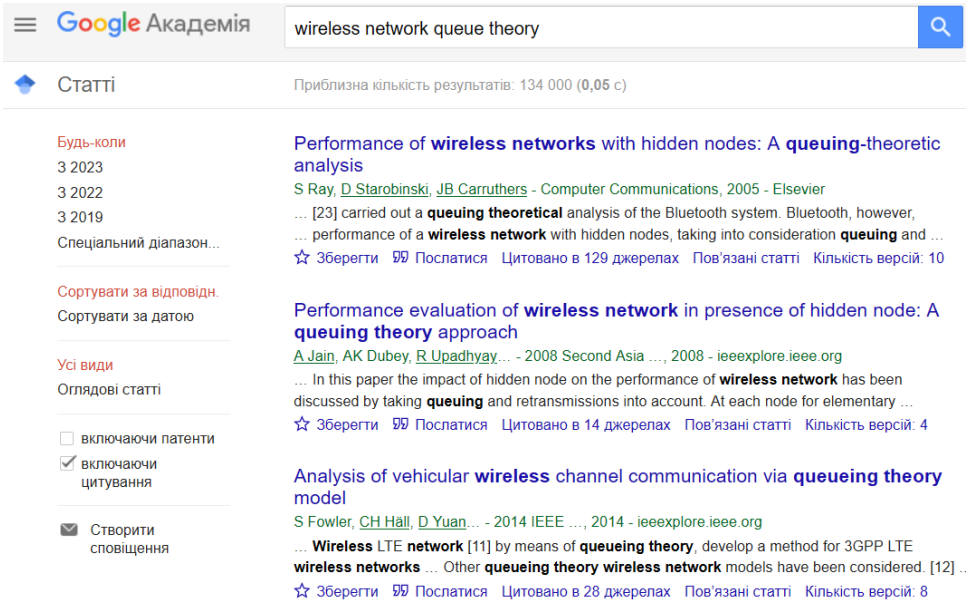


Рисунок 1.12 – Результат пошуку в Гугл Академії.

В Україні існують спеціалізовані наукометричні бази, які фокусуються на українській науковій літературі. Наприклад, «Наукова періодика України» Національної бібліотеки України імені Вернадського, яка індексує наукові журнали України і містить понад 1,3 млн. повних текстів статей (рисунок 1.13).

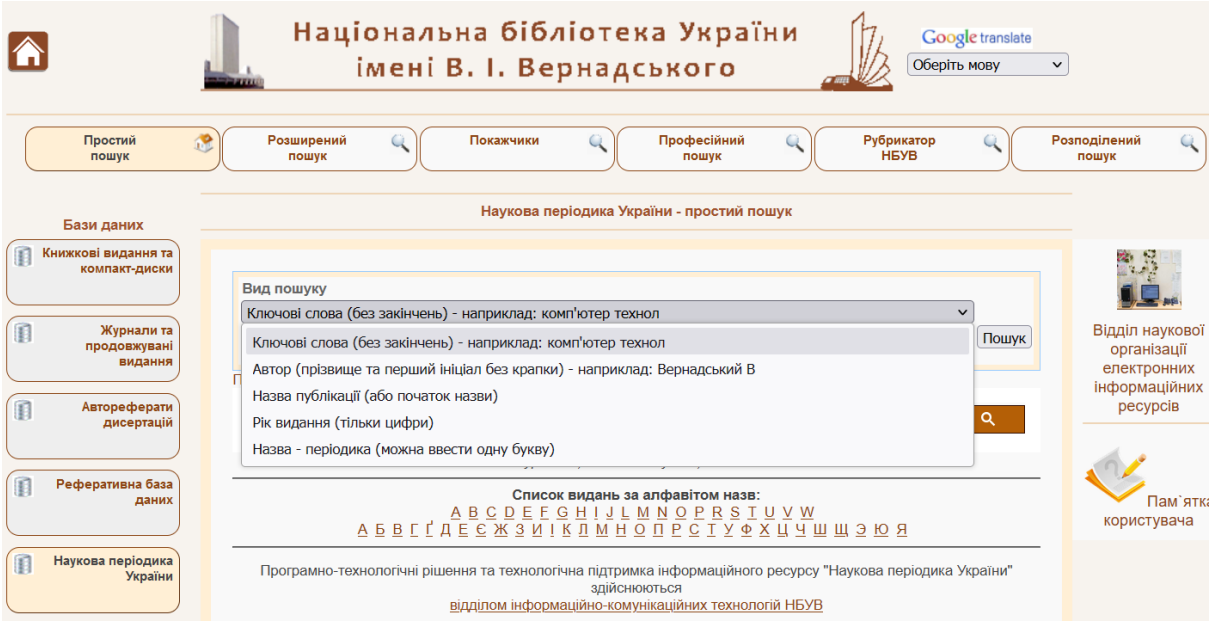
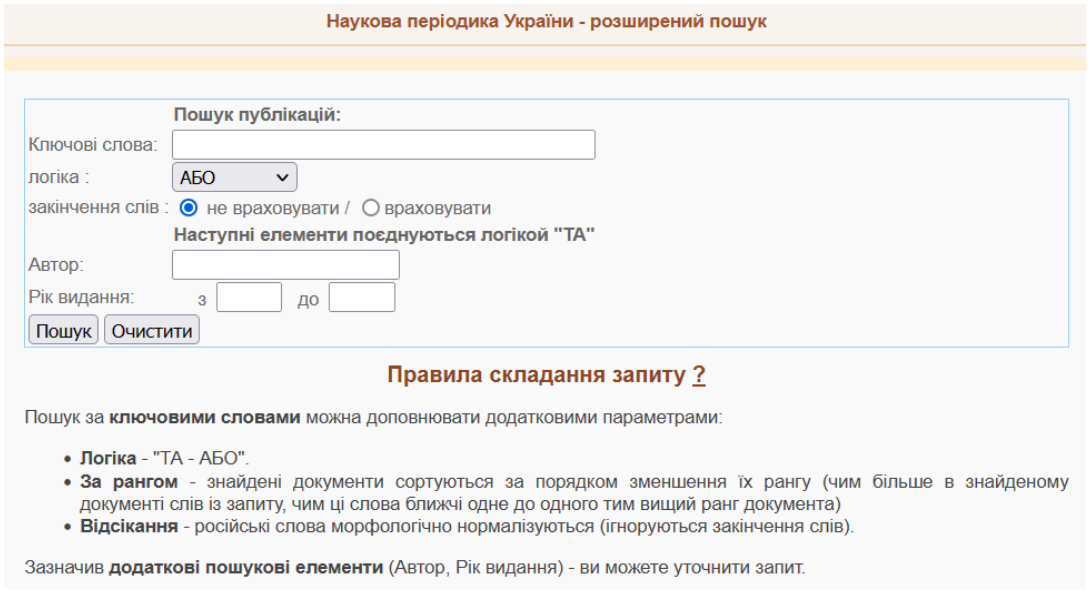


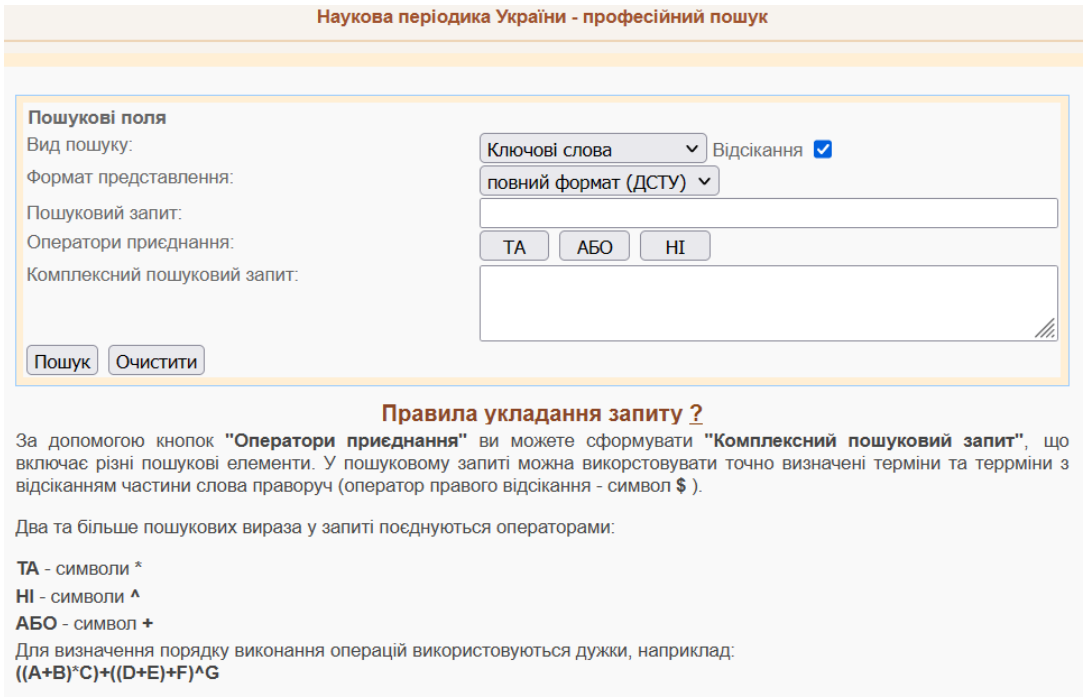
Рисунок 1.13 – База наукової періодики Національної бібліотеки України імені Вернадського

Розширений пошук можливий в трьох варіантах: розширений, професійний, розподілений. Для розширеного пошуку використовуються ключові слова, імена авторів, рік видання. Використовуються логічні оператори АБО, ТА та опускання закінчень слів (рисунок 1.14, а). В режимі професійного пошуку можливо використати оператори пріоритету виконання операцій (рисунок 1.14, б).

Розподілений пошук НБУВ – це пошук по каталогах в обласних бібліотеках України.



а)



б)

Рисунок 1.14 – Типи пошукових функцій

а) розширений пошук, б) професійний пошук

В Україні функціонують різноманітні електронні репозиторії та агрегатори наукової інформації, які надають доступ до широкого спектру наукових робіт, включаючи дисертації, наукові статті тощо. Агрегатор Open Ukrainian Citation Index (OUCI) [43] надає доступ до широкого спектру наукових публікацій, включаючи статті, дисертації, книги та інші наукові матеріали. Пошукова система може використовуватися для пошуку наукових робіт за різними

критеріями, такими як ключові слова, автори, назви робіт, тематика досліджень тощо. Ця база використовує сервіс Cited-by від Crossref [45] та підтримує ініціативу Initiative for Open Citations [46]. Зокрема в OUCI міститься інформація про 500 тис. публікацій у вітчизняних виданнях (рисунок 1.15).

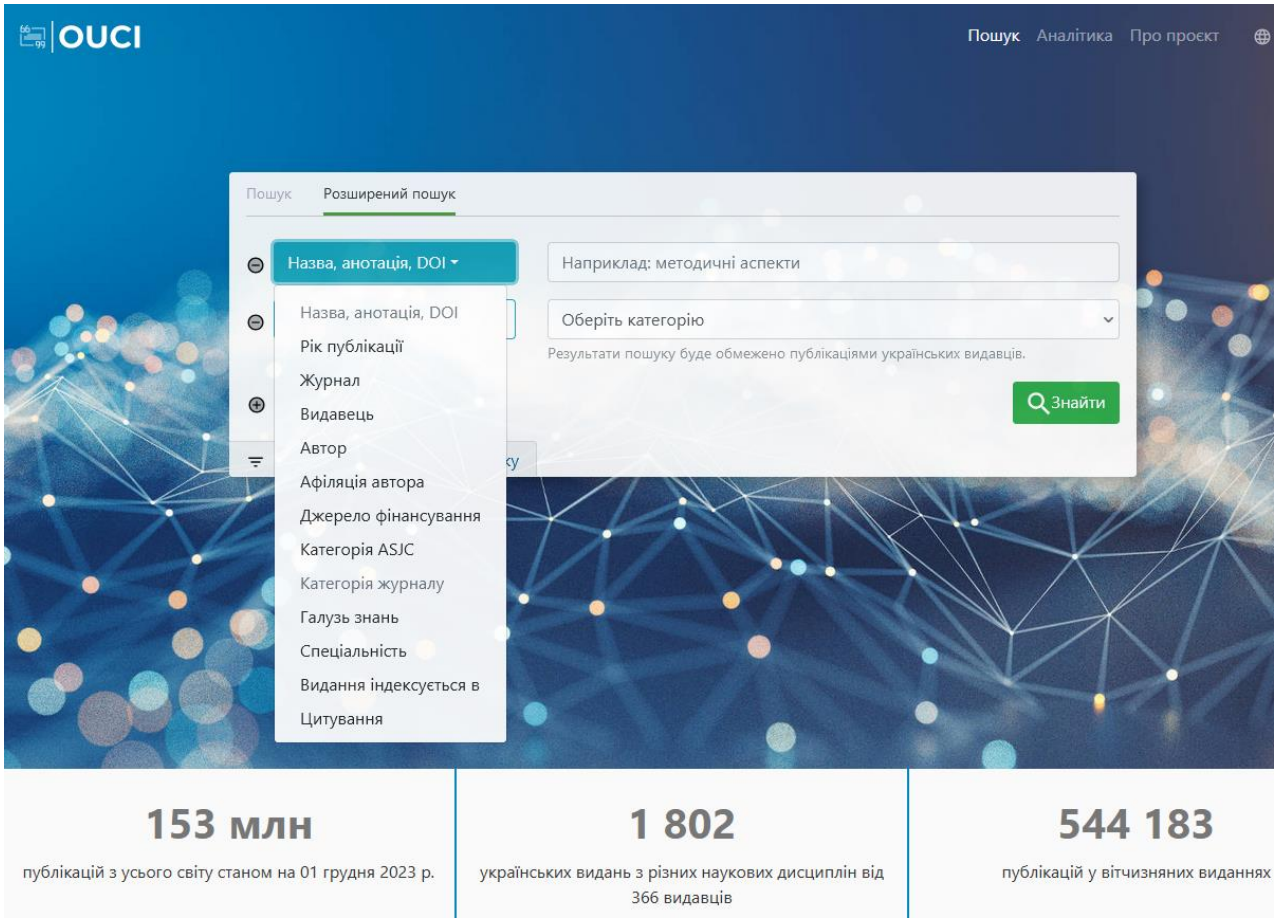


Рисунок 1.15 – Сторінка пошуку Open Ukrainian Citation Index

Функція розширеного пошуку дозволяє вибір з як поширених так і важливих для українського контексту, категорій: категорія журналу, афіляція, джерело фінансування, категорія по класифікації ASJS [44], галузь знань, спеціальність, видання де індексується джерело, цитування джерела.

Для пошуку баз в яких індексується шукана публікація доступні такі варіанти: Scopus, Web of Science, Nature Index journals, Directory of Open Access Journals (DOAJ), Flemish Academic Bibliographic Database for the Social Sciences and Humanities (VABB-SHW).

Українські наукові журнали та видання прагнуть інтегруватися в міжнародні наукові бази даних, такі як Scopus або Web of Science, що підвищує їх визнаність та доступність на міжнародному рівні.

1.3 Пошук інформації в наукометричних базах даних

Пошук інформації в наукометричних базах даних - це складний процес, який можна розділити на кілька основних етапів:

1. Визначення цілей пошуку. Перед початком пошуку необхідно чітко визначити предмет дослідження або конкретну проблему.

2. Вибір Наукометричних Баз. Вибрати наукометричні бази даних, які найкраще відповідають потребам дослідження. Це може бути Scopus, Web of Science, Google Scholar тощо. Врахувати, кожна база має свої унікальні особливості, зокрема щодо охоплення дисциплін, індексування журналів та доступності повних текстів.

3. Підготовка до Пошуку включає формулювання ключових слів та фраз. Необхідно розробити пошукові стратегії, а саме, розглянути використання булевих операторів (AND, OR, NOT), фраз пошуку в лапках, використання зірочок для заміни символів.

4. Здійснення пошуку. Включає і ітеративне введення пошукових запитів.

5. Оцінка первинних результатів. Потрібно оцінити релевантність отриманих документів для досліджуваної теми. За необхідності необхідно уточнити пошукові запити для покращення якості результатів.

Вибір валідної наукометричної бази даних є ключовим кроком в процесі пошуку. Цей процес можна розбити на декілька етапів.

1. Оцінка Потреб Дослідження. Розглянути, які наукові дисципліни охоплюються дослідженням. Різні бази можуть мати свої сильні сторони в певних галузях науки. Визначити, які типи публікацій потрібні - журнальні статті, конференційні матеріали, дисертації, книги тощо.

2. Вибір наукометричних баз. Наприклад, Google Scholar надає широкий доступ до різноманітних ресурсів, тоді як Scopus або Web of Science пропонують більш структуровану інформацію, але часто вимагають підписки. Для деяких специфічних галузей можуть існувати спеціалізовані бази, такі як PubMed для медичних наук або IEEE Xplore для інженерії.

3. Функціональність та інтерфейс. Перевірити можливості пошукової системи бази, включно з використанням булевих операторів, фільтрів, пошуком за ключовими словами.

4. Аналіз якості та охоплення. З'ясувати, які журнали індексуються в кожній базі, оскільки це впливає на охоплення та якість знайденої інформації. Важливо, щоб база регулярно оновлювалася і включала найновіші публікації у відповідній галузі.

Розглянемо інструменти, доступні в наукометричних базах для виконання пошуку, фільтрації результатів, налаштування сповіщень та створення звітів:

1. Інструмент пошуку є основним засобом для знаходження публікацій. Він включає можливості пошуку за ключовими словами, авторами, назвами журналів та іншими параметрами.

2. Фільтри дозволяють уточнити результати пошуку, використовуючи різні критерії, як-от рік публікації, область дослідження, тип документу та інші.

3. Функція налаштування сповіщень дозволяє користувачам отримувати повідомлення про нові публікації або цитування, які відповідають заданим критеріям пошуку.

4. Створення звітів включає генерацію аналітичних зведень та відомостей, заснованих на результатах пошуку та цитуваннях.

Крім структурних елементів самої статті, авторських ключових слів необхідними для пошуку є наукові категорії до яких відносить статтю видавництво або пошукова системи. Для класифікації та категоризації статей використовуються різноманітні системи:

– УДК (UDC) універсальна десяткова класифікація;
– ASJC -All Science Journal Classification;
– ICS -International Classification for Standards;

– JEL (Journal of Economic Literature) Classification Codes: для класифікації літератури у галузі економіки;

– MSC (Mathematics Subject Classification): для класифікації математичних документів;

– PACS (Physics and Astronomy Classification Scheme): для фізики та астрономії;

– CCS (ACM Computing Classification System) розроблена Асоціацією обчислювальної техніки (ACM) для класифікації робіт у галузі комп'ютерних наук;

– MeSH (Medical Subject Headings) для класифікації медичних і біологічних тем;

– LC (Library of Congress Classification): використовується у бібліотеках для класифікації книг та інших матеріалів;

– DDC (Dewey Decimal Classification); Система класифікації, що використовується в бібліотеках для організації матеріалів за предметами.

Система класифікації наукових журналів ASJC "All Science Journal Classification" [44] використовується в рамках бази даних Scopus. У цій системі кожен науковий журнал класифікується згідно з однією або декількома із сотень наявних наукових категорій кожна з яких має свій унікальний код.

Для розробки власних алгоритмів пошуку та збору наукової інформації з наукометричних баз даних, сайтів видавництв і конференцій, потрібно систематично враховувати:

– як організовані та представлені дані у кожному джерелі. Це включає формати документів, схеми найменування, структуру метаданих (автори, назви, дати публікації, ключові слова, реферати, цитування, індекси тощо);

– як функціонують пошукові інтерфейси цих ресурсів - наявні поля для пошуку, фільтри, опції сортування, та інші інструменти пошуку;

– які дані доступні вільно, а які потребують підписки або спеціального доступу. Розглянути можливості отримання доступу через академічні мережі або публічні інтерфейси програмування (API);

– звернути увагу на обмеження, пов'язані з авторським правом та використанням даних, особливо для комерційного застосування;

– оцінити системи класифікації та тегування, які використовуються у цих ресурсах, щоб зрозуміти, як можна систематизувати та категоризувати зібрані дані;

– розглянути можливості експорту даних (наприклад, у формати CSV, JSON, XML) та їхню сумісність із вашими системами;

– як часто дані оновлюються в кожному джерелі, оскільки це впливатиме на актуальність зібраних даних.

Алгоритми пошуку тексту, особливо для наукових публікацій, можуть бути досить різноманітними і складними. Ось деякі з найбільш використовуваних:

1. Булевий пошук використовує булеву логіку (і, або, не) для фільтрації результатів. Він ефективний для вузьких або дуже конкретних запитів [24].

2. Пошук за ключовими словами є основним методом пошуку, який використовує ключові слова та фрази для виявлення релевантних документів [25].

3. Алгоритми індексування індексують текстові документи, щоб швидко знаходити слова або фрази у великому обсязі даних [26].

4. Алгоритми семантичного пошуку використовують семантичні відношення між словами для знаходження більш релевантних результатів, ніж простий пошук за ключовими словами [29].

5. Для наукових публікацій часто використовується аналіз цитувань для визначення значущості та впливу роботи [28].

6. Фільтрація за метаданими, такими як дата публікації, автор, журнал тощо, дозволяє користувачам точніше визначати свої пошукові запити.

7. Сучасні алгоритми пошуку включають застосування машинного навчання та ШІ для більш точного визначення релевантності документів до запиту користувача [29, 33].

8. Алгоритми на основі онтологій використовуються для структурування та організації наукових знань, що дозволяє виявляти зв'язки між різними публікаціями [30].

9. Алгоритми кластеризації групують схожі документи разом, що допомагає користувачам швидше знаходити релевантні документи [31].

1.4 Висновки до розділу 1

У розділі проведено аналіз наукометричних баз даних: великих сховищ наукових публікацій, цитувань тощо. Проаналізовано такі агрегатори повних текстів: Scopus, Web of Science, IEEE Xplore, Google Scholar тощо. Також проаналізовано методику та етапи пошуку інформації в наукометричних базах даних.

2 МОДЕЛІ ПОШУКУ ІНФОРМАЦІЇ

2.1 Основні поняття та визначення

Інформаційний пошук (IR) займається представленням, зберіганням, організацією і доступом до інформаційних ресурсів.

Пошук даних у IR-системах відбувається на основі ключових слів запиту користувача. Пошук інформації відбувається на основі тексту природньою мовою, який не завжди добре структурований і може бути таким семантично неоднозначний. З іншого боку, система пошуку даних (така як реляційна база даних) має справу з даними, які мають чітко визначену структуру та семантику.

Пошук даних, надаючи рішення для користувача системи бази даних, не вирішує проблему пошуку інформації про предмет або тему. Щоб бути ефективною IR-система повинна інтерпретувати зміст інформаційних елементів (документів) у масиві та ранжувати їх за ступенем відповідності для запиту користувача. Ця інтерпретація вмісту документа включає вилучення синтаксису і семантичну інформацію з тексту документа та використання цієї інформації щоб відповідати потребам користувача в інформації. Складність полягає не лише в тому, щоб знати як щоб отримати цю інформацію, а також знати, як використовувати її для визначення релевантності.

Таким чином, поняття релевантності знаходиться в центрі пошуку інформації. Основна мета IR-системи – отримати всі документи, які мають відношення на запит користувача, отримуючи якомога менше нерелевантних документів.

Традиційні системи пошуку інформації зазвичай використовують індексні терміни для індексування та отримання документів. У вузькому сенсі термін індекс – це ключове слово (або група споріднених слів), який має певне власне значення (семантика іменника). У своїй більш загальній формі термін-індекс – це просто будь-яке слово, яке зустрічається в тексті документа. Пошук на основі індексу є простим. Очевидно, що однією з центральних проблем, пов’язаних із системами пошуку інформації, є проблема прогнозування того, які документи є

актуальними, а які ні. Таке рішення зазвичай залежить від алгоритму ранжування, який намагається встановити просте впорядкування отриманих документів. Документи, що з'являються вгорі цього порядку вважаються більш доречними. Таким чином, алгоритми рейтингування є ядром інформаційно-пошукових систем. Алгоритм ранжування працює відповідно до основних передумов щодо поняття релевантності документа.

Існують три класичні моделі інформаційного пошуку: булева, векторна та імовірнісна. У булевій моделі представлені документи та запити як набори індексних термінів. Таким чином, модель є теоретико-множинною. У векторній моделі документи та запити представлені як вектори у t-вимірному просторі. Таким чином, ми говоримо, що модель є алгебраїчною. У імовірнісній моделі запити представляються імовірнісно з використанням теорії імовірності.

Альтернативними булевими моделями є нечіткі моделі. Щодо альтернативних алгебраїчних моделей виділяємо то це латентне семантичне індексування та нейромережеві моделі. Щодо альтернативних імовірнісних моделей то це імовірнісні нейронні мережі.

Класичні моделі інформаційного пошуку вважають, що кожен документ описується набором репрезентативних ключових слів, які називаються термінами індексу. Індексний термін це слово, семантика якого допомагає запам'ятати документ, головні його теми. Таким чином, індексні терміни використовуються для індексування та підсумовування вмісту документа. Загалом терміни-покажчики – це переважно іменники, тому що іменники мають значення самі по собі, тому їх семантику легше ідентифікувати схопити. Прикметники, прислівники та сполучні слова менш корисні як терміни-покажчики оскільки вони працюють переважно як доповнення.

Введемо такі позначення:

k_i – i-те ключове слово;

d_j – j-й документ;

w_{ij} – вага k_i ключового слова у d_j документі.

Вага w_{ij} відображає семантичне значення ключового слова.

Нехай нам задано множину ключових слів:

$$\boldsymbol{K} = \{k_1, k_2, \ldots, k_t\}$$

З документом d_j зв'язаний вектор $\vec{d}_j$, що представляється $\vec{d}_j = \{w_{1j}, w_{2j}, ..., w_{tj}\}$.

Введемо функцію g_i, така що $g_i\left(\vec{d}_j\right) = w_{ij}$.

Пара (k_{i+1}, d_j) є некорельованим з парою (k_i, d_j).

Дамо визначення *IR*-системи. Інформаційна модель пошукової системи є квартет

$$IR = \left\langle \boldsymbol{D}, \boldsymbol{Q}, F, \boldsymbol{R}(q_i, d_j) \right\rangle,$$

де $\boldsymbol{D}$ – множина документів в наборі;

$\boldsymbol{Q}$ – множина запитів;

F – фреймворк для представлення документів і їх взаємозв'язків;

$R(q_i, d_j)$ – функція ранжування, яка повязана з $q_k \in Q$ запитом і представленим документом. Після ранжування система визначає впорядкування документів щодо запиту q_j.

2.2 Булева пошукова модель

Булева модель – це проста пошукова модель, заснована на теорії множин і виразах булевої алгебра. Оскільки концепція множини досить інтуїтивно зрозуміла, булева модель надає структуру, яку легко зрозуміти звичайному користувачеві IR-системи.

Приведемо характеристики основних логічних операцій.

Операція заперечення є унарною, тому що має один аргумент. Інакше її називають інверсією, доповненням, НЕ і позначають $\overline{X}$ або $\neg X$, NOT X.

Запереченням A деякого висловлювання A називається таке висловлювання, яке істинне, коли A помилкове, і помилкове, коли A істинне. Визначення заперечення може бути записано за допомогою таблиці істинності (таблиця 2.1).

Таблиця 2.1 – Таблиця істинності логічної операції заперечення

X	$\overline{X}$
0	1
1	0

Логічне множення. Операцію логічного множення ще називають кон'юнкцією, логічним І. Для позначення даної операції використовують символи ∧, &, точку, яку можна опускати, AND. Кон'юнкція двох висловлювань X і Y називається таке висловлювання, яке істинне тоді і тільки тоді, коли істинні обидва висловлювання X і Y. Визначення кон'юнкції може бути записане у вигляді таблиці істинності (таблиця 2.2).

Таблиця 2.2 – Таблиця істинності логічної операції множення

X	Y	X & Y
0	0	0
0	1	0
1	0	0
1	1	1

Визначення кон'юнкції двох висловлювань природним чином поширюється на будь-яке скінчене число складових: кон'юнкція X_1 & X_2 & X_3 & …& X_n істинна тоді і тільки тоді, коли істинні всі

висловлювання X_1, X_2, X_3, …, X_n, отже, приймає значення «хибність», коли помилкове хоча б одне з цих висловлювань.

Операцію логічного додавання інакше називають диз'юнкцією, логічним АБО. Для позначення логічного додавання використовують символи ∨, +, OR.

Таким чином, диз'юнкцією двох висловлювань називається таке нове висловлювання, яке істинне тоді і тільки тоді, коли істинне хоча б одне з цих висловлювань. Визначення диз'юнкції може бути записане у вигляді таблиці істинності (таблиця 2.3).

Таблиця 2.3 – Таблиця істинності логічної операції додавання

X	Y	X + Y
0	0	0
0	1	1
1	0	1
1	1	1

Визначення диз'юнкції двох висловлювань природним чином поширюється на будь-яке скінчене число складових: диз'юнкція $X_1 + X_2 + X_3 + \ldots + X_n$ істинна тоді і тільки тоді, коли істинне хоча б одне з висловлювань X_1, X_2, X_3, …, X_n, а, отже, приймає значення «хибність», коли всі висловлювання помилкові.

Застосуємо дані логічні операції для логічної пошукової моделі.

Для заданої множини ключових слів $\boldsymbol{K} = \{k_1, k_2, \ldots, k_t\}$ ваги ключових слів є двійковими $w_{ij} = \{0, 1\}$.

Крім того введемо множину операцій над логічними змінними $\boldsymbol{O}$:

$$\boldsymbol{O} = \{\wedge, \vee, \neg\},$$

де ∧ – операція кон'юнкції (логічне множення);

∨ – операція диз'юнкції (логічне додавання);

$\neg$ – заперечення.

Дані операції виконуються над ключовими словами.

Логічний вираз запиту може бути представлений у диз'юнктивній нормальній формі.

Розглянемо приклад.

Нехай запит буде представлений у формі

$$q = k_a \wedge (k_b \vee \neg K_c).$$

Здійснимо перетворення

$$q = k_a \wedge (k_b \vee \neg K_c) = (k_a \wedge k_b) \vee (k_a \wedge \neg K_c) =$$
$$= (\overline{\overline{k_a \wedge k_b}}) \vee (\overline{\overline{k_a \wedge \neg K_c}}) = (\overline{\overline{k_a} \vee \overline{k_b}}) \vee (\overline{\overline{k_a} \vee K_c}).$$

Даний запит в диз'юнктивній нормальній формі може бути представлений

$$\vec{q}_{dnf} = (1, 1, 1) \vee (1, 1, 0) \vee (1, 0, 0).$$

Отже, для булевої моделі ключові слова є бінарними і мають вагу $w_{ij} = \{0, 1\}$.

Тому запит q є булевим виразом. Допустимо що $\vec{q}_{dnf}$ є диз'юнктивна нормальна форма для запиту q. Допустимо що $\vec{q}_{cc}$ є кон'юнктивна нормальна форма для $\vec{q}_{dnf}$.

Тоді подібність документа d_j до запиту q визначається так:

$$sim(\vec{d}_j, q) = \begin{cases} 1, & \text{якщо } \exists \vec{q}_{cc} \mid (\vec{q}_{cc} \in \vec{q}_{dnf}) \wedge (\forall k_i; q_i(\vec{d}_j) = q_i(\vec{q}_{cc}), \\ 0, & \text{в інших випадках.} \end{cases}$$

Якщо $sim(\vec{d}_j, \vec{q}) = 1$, тоді це означає, що документ d_j є релевантним до запиту q. В іншому випадку він є неревантним.

Крім того, запити вказуються як логічні вирази, які мають точну семантику. Отже, завдяки формалізму булева модель стала популярною і основною моделлю в комерційних пошукових системах.

Недоліки моделі такі: 1) стратегія пошуку базується на двійковому критерії прийняття рішення (тобто документ є релевантний або нерелевантний) без оцінювання; 2) хоча булеві вирази мають точну семантику, часто це не просто зробити перевести інформаційну потребу в логічний вираз.

Незважаючи на ці недоліки, булева модель все ще є домінуючою моделлю в комерційних системах.

2.3 Векторна пошукова модель

Векторна модель [47-49] також може використовувати двійкові ваги, але пропонує структуру, у якій можливе часткове збігання. Це забезпечується шляхом призначення небінарних ваг ключових слів у запитах і в документах. Ці вагові коефіцієнти остаточно використовуються для обчислення ступеня подібності між кожним документом, що зберігається в системі, і запитом користувача. Сортуючи отримані документи у порядку зменшення ступеня подібності, векторна модель бере до уваги документи, які відповідають запиту термінів лише частково. Основний кінцевий ефект полягає в тому, що ранжований документ краще відповідає запиту користувача, ніж набір відповідей, отриманий за допомогою булевої моделі.

Вагові коефіцієнти ключових слів можна обчислити різними способами. У роботі авторів Солтон і Макгілл [49] розглядають різні методи зважування ключових слів. Інший підхід до зважування ключових слів полягає в кластеризації ключових слів. Маючи набір *C* об’єктів і нечіткий опис набору *A*, мета простий алгоритм кластеризації може полягати в розділені набору *C*

об’єктів на два набори: перший, який складається з об’єктів, пов’язаних із набором *A*, і другий той, який складається з об’єктів, не пов’язаних із набором *A*. Тут розпливчастий опис означає, що ми не маємо повної інформації для точного визначення об’єкти є і не входять до множини *A*.

Наприклад, пацієнти лікаря-спеціаліста при раку можна розділити на п’ять класів: термінальний, прогресуючий, метастазний, з діагнозом, здоровий. Знову ж таки, можливі описи класів можуть бути неточними (і не унікальний), і проблема полягає в тому, щоб вирішити, до якого з цих класів слід призначити нового пацієнта. Однак у подальшому ми лише обговорюємо простіша версія проблеми кластеризації (тобто така, яка розглядає лише два класи), тому що все, що потрібно, це рішення про те, які документи прогнозовано як релевантні, а які – як нерелевантні (з щодо даного запиту користувача).

Щоб розглядати проблему IR як проблему кластеризації ми розглядаємо набір *C* об’єктів і запит користувача. У цьому сценарії проблему IR можна звести до проблеми визначення того, які документи є у множині *A*, а які ні (тобто проблему IR можна розглядати як а проблему кластеризації). У проблемі кластеризації необхідно розв’язати дві основні проблеми.

По-перше, потрібно визначити, які характеристики краще описують ключові слова у множині *A*. По-друге, потрібно визначити, які ознаки краще відрізняють об’єкти в наборі *A* від інших об’єктів у наборі *B*. Перший набір ознак забезпечує кількісну оцінку центра кластерної подібності, тоді як другий набір ознак забезпечує кількісну оцінку міжкластерної несхожості. Найуспішніші алгоритми кластеризації намагаються збалансувати ці два ефекти.

У векторній моделі внутрішньокластерна подібність кількісно визначається шляхом вимірювання ненормована частота терміну k_i всередині документа d_j. Таким терміном є частота t_f і є одним з показників того, наскільки добре це термін описує вміст документа (тобто внутрішньодокументну характеристику). Крім того, міжкластерна відмінність кількісно визначається шляхом вимірювання зворотного частотності терміна k_i серед документів набору. Цей фактор зазвичай називають зворотною частотою документа або фактором IDF. Мотивацією для використання фактора IDF є ті терміни, які

зустрічаються в багатьох наборах не дуже корисні для того, щоб відрізнити релевантний документ від а нерелевантного. Як і у випадку з хорошими алгоритмами кластеризації, найефективнішим розв'язком є алгоритми зважування для IR, які намагаються збалансувати ці два ефекти.

Для векторної моделі ваги w_{ij} зв'язані з парою (k_i, d_j) є додатними і небінарними. Також у запиті ключові слова є зваженими. Допустимо w_{iq} – вага, яка зв'язана з парою (k_i, q), де $w_{iq} \geq 0$. Тоді вектор запиту q визначається як:

$$\vec{q} = \{w_{1q}, w_{2q}, ..., w_{tq}\},$$

де t – загальна кількість слів в системі. Документ d_j представляється у вигляді вектора: $\vec{d}_j = \{w_{1j}, w_{2j}, ..., w_{tj}\}$. Документ d_j і запит користувача q представляються за допомогою t-вимірних векторів. Тому векторна модель пропонує оцінювати ступінь подібності документа d_j із запитом q як кореляцію між векторами $\vec{d}_j$ і $\vec{q}$. Ця міра може бути визначена як cos кута між цими двома векторами.

$$sim(\vec{d}_j, \vec{q}) = \frac{\vec{d}_j \times \vec{q}}{|\vec{d}_j| \times |\vec{q}|} = \frac{\sum_{i=1}^{t} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \times \sqrt{\sum_{i=1}^{t} w_{iq}^2}}.$$

На рисунку 2.1 приведена графічна інтерпретація подібності запиту і документу.

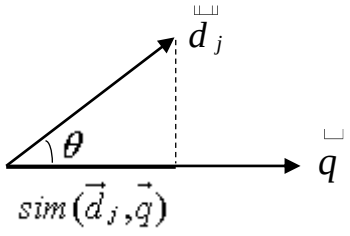


Рисунок 2.1 – Графічна інтерпретація подібності запиту і документу

Ваги $w_{ij} \geq 0$, $w_{iq} \geq 0$ і $sim(\vec{d}_j, \vec{q})$ змінюються від 0 до 1.

Позначимо через N загальну кількість документів в системі і n_i – кількість документів у яких з'являється ключове слово k_i. Допустимо $freg_{ij}$ – частота згадування ключового слова k_i в тексті документу d_j.

Тоді нормалізована частота f_{ij} ключового слова k_i в документі d_j обчислюється за формулою:

$$f_{ij} = \frac{freg_{ij}}{\max\limits_{l} freg_{lj}},$$

де максимум обчислюється для всіх ключових слів, які зустрічаються в документі d_j. Дальше, допустимо idf_i – інверсна частота для ключового слова k_i, яка обчислюється за формулою:

$$idf_i = \log \frac{N}{n_i}.$$

Відома схема зважування ключових слів обчислюється за формулою:

$$w_{ij} = f_{ij} \log \frac{N}{n_i}.$$

Інший вираз для зважування ключового слова:

$$w_{iq} = \left(0{,}5 + \frac{0{,}5\, freg_{iq}}{\max\limits_{l} freg_{lq}}\right) \cdot \log \frac{N}{n_i},$$

де $freg_{iq}$ – необроблена частота ключового слова k_i у тексті інформаційного запиту q.

Основними перевагами векторної моделі є такі: 1) схема зважування ключових слів покращує продуктивність пошуку; 2) стратегія часткового збігу дозволяє пошук документів, що наближають умови запиту; 3) модель сортує документи за ступенем їх схожості із запитом. Теоретично векторна модель має такий недолік, що ключові слова вважаються незалежними. Однак на практиці врахування залежностей ключових слів може бути недоліком. Через локальність багатьох залежностей ключових слів, їх невибіркове застосування до всіх документів у наборі насправді може погіршити загальну продуктивність.

Незважаючи на свою простоту, векторна модель є стійкою стратегією ранжування загальної вибірки документів. Крім того, вона є простою і швидкою. Тому векторна модель є популярною пошуковою моделлю.

2.4 Імовірнісна пошукова модель

Приведемо основні теоретичні відомості.

Означення. Класичною *імовірністю* події A називається відношення числа елементарних рівноможливих подій, що сприяють появі події A, до загального числа елементарних рівноможливих подій, що утворюють повну групу

$$P(A) = \frac{m}{n}.$$

Властивості:

1) $P(A) = 1$, якщо A подія достовірна;

2) $P(A) = 0$, якщо A подія неможлива;

3) $0 < P(A) < 1$, якщо A випадкова подія.

Елементи комбінаторики. Означення. Комбінаціями називаються групи, які відрізняються одна від одної хоч би одним елементом. Число комбінацій обчислюється за формулою:

$$C_n^m = \frac{n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n-m+1)}{m!};$$

або

$$C_n^m = \frac{n!}{(n-m)! \cdot m!}.$$

Властивості числа комбінацій:

1) $C_n^0 = C_n^n = 1$;

2) $C_n^1 = C_n^{n-1} = n$;

3) $C_n^m = C_n^{n-m}$;

4) $C_n^0 + C_n^1 + C_n^2 + \ldots + C_n^n = 2^n$.

Означення. Розміщеннями називаються групи, які відрізняються одна від іншої хоч би одним елементом або порядком розташування цих елементів в групі. Число розміщень обчислюється за формулою:

$$A_n^m = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n-m+1).$$

Означення. Перестановками називаються групи, які відрізняються тільки порядком розташування елементів, а не самими елементами. Число перестановок обчислюється за формулою:

$$P_n = A_n^n = n!$$

Означення. Відносною частотою випадкової події називається відношення числа випробувань, в яких подія відбулася, до загального числа фактично проведених випробувань.

$$W(A) = \frac{M}{N}, де$$

$$0 \le W(A) \le 1.$$

Відносна частота події береться за статистичне означення ймовірності.

Означення. Умовною ймовірністю $P_A(B)$ називається ймовірність події B при умові, що подія A відбулася.

Означення. Дві події називаються незалежними, якщо ймовірність однієї з них не змінюється від того, відбулася чи ні інша.

$$P_A(B) = P_{\overline{A}}(B),$$

де $\overline{A}$ – подія, яка полягає в тому, що при випробуванні подія A не відбувається, відбувається подія, протилежна до події A.

Означення. Дві події називаються залежними, якщо ймовірність однієї з них змінюється внаслідок відбуття іншої, тобто

$$P_A(B) \neq P_{\overline{A}}(B).$$

Теорема додавання ймовірностей сумісних подій.

Ймовірність появи хоч би однієї з двох сумісних подій дорівнює сумі ймовірностей тих подій без ймовірності їх сумісної появи:

$$P(A+B) = P(A) + P(B) - P(AB).$$

Теорема має місце і для довільного скінченого числа сумісних подій.

Теорема множення ймовірностей.

Ймовірність сумісної появи двох подій дорівнює добутку ймовірностей однієї з них на умовну ймовірність другої, обчислену в припущенні, що перша подія вже відбулася:

$$P(AB) = P(A) \cdot P_A(B) = P(B) \cdot P_B(A),$$

а для незалежних подій

$$P(AB) = P(A) \cdot P(B),$$

тобто ймовірність сумісної появи двох незалежних подій дорівнює добутку ймовірностей цих подій.

Наслідок. Ймовірність сумісної появи декількох подій дорівнює добутку ймовірностей однієї з них на умовну ймовірність всіх інших, причому ймовірність кожної наступної події обчислюється в припущенні, що всі попередні події вже відбулися

$$P(A_1A_2A_3...A_n) = P(A_1) \cdot P_{A_1}(A_2) \cdot P_{A_1A_2}(A_3) \cdot ... \cdot P_{A_1A_2...A_{n-1}}(A_n),$$

а також ймовірність сумісної появи декількох подій, незалежних в сукупності, дорівнює добутку ймовірностей тих подій:

$$P(A_1A_2A_3...A_n) = P(A_1) \cdot P(A_2) \cdot P(A_3) \cdot ... \cdot P(A_n).$$

Теорема. Сума ймовірностей подій, які утворюють повну групу подій, дорівнює одиниці:

$$P(A_1) + P(A_2) + P(A_3) + ... + P(A_n) = 1,$$

а сума ймовірностей події A і протилежної події $\overline{A}$ дорівнює 1, тобто $P(A) + P(\overline{A}) = 1$ або $p + q = 1$.

Ймовірність появи хоч би однієї події та тільки однієї події.

Якщо в результаті випробування можуть відбутися події A_1, A_2, ..., A_n, ймовірності появи кожної з них відомі і які є незалежними в сукупності, то

ймовірність події A – появи хоч би однієї з цих подій у випробуванні, знаходимо за формулою

$$P(A) = 1 - q_1 \cdot q_2 \cdot \ldots \cdot q_n,$$

де $q_1 = P(\overline{A_1})$, $q_2 = P(\overline{A_2})$, ..., $q_n = P(\overline{A_n})$.

Якщо, $P(A_1) = P(A_2) = \ldots = P(A_n) = p$, то $P(A) = 1 - q^n$, де $q = 1 - p$. А якщо події A_1, A_2, ..., A_n є залежними, то $P(A) = 1 - P(\overline{A_1}) \cdot P_{\overline{A_1}}(\overline{A_2}) \cdot \ldots \cdot P_{\overline{A_1}\,\overline{A_2}\ldots\overline{A_{n-1}}}(\overline{A_n})$.

Формула повної ймовірності та формула Байєса.

Ймовірність події A, яка може відбутися лише при появі однієї з несумісних подій B_1, B_2 ,..., B_n, що утворюють повну групу, дорівнює сумі добутків ймовірностей кожної з них на відповідну умовну ймовірність події A. Отже, формула повної ймовірності

$$P(A) = P(B_1)P_{B_1}(A) + P(B_2)P_{B_2}(A) + \ldots + P(B_n)P_{B_n}(A),$$

де $P(B_1) + P(B_2) + \ldots + P(B_n) = 1$.

Нехай подія A може відбутися лише при умові появи однієї з несумісних гіпотез B_1, B_2, ..., B_n, які утворюють повну групу подій. Якщо подія A вже відбулася, то ймовірності гіпотез можуть бути переоцінені за формулою Байєса

$$P_A(B_i) = \frac{P(B_i) \cdot P_{B_i}(A)}{P(A)} \ (i = \overline{1,n}),$$

де $P(A)$ – імовірність настання події A, обчислена за формулою повної ймовірності.

Застосуємо даний математичний апарат для імовірнісної моделі пошуку інформації.

Нехай на запит користувача формується набір потрібних документів. Назвемо це набір ідеальним набором документів. Ми повинні сформувати такий запит, щоб отримати ідеальний набір документів. Таким чином, процес запиту – це процес уточнення властивостей ідеального набору відповідей (що аналогічно інтерпретації IR проблемі як проблемі кластеризації). Але ми не знаємо цих властивостей. Ми знаємо, що існують ключові слова, семантику яких можна використовувати для характеристики цих властивостей. Це початкове припущення дозволяє нам створити попередній імовірнісний опис ідеального набору відповідей, який використовується для отримання першого набору документів. Потім ініціюється взаємодія з користувачем за допомогою вдосконалення ймовірнісного опису ідеального набору відповідей. Такі взаємодія може відбуватися наступним чином. Користувач переглядає отримані документи та вирішує, які саме актуальні, а які ні (лише перші верхні документи підлягає обстеженню). Потім система використовує цю інформацію для уточнення опису ідеального набору відповідей. Повторюючи цей процес багато разів, це очікується що такий опис буде розвиватися і ставати ближчим до реального опису.

Імовірнісна модель базується на наступному фундаментальному припущенні.

Припущення (ймовірнісний принцип). Дано запит користувача q і документ d_j у наборі. Імовірнісна модель намагається оцінити ймовірність того, що користувач вважатиме документ d_j цікавим (тобто актуальним). Модель припускає, що ймовірність релевантності залежить від запиту та отриманого документа. Крім того, модель припускає, що існує підмножина всіх документів, які користувач віддає перевагу як набір відповідей на запит q. Такий ідеальний набір відповідей позначений R і повинен максимізувати загальну ймовірність. Передбачається, що документи в наборі R мають відношення до запиту. Передбачається, що документи, які не входять до цього набору, будуть нерелевантними. Це припущення не вказує як обчислити ймовірність релевантності.

За запитом q імовірнісна модель призначає кожному документу d_j міру подібності як відношення:

$$\frac{P(d_j|\text{релевантний до q})}{P(d_j|\text{нерелевантний до q})},$$

що розраховує шанс того, що документ d_j релевантний до запиту q. Для імовірнісної моделі ваги змінних ключових слів є бінарними, тобто

$$w_{ij} \in \{0, 1\}, w_{iq} \in \{0, 1\}.$$

Запит q є підмножиною ключових слів. Допустимо R є множиною документів, які є релевантними. Тоді $\bar{R}$ є доповненням до R (тобто є набором нерелевантних документів).

Допустимо $P(R|\vec{d}_j)$ – ймовірність того, що документ d_j релевантний до запиту q і $P(\bar{R}|\vec{d}_j)$ – ймовірність того, що документ d_j нерелевантний до запиту q. Тоді подібність $sim(\vec{d}_j, \vec{q})$ документа d_j до запиту q визначається як відношення

$$sim(\vec{d}_j, \vec{q}) = \frac{P(R|\vec{d}_j)}{P(\bar{R}|\vec{d}_j)}.$$

Використовуючи теорему Байєса, отримаємо:

$$sim(\vec{d}_j, \vec{q}) = \frac{P(\vec{d}_j|R)P(R)}{P(\vec{d}_j|\bar{R})P(\bar{R})},$$

де $P(\vec{d}_j|R)$ – ймовірність випадково вибраного документа d_j із набору R релевантних документів;

$P(R)$ – ймовірність того, що документ випадково вибраний з набору R є релевантним;

$P(\vec{d}_j \mid \overline{R})$ – ймовірність випадково вибраного документа d_j із набору $\overline{R}$ релевантних документів;

$P(\overline{R})$ – ймовірність того, що документ випадково вибраний з набору $\overline{R}$ є релевантним.

Оскільки $P(R)$ і $P(\overline{R})$ є однаковими для всіх документів в наборі, тому ми можемо подібність записати наступним чином:

$$sim(\vec{d}_j, \vec{q}) \approx \frac{P(\vec{d}_j \mid R)}{P(\vec{d}_j \mid \overline{R})}.$$

Враховуючи незалежність ключових слів, отримаємо

$$sim(\vec{d}_j, \vec{q}) \approx \frac{\left(\prod_{i=1}^{t} P(k_i \mid R)\right) \cdot \left(\prod_{i=1}^{t} P(\overline{k}_i \mid R)\right)}{\left(\prod_{i=1}^{t} P(k_i \mid \overline{R})\right) \cdot \left(\prod_{i=1}^{t} P(\overline{k}_i \mid \overline{R})\right)}, \qquad (2.1)$$

де $P(k_i \mid R)$ – ймовірність того, що ключове слово k_i представлено у документі випадково вибраному з набору R;

$P(\overline{k}_i \mid R)$ – ймовірність того, що ключове слово k_i відсутнє у документі випадково вибраному з набору R;

$P(k_i \mid \overline{R})$ – ймовірність того, що ключове слово k_i представлено у документі випадково вибраному з набору $\overline{R}$;

$P(\overline{k}_i \mid \overline{R})$ ймовірність того, що ключове слово k_i відсутнє у документі випадково вибраному з набору $\overline{R}$.

Використавши повноту подій і провівши логарифмування формули (2.1), отримаємо наступну формулу:

$$sim(\vec{d}_j, \vec{q}) \approx \sum_{i=1}^{t} w_{iq} w_{ij} \left(\log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)}\right) + \left(\log \frac{1 - P(k_i \mid \overline{R})}{P(k_i \mid \overline{R})}\right). \qquad (2.2)$$

Формула (2.2) є основним рівнянням для обчислення подібності документа до запиту.

Для початкової множини R, необхідно на початку обчислювати $P(k_i \mid R)$ і $P(k_i \mid \overline{R})$.

Зробимо такі припущення:

1) імовірність $P(k_i \mid R)$ є константою для всіх ключових слів k_i і дорівнює 0,5;

2) розподіл ключових слів у нерелевантних документах може бути такий самий, як розподіл ключових слів у всіх документах набору.

Ці два припущення дають:

$$P(k_i \mid R) = 0,5,$$

$$P(k_i \mid \overline{R}) = \frac{n_i}{N},$$

де n_i – кількість документів, які включають ключове слово k_i;

N – загальна кількість документів у наборі.

Покращення початкових імовірностей можна здійснити так.

Нехай V – підмножина документів початково проранжованих імовірнісною моделлю. Таку підмножину можна визначити, як підмножину r – ранжованих документів, де r – порогове значення.

Дальше позначимо підмножину V_i – кількість документів, які мають ключове слово k_i. Для покращення оцінки ймовірності ми повинні обчислити такі імовірності: $P(k_i \mid R)$ та $P(k_i \mid \overline{R})$.

Ми можемо зробити такі припущення:

$$P(k_i \mid R) = \frac{V_i}{V},$$

$$P(k_i \mid \overline{R}) = \frac{n_i - V_i}{N - V}.$$

Цей процес може бути рекурентно повторений.

Ці формули є некоректними, коли $V = 1$, $V_i = 0$. Тому до цих формул додаються коригувальні коефіцієнти. Отже, формули можна представити так:

$$P(k_i \mid R) = \frac{V_i + 0,5}{V + 1},$$

$$P(k_i \mid \overline{R}) = \frac{n_i - V_i + 0,5}{N - V + 1}.$$

Постійні коефіцієнти у формулах не завжди є задовільними.

Тому формули по-іншому представляються так:

$$P(k_i \mid R) = \frac{V_i + \frac{n_i}{N}}{V + 1},$$

$$P(k_i \mid \overline{R}) = \frac{n_i - V_i + \frac{n_i}{N}}{N - V + 1}.$$

Основна перевага імовірнісної моделі, в теорії, полягає в тому, що документи ранжуються в порядку зменшення ймовірності їхньої релевантності. недоліки включають: (1) необхідність вгадувати початкове розділення документів на релевантні та нерелевантні множини; (2) той факт, що метод не приймає враховувати частоту, з якою індексний термін зустрічається в документі (тобто всі ваги є двійковими); і (3) прийняття припущення про незалежність для умов індексу. Однак, як обговорювалося для векторної моделі, це не ясно що незалежність термінів індексів є поганим припущенням у практичних ситуаціях.

2.5 Висновки до розділу 2

Взагалі булева модель вважається найслабшим класичним методом. Його основною проблемою є нездатність розпізнавати часткові збіги, що часто призводить до поганої продуктивності. Існує певна суперечка щодо того, чи є ймовірнісним модель перевершує векторну модель. Крофт провів кілька дослідів і припустив, що ймовірнісна модель забезпечує кращу ефективність пошуку. Однак експерименти, проведені пізніше Солтоном і Баклі, спростовують це твердження.

За допомогою кількох різних експериментів Солтон і Баклі показали, що векторна модель перевершила імовірнісну модель із загальними наборами. Це також, здається, домінуюча думка серед дослідників, практиків, і веб-спільнота, де векторна модель дуже популярна.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПОШУКОВОЇ СИСТЕМИ

3.1 Системні вимоги до програми

Для розробки програмної системи використовувався персональний ноутбук з такими системними характеристиками:

– чотирьох ядерний процесор 11th Gen Intel Core i7-1185G7 з тактовою частотою 3.00 ГГц;

– оперативна пам’ять DDR4 обсягом 32Гб;

– інтегрована відеокарта GPU Intel Iris Xe Graphics;

– SSD диск розміром 512 ГБ.

Також на комп’ютері встановлене наступне програмне забезпечення:

– операційна система Windows 10 x64;

– локальний веб-сервер типу XAMPP/WAMP/MAMP та програмне середовище, створене спеціально для потреб веб-розробників;

– IDE (інтегроване середовище розробки) інструмент PhpStorm 2023 для розробки PHP та веб-проектів від JetBrains;

– інструменти для розробки фронтенда:

– Node.js та npm: для встановлення та керування фронтенд-залежностями та інструментами;

– Webpack/Babel: якщо потрібна більш складна обробка фронтенда, наприклад, для використання ES6;

– утиліти для PHP:

– Composer: Менеджер залежностей для PHP, що дозволяє легко включати сторонні бібліотеки до проекту;

– Git (система контролю версій) та GitHub для зберігання коду проекту.

3.2 Мова програмування PHP та її характеристики

Популярна універсальна мова сценаріїв, яка особливо підходить для веб-розробки (рисунок 3.1). Швидка, гнучка та практична, PHP використовується у всьому – від блогу до найпопулярніших веб-сайтів у світі.

PHP Version 8.0.8

System	Windows NT EPUAODEW00A2 10.0 build 19045 (Windows 10) AMD64
Build Date	Jun 29 2021 15:59:22
Build System	Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cscript /nologo /e:jscript configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=..\..\..\instantclient\sdk,shared" "--with-oci8-19=..\..\..\instantclient\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	*no value*
Loaded Configuration File	C:\Server\modules\php\PHP_8.0\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20200930
PHP Extension	20200930
Zend Extension	420200930
Zend Extension Build	API420200930,TS,VS16
PHP Extension Build	API20200930,TS,VS16
Debug Build	no
Thread Safety	enabled
Thread API	Windows Threads
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, zlib.*, bzip2.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.0.8, Copyright (c) Zend Technologies

Рисунок 3.1 – Загальна інформація PHP v8.0

Ключові характеристики PHP:

1. Вбудована підтримка HTML: PHP код можна легко вставити в HTML документи. Це дозволяє розробникам створювати динамічний вміст, який виконується на сервері, перш ніж сторінка надсилається клієнту.

2. Серверно-орієнтована мова: PHP в основному використовується для серверної розробки. Це означає, що PHP скрипти виконуються на сервері, а результати виконання передаються до веб-браузера користувача.

3. Легкість у вивченні та використанні: PHP має простий синтаксис та легко вчити, що робить її популярною серед новачків у веб-розробці.

4. Гнучкість та розширюваність: PHP сумісна з багатьма базами даних, операційними системами та веб-серверами, що робить її вкрай гнучкою. Крім того, існує велика кількість додаткових розширень та бібліотек.

5. Підтримка об'єктно-орієнтованого програмування: PHP підтримує ООП (об'єктно-орієнтоване програмування), що дозволяє розробникам створювати модульні програми з використанням класів та об'єктів.

6. Безпека: PHP містить різні вбудовані функції для забезпечення безпеки веб-додатків, хоча правильне їх використання залежить від розробника.

7. Велика спільнота та підтримка: PHP має велику та активну спільноту розробників, завдяки чому доступна обширна документація, форуми для підтримки та безліч ресурсів для навчання.

8. Вбудовані функції для роботи з веб: PHP містить широкий спектр вбудованих функцій для роботи з HTTP запитами, обробки форм, сесій, кукісів та інших складових веб-розробки.

9. Швидкість виконання: Завдяки постійним оптимізаціям і оновленням, PHP забезпечує хорошу швидкість виконання, особливо у останніх версіях.

10. Крос-платформність: PHP може виконуватися на різних операційних системах, таких як Windows, Linux та MacOS, що робить її дуже доступною для різноманітних розробників.

11. Постійне оновлення: Остання версія релізу на даний момент це PHP v8.3.

3.3 Структура програми

Опишемо структуру програми (рисунок 3.2).

Фронтенд (Front-End):

– HTML (HyperText Markup Language): використовується для структурування контенту на веб-сторінці;

– CSS (Cascading Style Sheets): використовується для стилізації та візуального оформлення веб-сторінок;

– JavaScript: Мова програмування, що використовується для додавання інтерактивності та динамічних елементів на веб-сторінку.

Бекенд (Back-End):

– серверна мова програмування: PHP v8.0;

– сервер: Apache v2.4, який дозволяє веб-програмі обробляти запити.

База Даних (Database):

– система управління базами даних (DBMS): MySQL v8.0 – використовується для зберігання та управління даними.

API (Application Programming Interface):

– REST API: Забезпечує інтерфейс для взаємодії між бекендом стороннього провайдера як Elsiever, IEEE. Результат отримуємо у форматі JSON.

Безпека (Security):

– аутентифікація та авторизація: Механізми для контролю доступу користувачів.

Для розробки використовується Laravel v9.x – це PHP фреймворк веб-додатків з виразним, елегантним синтаксисом. Побудований на популярній архітектурі Model-View-Controller (MVC), архітектура якої приведена на рисунку 3.3.

Це означає, що Laravel використовує паттерн проектування, який розділяє додаток на три основні компоненти: модель (modal), вид (view) та контролер (controller). Цей підхід сприяє організації коду, робить його більш модульним та спрощує розробку та тестування.

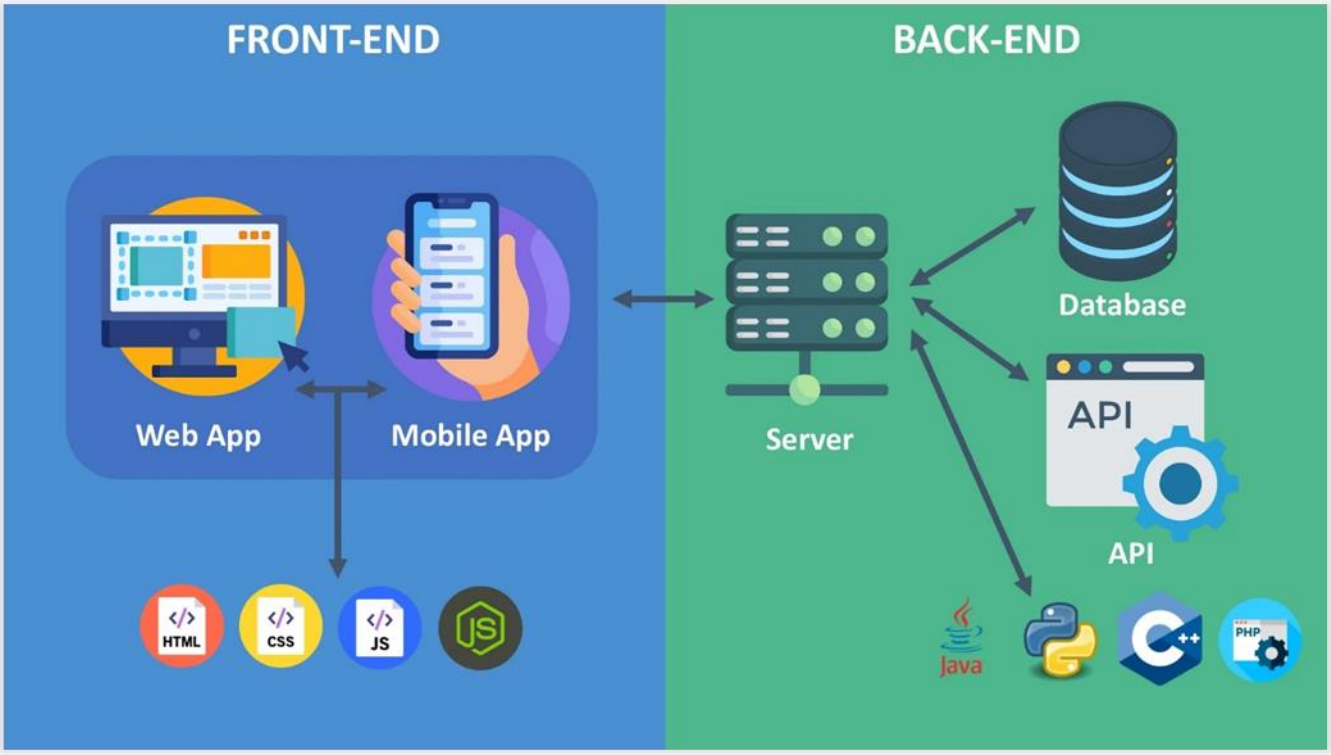


Рисунок 3.2 – Взаємодія програми

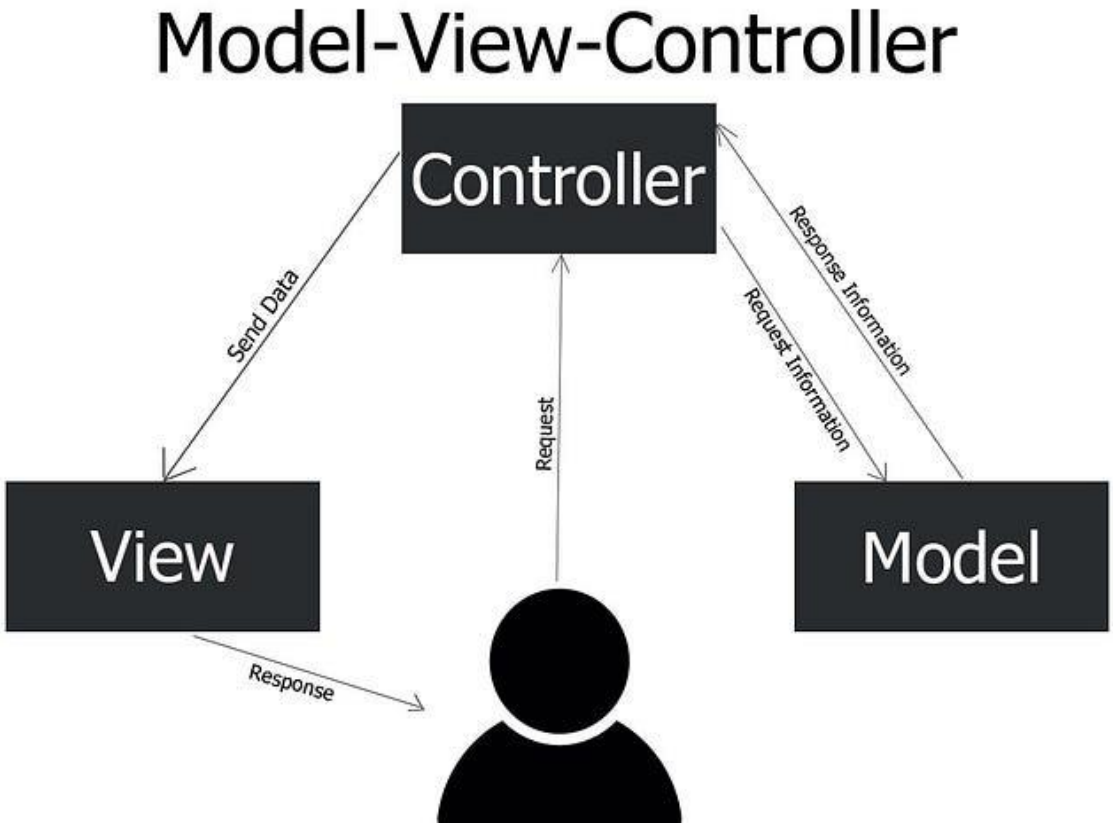


Рисунок 3.3 – Архітектура MVC

Деталізуємо рисунок 3.3.

Модель (Model):

– роль: відповідає за бізнес-логіку та управління даними. Моделі взаємодіють із базою даних або будь-яким іншим джерелом даних;

– функції: зберігає дані, правила та бізнес-логіку, відповідає за запити, зміни та обробку даних;

– незалежність: моделі повинні бути незалежними від виду і контролерів, що дозволяє легко модифікувати або замінювати одну модель на іншу без змін у інших частинах програми.

Вид (View):

– роль: представляє інтерфейс користувача (UI). Вид відображає дані, які отримує від моделі, у форматі, придатному для взаємодії з користувачем;

– функції: відображає дані, які отримує від контролера, генерує вивід (наприклад, HTML);

– ізоляція: вид повинен бути ізольований від бізнес-логіки, тому зміни у виді не впливають на модель або контролер.

Контролер (Controller):

– роль: відповідає за приймання вхідних даних від користувача, їх обробку та передачу команд моделі або виду;

– функції: обробляє запити користувача, викликає відповідні функції або методи моделей, вибирає відповідний вид для відображення;

– медіатор: контролер діє як посередник між моделлю та видом, забезпечуючи їх взаємодію.

Переваги архітектури MVC:

– підтримка: легша підтримка та оновлення програми завдяки чіткому розділенню відповідальностей;

– масштабування: програму легше масштабувати та модифікувати;

– поділ ролей: розробники, дизайнери та тестувальники можуть працювати більш незалежно та ефективно;

– тестування: компоненти можуть бути тестовані окремо, що спрощує юніт-тестування.

MVC часто використовується у веб-додатках, де вид відповідає за HTML або JSON вивід, контролер обробляє HTTP запити, а модель взаємодіє з базою даних.

3.4 Опис програми за допомогою UML діаграми

Опишемо програми за допомогою UML діаграми.

UML (Unified Modeling Language) – це стандартизована мова візуального моделювання, яка широко використовується в області розробки програмного забезпечення. Вона допомагає проектувати та зрозуміло візуалізувати структуру та поведінку систем. UML діаграми використовують для опису програм, що дозволяє ефективно представити архітектуру програмного забезпечення, процеси взаємодії компонентів, а також внутрішню логіку роботи системи.

Застосування UML діаграм є важливим етапом у плануванні та документуванні різних аспектів програмних проектів. Вона допомагає командам розробників та зацікавленим сторонам краще розуміти та аналізувати проект, а також сприяє ефективній комунікації між учасниками проекту.

Програма має три компоненти, які допомагають реалізувати необхідний функціонал (рисунки 3.4 – 3.6):

- для збору інформації,
- для актуалізації даних,
- пошуку статей.

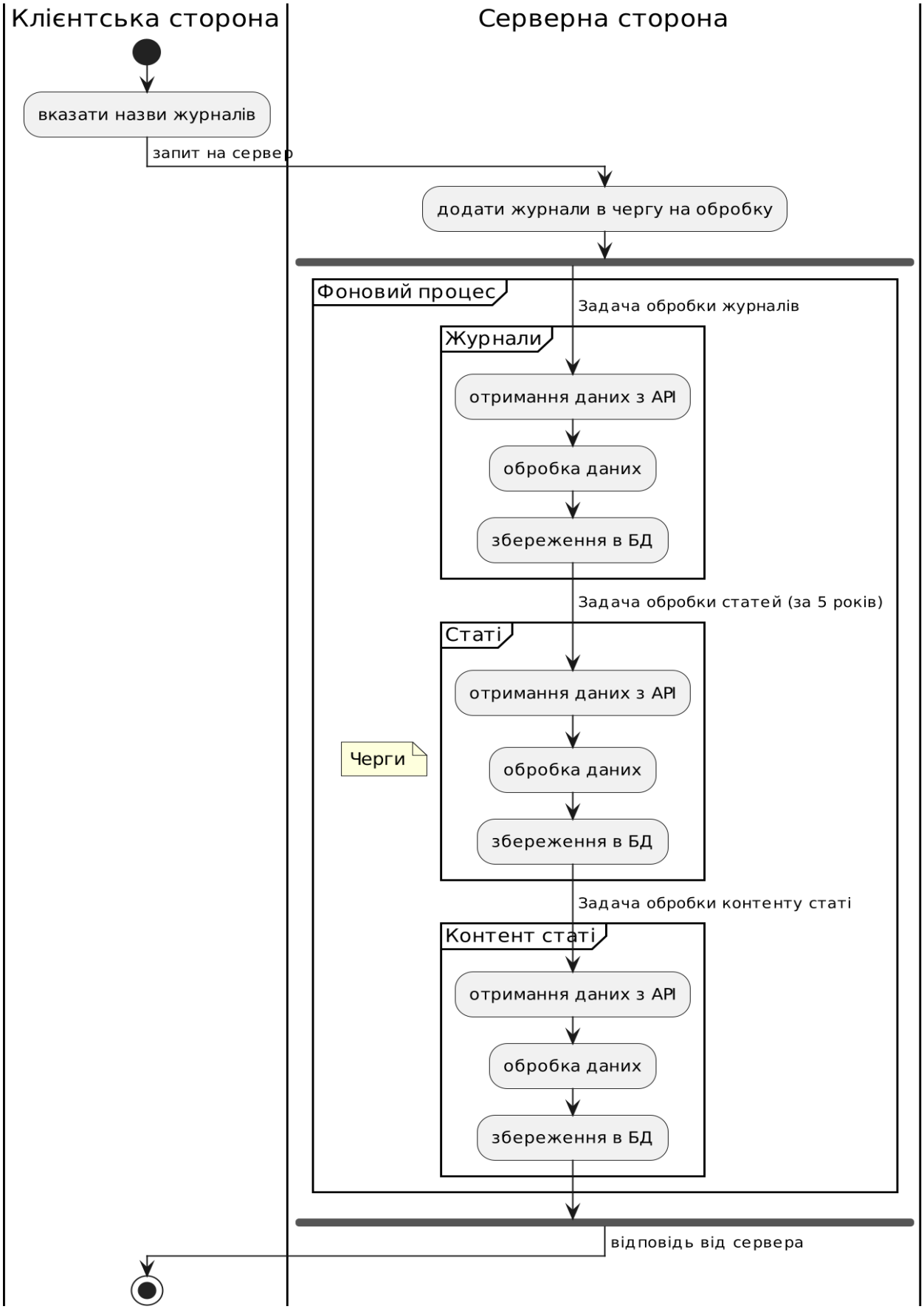


Рисунок 3.4 – Компонент для збору інформації за 5 років

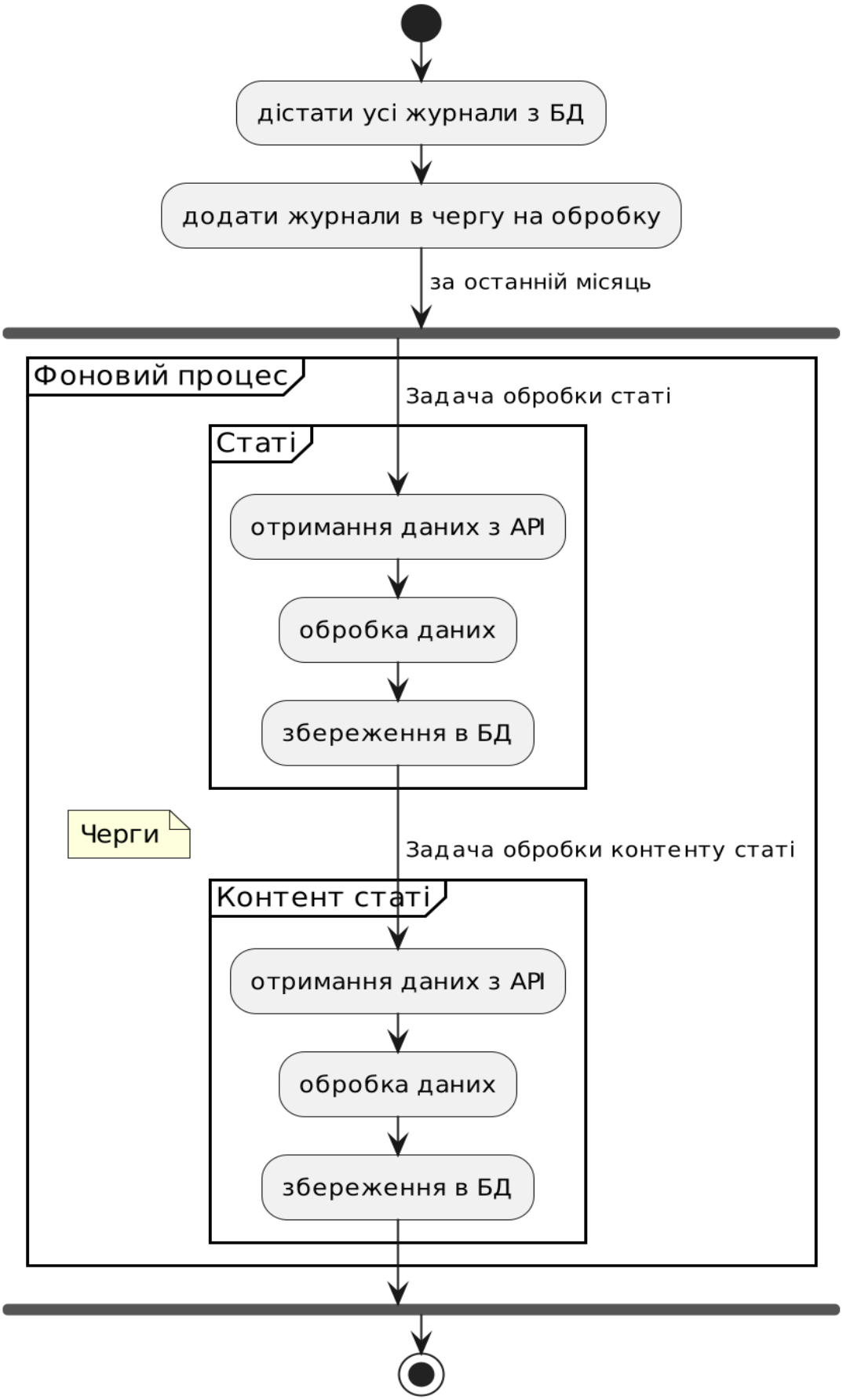


Рисунок 3.5 – Компонент для актуалізації даних (за допомогою розкладу - CRON)

Рисунок 3.6 – Компонент пошуку статей

3.5 Структура бази даних MySQL (датологічна модель)

Розглянемо структуру бази даних MySQL (датологічна модель).

Структура MySQL (датологічна модель) (рисунок 3.7) включає кілька ключових елементів:

1. Таблиці: Основні елементи бази даних, де зберігаються дані. Кожна таблиця містить один або декілька стовпців та рядків. Стовпці містять типи даних та описують характеристику даних, що зберігаються, а рядки - це фактичні дані.

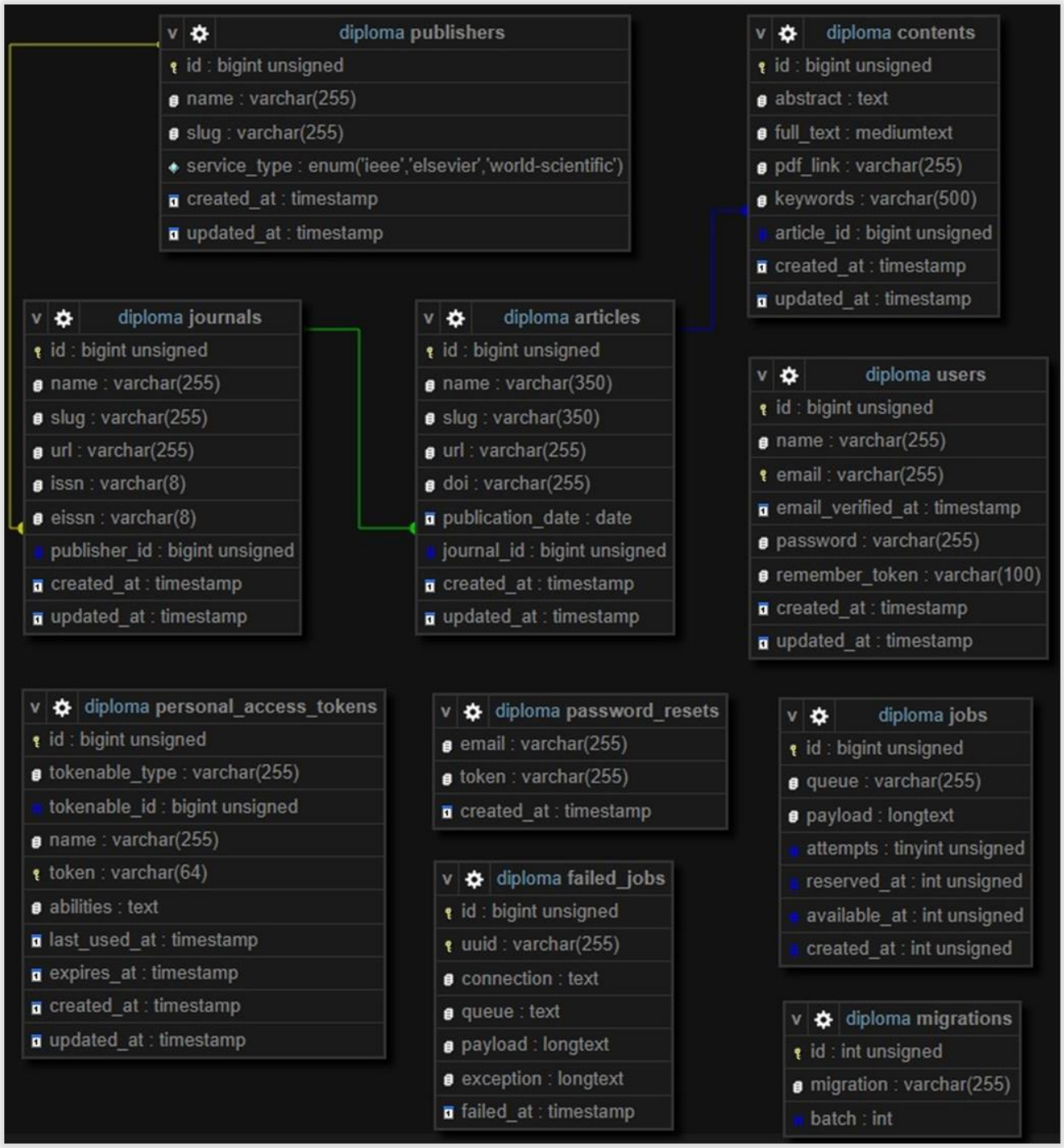


Рисунок 3.7 – Датологічна модель бази даних проекту

2. Поля (стовпці): Кожне поле у таблиці має визначений тип даних (наприклад, INT, VARCHAR, DATE) і може мати обмеження (наприклад, NOT NULL, UNIQUE).

3. Рядки: Це горизонтальні записи у таблиці, що представляють одиницю інформації. Кожен рядок має одне або декілька полів.

4. Первинний ключ (Primary Key): Унікальний ідентифікатор для кожного рядка в таблиці. Первинний ключ забезпечує унікальність запису та може бути використаний для зв'язування з іншими таблицями.

5. Зовнішній ключ (Foreign Key): Використовується для зв'язування двох таблиць. Зовнішній ключ у одній таблиці відсилає до первинного ключа іншої таблиці.

6. Індекси: Це спеціальні структури, які допомагають прискорити доступ до даних. Індекси можуть бути створені для одного або декількох полів у таблиці.

7. Зв'язки: Визначають відношення між таблицями. Основні типи зав’язків – один-до-одного, один-до-багатьох, багато-до-багатьох.

8. Нормалізація: Процес організації даних у базі даних. Нормалізація включає розділення великих таблиць на менші та зменшення повторюваності даних.

9. Збережені процедури та тригери: Це скрипти, які виконуються в базі даних для автоматизації певних процесів або для реагування на певні події.

Опишемо основні таблиці та поля які використовуються для збереження інформації:

– publishers – таблиця видавництва, які займаються публікацією наукових робіт та матеріалів;

– journals – таблиця журналів для яких виконується пошук та збереження статей у відкритому доступі:

  – *name*: заголовок (назва) журналу;
  – *url*: посилання на журнал на сайті видавництва;
  – *issn* (international standard serial number): для серійних видань;
  – *eissn* (electronic ISSN): для електронних (цифрових) версій журналів;

– articles – таблиця статей які були знайдені у відкритому доступі для конкретного журналу:

  – *name*: заголовок (назва) статті;
  – *url*: посилання на статтю на сайті видавництва;

– *doi* (digital object identifier): для ідентифікації електронних документів (наукова стаття, книга або доповідь);

– *publication_date*: дата коли стаття була опублікована;

– contents – таблиця в якій зберігається зміст кожної статті:

– *abstract*: короткий огляд або резюме основного змісту;

– *full_content*: зміст статті, лише текст отриманий від видавництва за допомогою API;

– *pdf_link*: посилання на PDF версію статті;

– *keywords*: ключові слова які були вказані автором статті.

3.6 Інтерфейс користувача

Щоб почати працювати з функціоналом сайту, потрібно зареєструватися, а потім здійснити вхід на закритий розділ. Форма реєстрації приведена на рисунку 3.8, а форма входу – на рисунку 3.9.

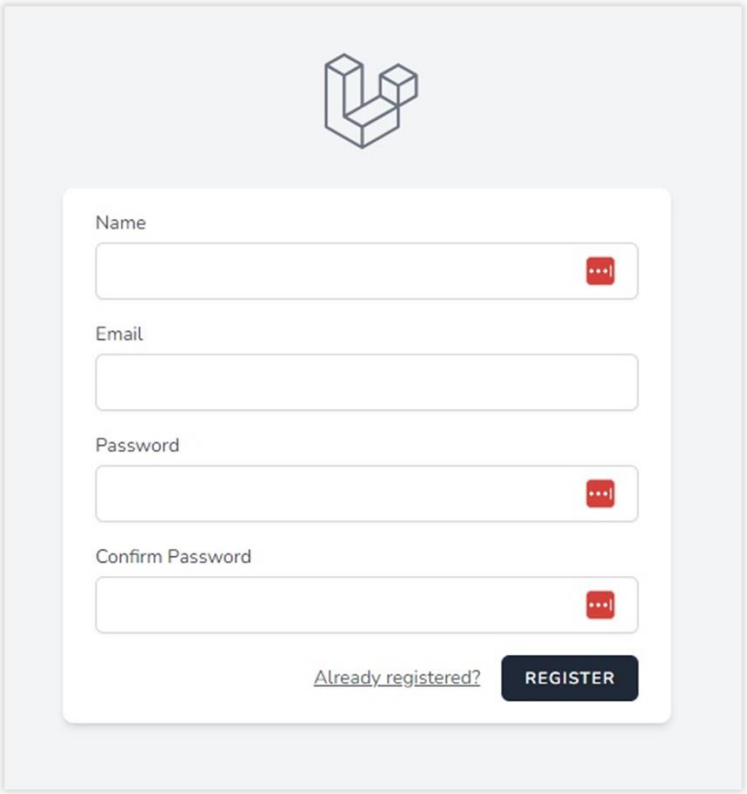


Рисунок 3.8 – Форма реєстрації

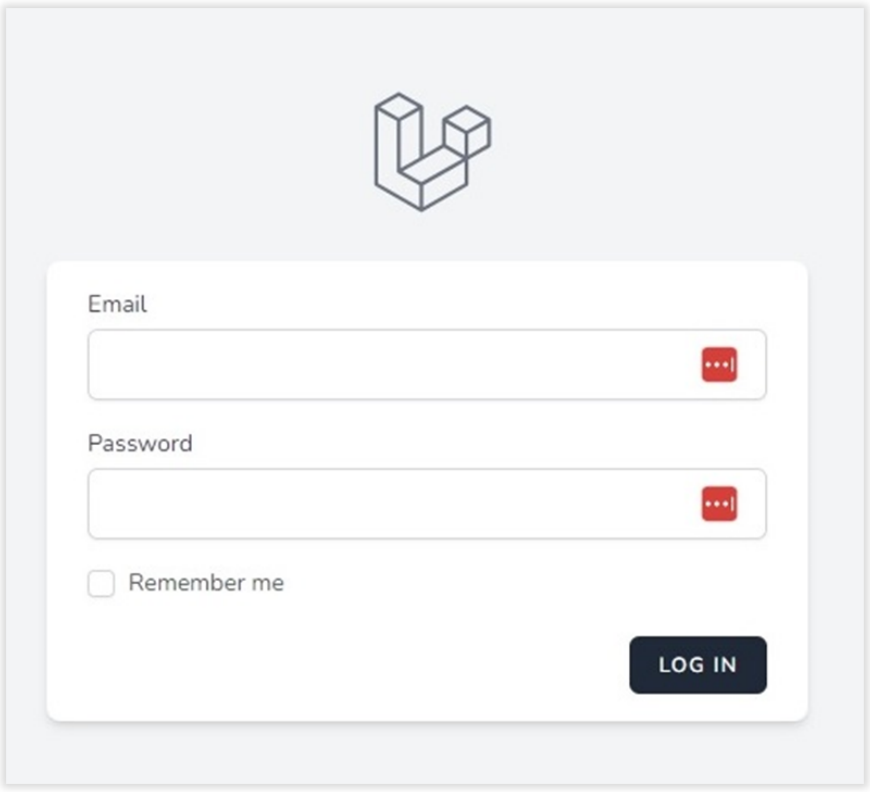


Рисунок 3.9 – Форма входу

Після входу ми бачимо головну сторінку закритого розділу (рисунок 3.10), де також можна відкрити випадаюче меню. Основні пункти це Publishers та Search.

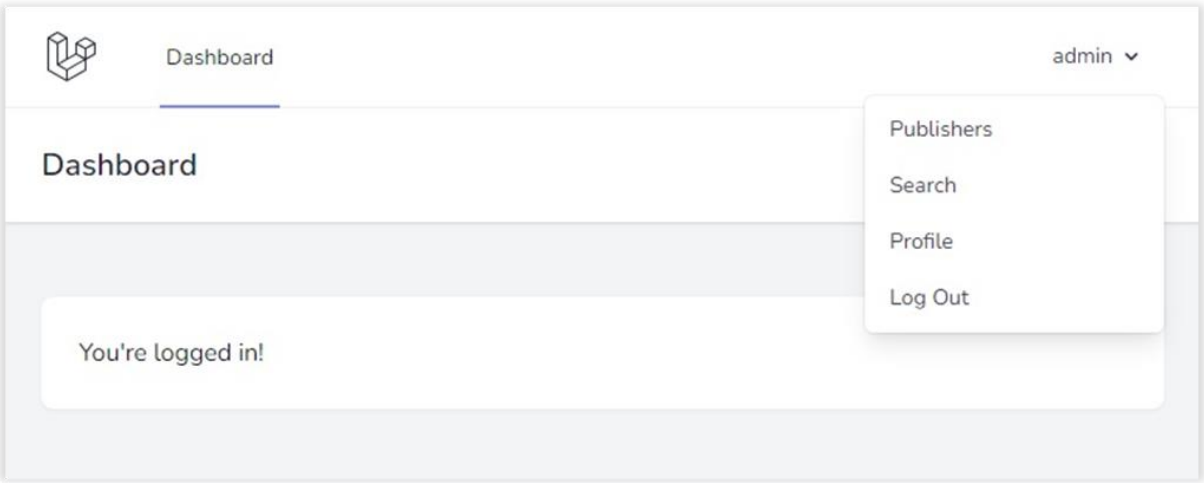


Рисунок 3.10 – Інформаційна панель закритого розділу

Якщо перейти на сторінку *Publishers* (рисунок 3.11), то можна побачити список видавництв з якими в даний момент працює система та з яких отримує дані. Загальна інформація відображає назву та кількість доданих журналів.

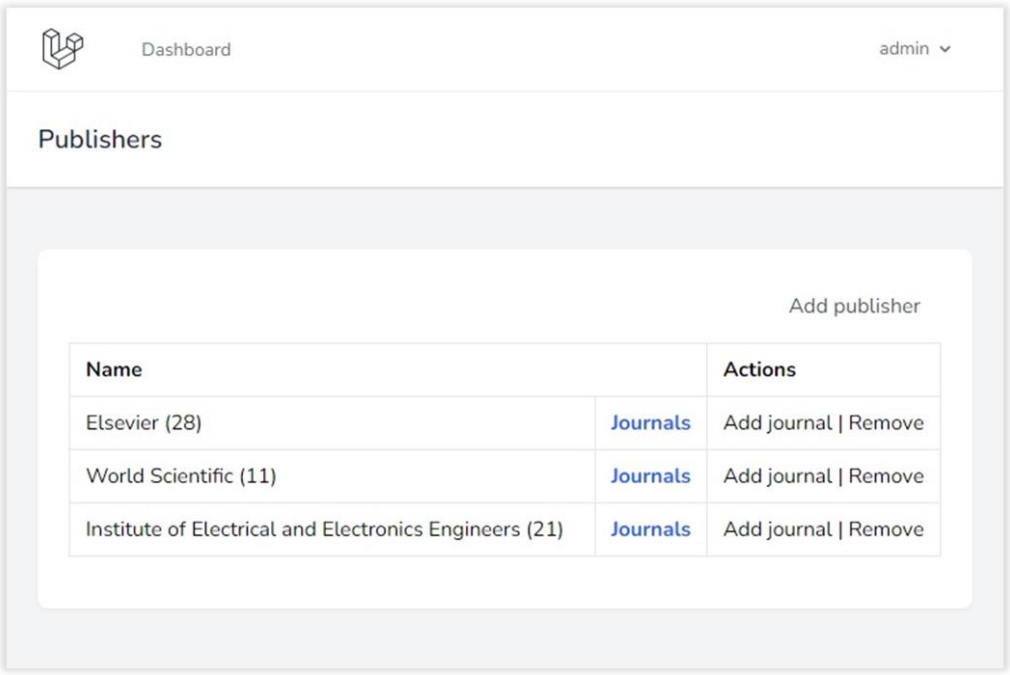


Рисунок 3.11 – Сторінка списку видавництв

Можемо як видалити додане видавництво так і додати нові журнали скориставшись кнопкою Add journal. Кожна назва повинна починатися з нової стрічки для можливості розглядати як окрему в подальшому у коді.

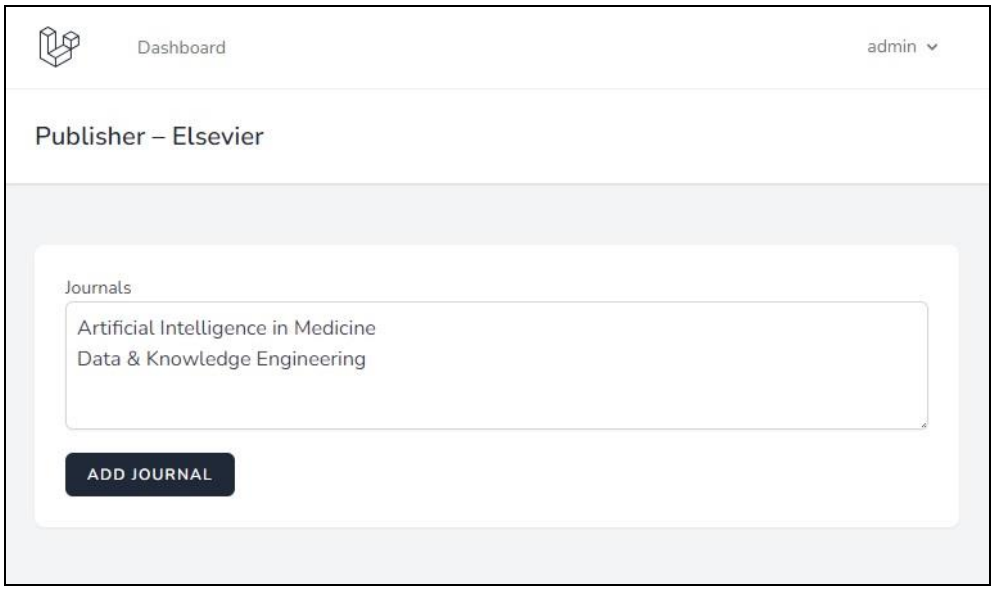


Рисунок 3.12 – Сторінка для додавання назви журналів

Коли користувач натисне кнопку Add journal відправиться запит на сторону сервера для додавання та обробки. Процес обробки відбувається у фоновому режимі один за одним. Спочатку отримується інформація по журналу, потім статті для нього, потім для кожної статті витягується корисна інформація (абстракт, посилання на PDF сторінку, повний текст статті, якщо доступно та інші.

Зі сторінки списку видавництв можна більше детально подивитися які журнали були додані, та скільки знайдено статей у відкритому доступі за певний період натиснувши на посилання Journals. Є можливість видалити журнал який вже не цікавий, перейти на сторінку журналу натиснувши на назву (рисунок 3.13).

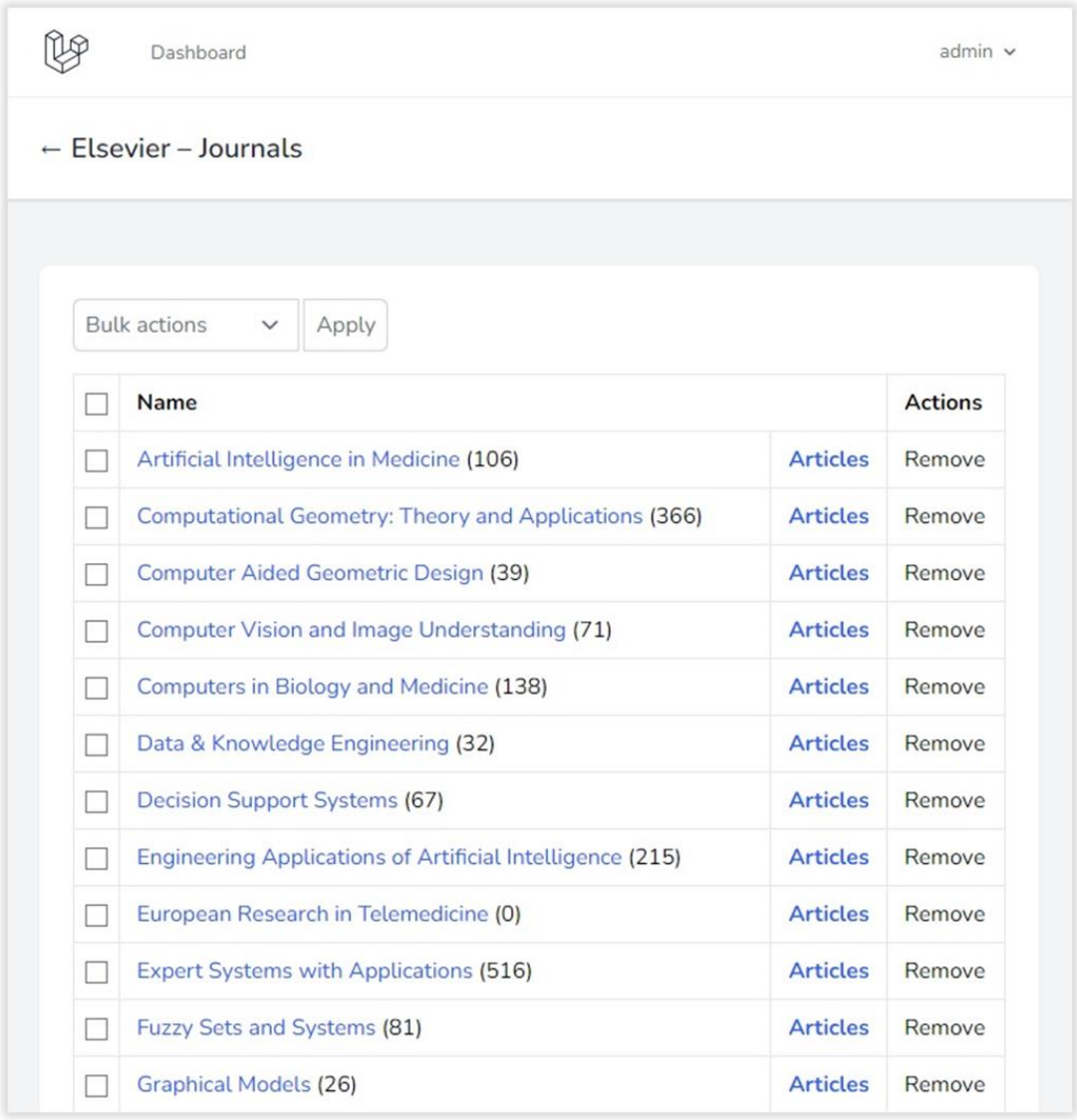


Рисунок 3.13 – Сторінка списку журналів

Щоб переглянути список статей для конкретного журналу необхідно натиснути на посилання Articles (рисунок 3.14).

Дана сторінка деяку корисну інформацію:

– назва (є посиланням на сторінку статті);

– посилання на PDF файл;

– дата публікації.

Також можна подивитися більше детальну інформацію по статті натиснувши кнопку Show, а також видалити якщо це необхідно (рисунок 3.15).

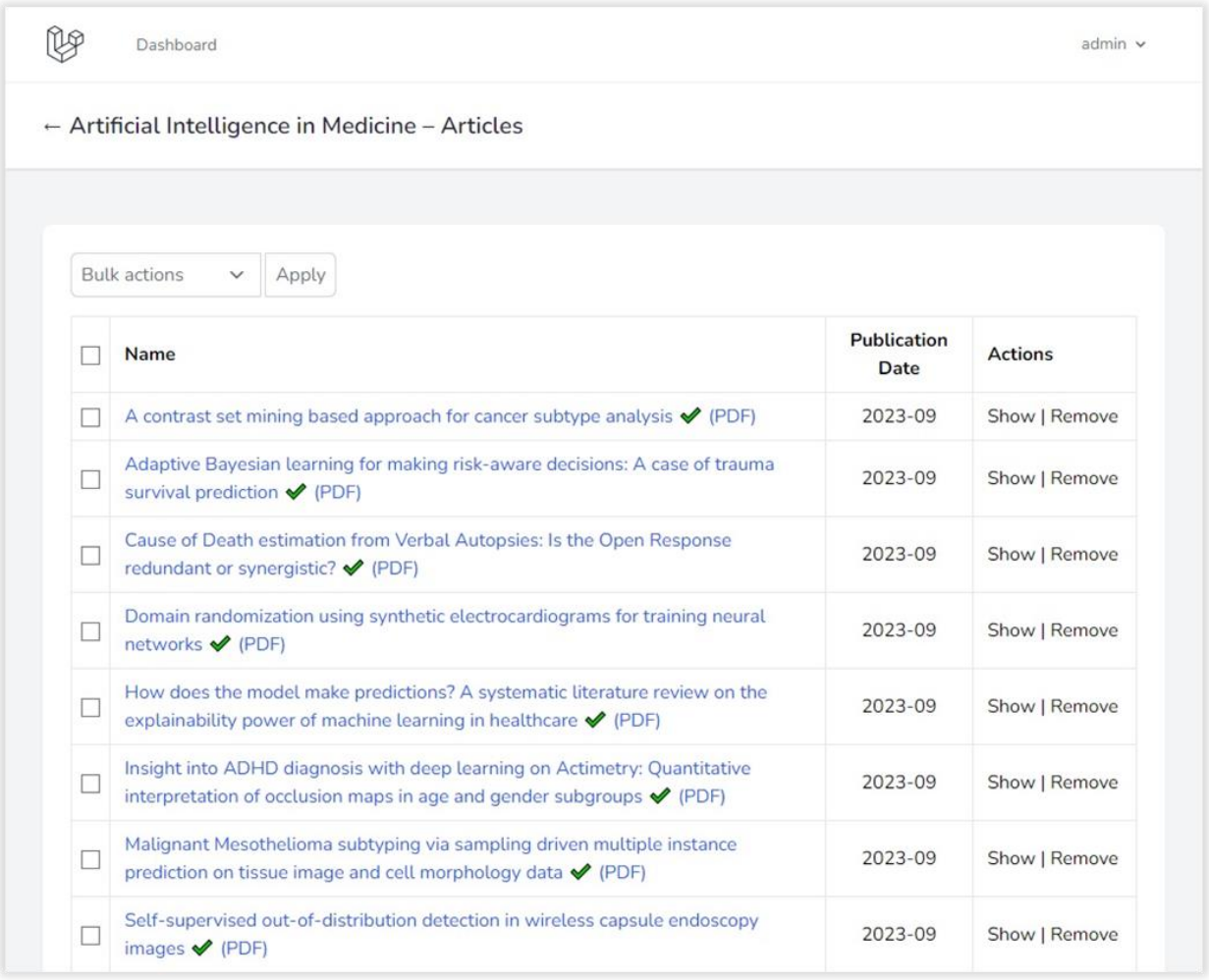


Рисунок 3.14 – Сторінка списку статей

Після того як уся необхідна інформація була отримана за допомогою API від видавництв, можна виконати пошук по статтях за допомогою ключових слів (рисунок 3.16).

Як видно з рисунка ми можемо виконувати пошук і фільтрацію за наступними полями:

– From: початкова дата, обов’язкове поле;

– To: кінцева дата, обов’язкове поле;

– Journal name: будуть знайдені статі які відносяться до вибраного журналу, опціонально. Якщо не вибрано, то по пошук буде відбуватися включаючи усе;

– Keywords: ключові слова, поле обов’язкове для заповнення;

– Search by: поле в якому виконується пошук (заголовок або контент статті).

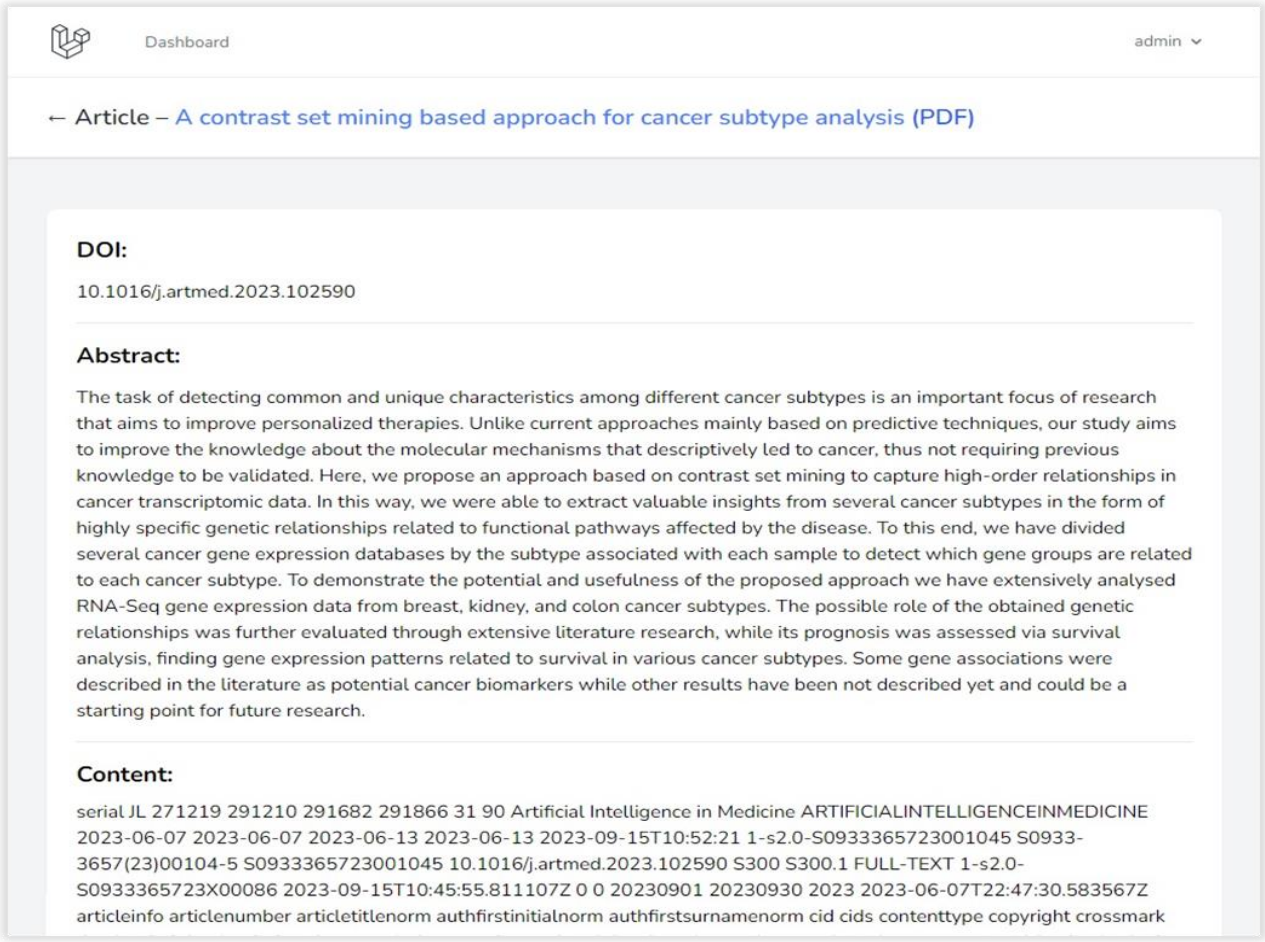


Рисунок 3.15 – Сторінка статті з інформацією

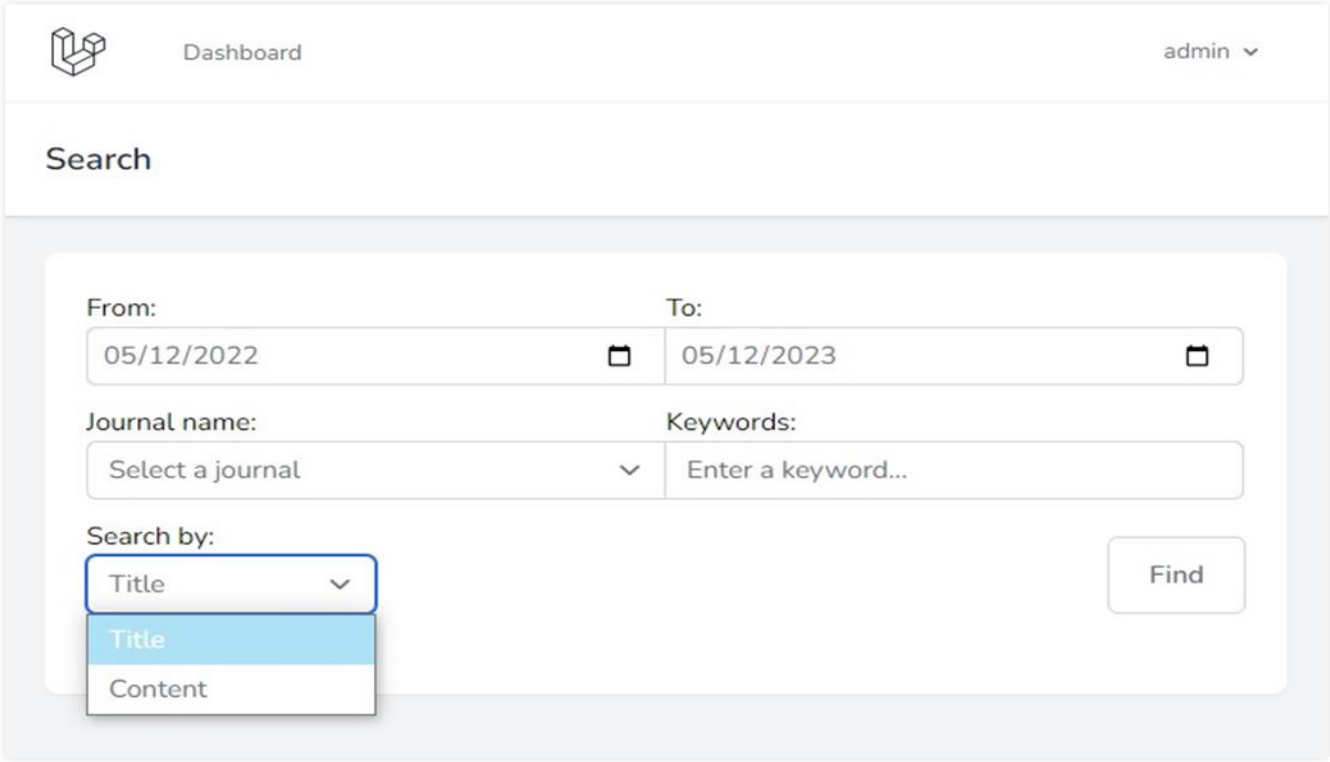


Рисунок 3.16 – Сторінка пошуку статей за ключовими словами

Результат пошуку можна експортувати у файл BibTeX формату *.bib, а потім імпортувати файл в JabRef (системою управління бібліографічною інформацією) для подальшої роботи та систематизації.

3.7 Тестування (приклади використання)

Тестування функціоналу проводиться з інформацією яка були збережена до бази даних MySQL. Розглянемо три випадки:

– фільтрація результатів за допомогою ключових слів у назві статті;

– фільтрація результатів за допомогою ключових слів в змісті статті;

– фільтрація результатів по назві журналу та ключових слів у (назві/змісті) статті.

Виставляємо критерії для фільтрації результатів які збираємося отримати і отримуємо результат, як це показано на рисунках 3.17 – 3.19.

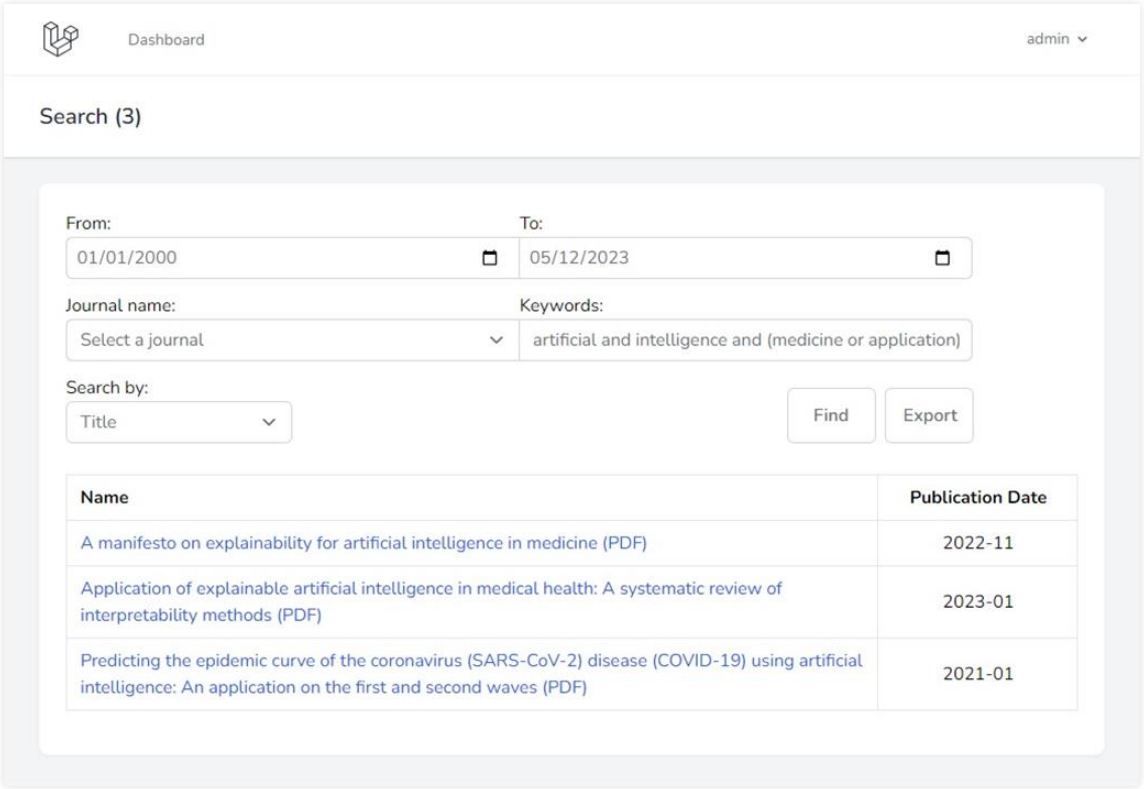


Рисунок 3.17 – Результат пошуку за ключовими словами у назві статті

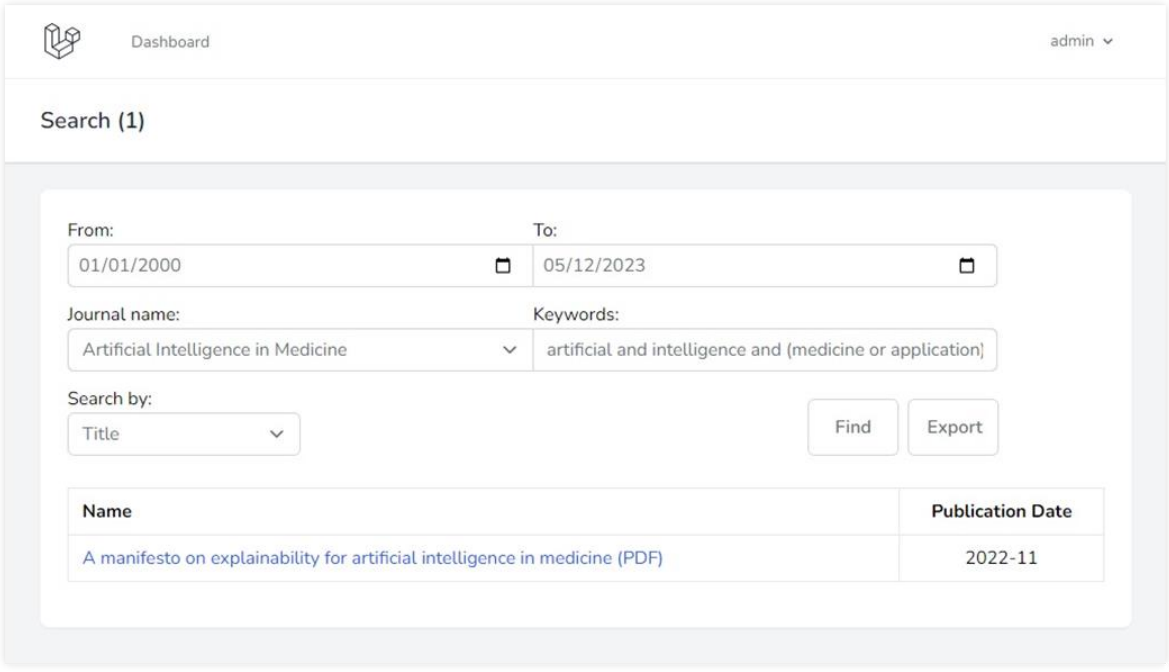


Рисунок 3.18 – Результат пошуку статей для конкретного журналу за ключовими словами у назві

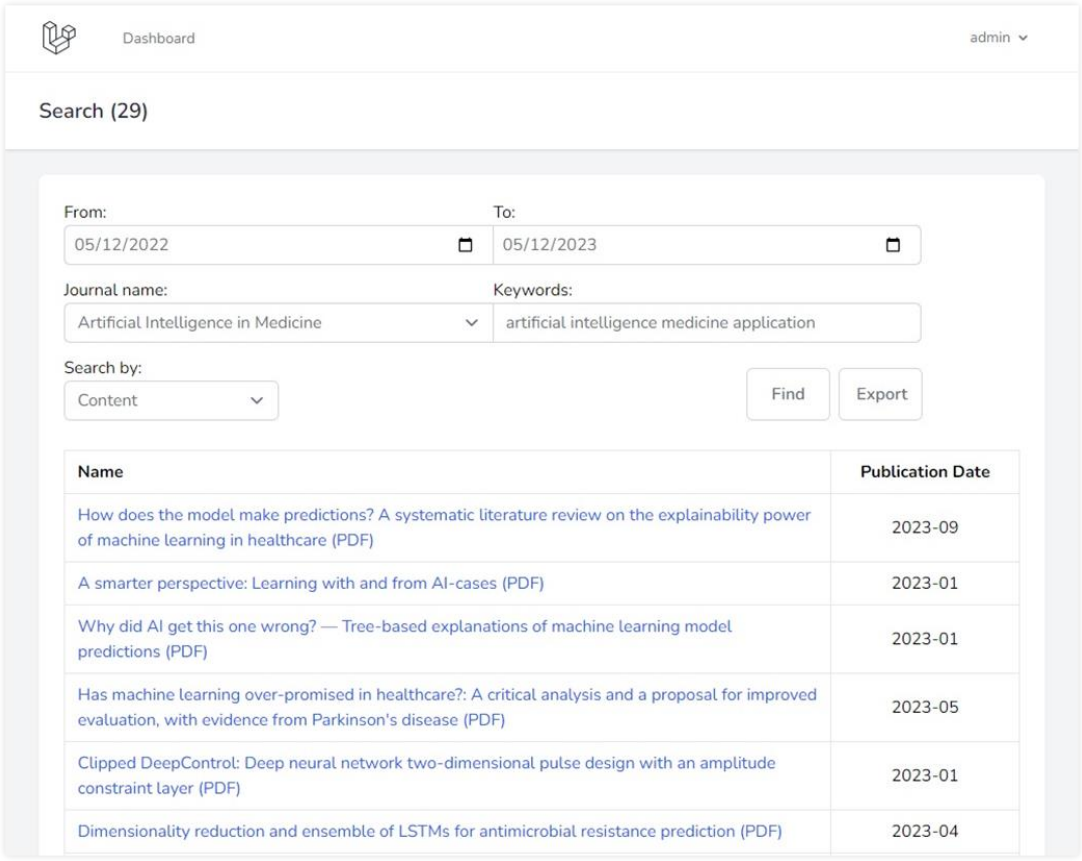


Рисунок 3.19 – Результат пошуку статей для конкретного журналу за ключовими словами у змісті

Результат можна експортувати у файл BibTeX формату *.bib натиснувши кнопку Export. Якщо працюєте до прикладу з JabRef то скачаний файл export.bib потім можна імпортувати в програму та отримати подібний результат.

Кроки для імпорту файлу *.bib показані на рисунках 3.20 – 3.22.

3.8 Опис інтерфейсу програмування застосунків (API) видавництв

У цьому розділі представлений опис (API) видавництв Elsevier та IEEE для отримання різноманітної інформації, пов'язаної з журналами та статтями. Основні URL-адреси (endpoints), які використовувалися для отримання даних про журнали, списку статей для конкретного журналу, а також абстрактів та контенту статей. Посилання на документації по Elsevier та IEEE.

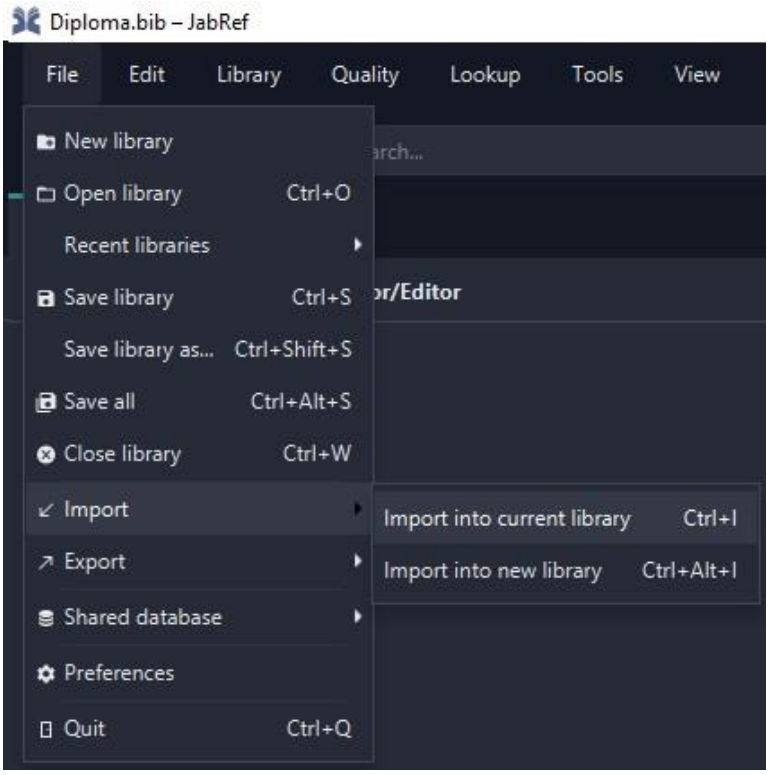


Рисунок 3.20 – Імпорт в існуючу бібліотеку

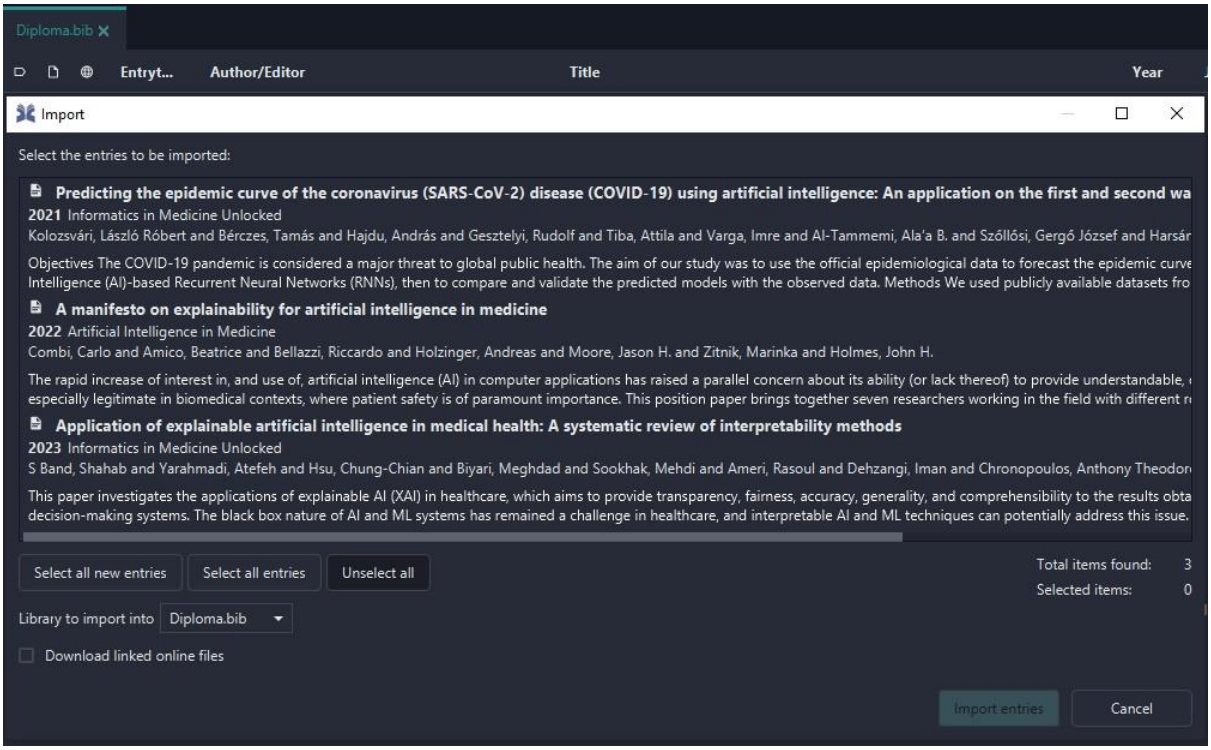


Рисунок 3.21 – Вікно імпорту статей до системи

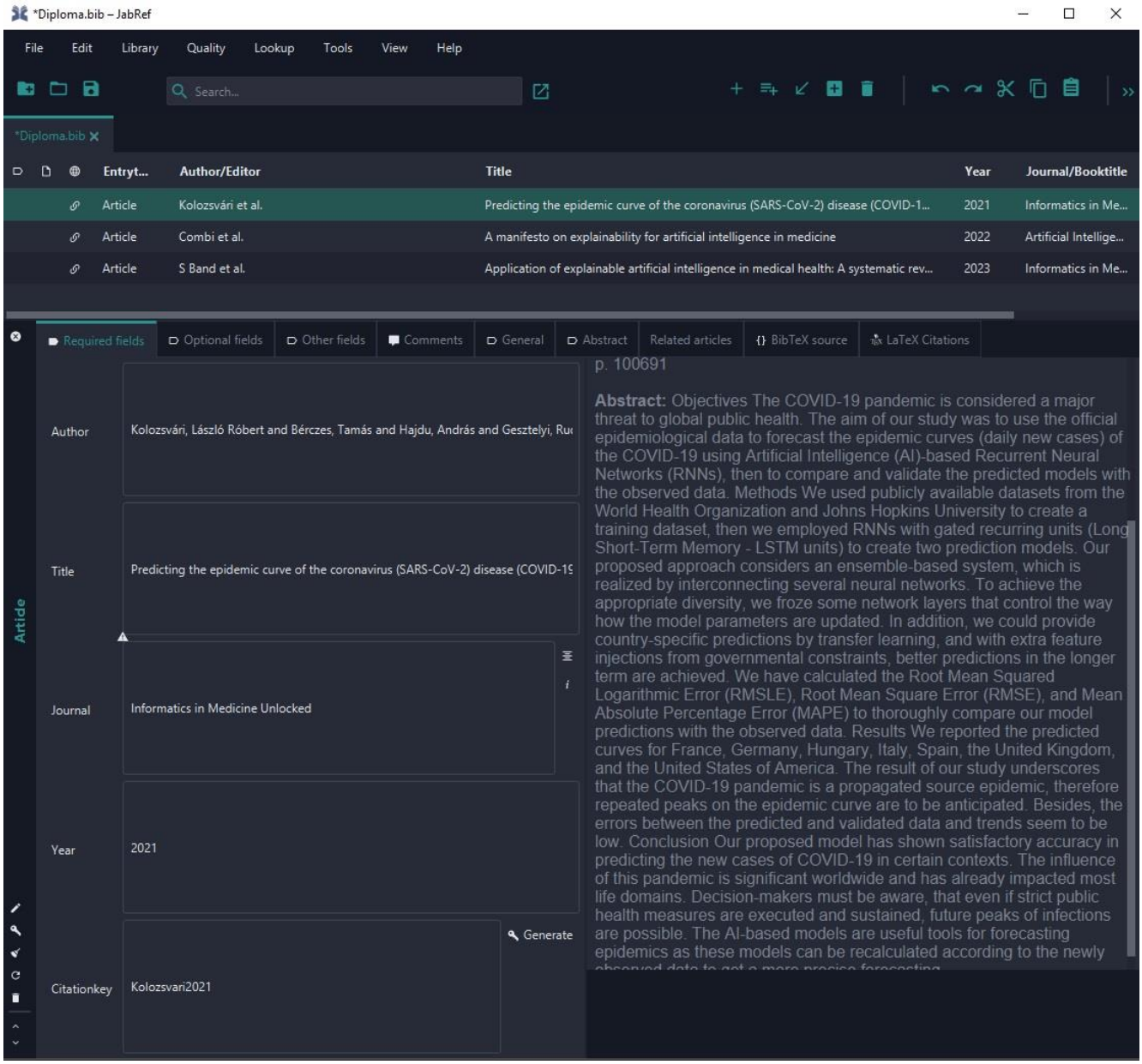


Рисунок 3.22 – Список імпортованих статей та інтерфейс користувача в JabRef

Пошук та отримання інформації за назвою журналу:

– Elsevier:
  – Метод: GET ;
  – URL: https://api.elsevier.com/content/serial/title;
  – Параметри запиту:
    – title – назва журналу;
    – apiKey – унікальний ключ, забезпечує доступ до ресурсу;
    – httpAccept – формат відповіді (application/json);
– IEEE:

- Метод: GET;
- URL: https://ieeexploreapi.ieee.org/api/v1/search/articles;
- Параметри запиту:
  - apikey – унікальний ключ, забезпечує доступ до ресурсу;
  - format – формат відповіді (json);
  - publication_title – назва журналу;
  - content_type – тип контенту (Journals,Magazines);

Пошук та отримання списку статей для журналу по issn/eissn:

- Elsevier:
  - Метод: GET;
  - URL: https://api.elsevier.com/content/search/scopus;
  - Параметри запиту:
    - apiKey – унікальний ключ, забезпечує доступ до ресурсу;
    - httpAccept – формат відповіді (application/json);
    - query – представляє логічний пошук (SRCTYPE(j) AND DOCTYPE(ar) AND OPENACCESS(1) AND ISSN(значення OR інше значення)):
      - SRCTYPE(J) – повертає документи з джерел журналу;
      - DOCTYPE(ar) – повертає документи, класифіковані як статті;
      - OPENACCESS(1) – повертає вміст який у відкритому доступу;
      - ISSN(значення) – унікальний ідентифікаційний номер, який присвоюється серійним виданням;
    - field – назви полів, які потрібно повернути у відповіді;
    - date – діапазон дат для фільтрації записів (приклад 2013-2023);
- IEEE:
  - Метод: GET;
  - URL: https://ieeexploreapi.ieee.org/api/v1/search/articles;
  - Параметри запиту:
    - apikey – унікальний ключ, забезпечує доступ до ресурсу;
    - format – формат відповіді (json);
    - content_type – тип контенту (Journals, Magazines);
    - issn – міжнародний стандартний серійний номер журналу;

– open_access – повертає документи у відкритому доступі (True);

– start_year – початкове значення року публікації для обмеження результатів;

– end_year – кінцеве значення року публікації для обмеження результатів.

Пошук та отримання деталей статі по DOI:

– Elsevier:

– Метод: GET;

– URL: https://api.elsevier.com/content/article/doi/{doi}:

– {doi} – унікальний ідентифікатор цифрового об’єкта який CrossRef призначає статті/документу;

– Параметри запиту:

– apiKey – унікальний ключ, забезпечує доступ до ресурсу;

– httpAccept – формат відповіді (application/json);

– IEEE:

– Метод: GET;

– URL: https://ieeexploreapi.ieee.org/api/v1/search/articles;

– Параметри запиту:

– apikey – унікальний ключ, забезпечує доступ до ресурсу;

– format – формат відповіді (json);

– doi – унікальний ідентифікатор цифрового об’єкта який CrossRef призначає статті/документу.

CrossRef (Crossref) – це міжнародна некомерційна організація, яка надає послуги реєстрації цифрових об'єктів, таких як наукові статті, журнали, книги та інші видання. Головною метою CrossRef є підтримка наукової спільноти шляхом створення стабільних та унікальних ідентифікаторів (DOIs – Digital Object Identifiers) для наукових публікацій.

3.9 Висновки до розділу 3

У розділі описано системні вимоги до функціонування прикладного програмного забезпечення: персональний ноутбук з чотирьох ядерним процесором 11th Gen Intel Core i7-1185G7 із тактовою частотою 3.00 ГГц; операційна система Windows 10 x64. Описано пошукову систему, яка реалізована на мові PHP v8.3. Для розробки використано PHP фреймворк веб-додатків Laravel v9.x., який побудований на MVC.

Розроблено даталогічну модель бази даних публікацій світових наукометричних баз. Розроблено інтерфейс пошукової системи та проведено тестування на прикладі журналів світових видавництв.

ВИСНОВКИ

У дипломній роботі проаналізовані моделі та відомі наукометричні бази і отримано такі результати:

1.Проведено аналіз наукометричних баз даних: великих сховищ наукових публікацій, цитувань тощо.

2. Проаналізовано такі агрегатори повних текстів: Scopus, Web of Science, IEEE Xplore, Google Scholar, що дало можливість виділити основні операції при пошуку інформації.

3. Проаналізовані класичні моделі пошуку інформації і показано, що булева модель не розпізнає часткові збіги, що призводить до низької продуктивності.

4. Імовірнісна модель забезпечує кращу ефективність пошуку. Векторна модель перевершує імовірнісну модель із загальними наборами, що пояснюється її поширеністю в реальних пошукових системах.

5. Розроблено пошукову систему, яка реалізована на мові PHP v8.3. Для розробки використано PHP фреймворк веб-додатків Laravel v9.x., який побудований на технології MVC, що дало можливість здійснювати пошук та формувати базу даних наукових публікацій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методи, алгоритми та програмні засоби опрацювання біомедичних зображень / Березький О. М. та ін. Тернопіль: Економічна думка, ТНЕУ, 2017. 330 с.

2. Березький О. М., Батько Ю.М., Мельник Г.М. Методи сегментації біомедичних зображень. Вісник Хмельницького національного університету. Технічні науки. 2010. №1. С.189- 197.

3. Berezsky O., Verbovyy S., Pitsun O. Hybrid Intelligent information technology for biomedical image processing. Proceedings of the IEEE International Conference «Computer Science and Information Technologies» CSIT'2018, Lviv. Ukraine. 11-14 September, 2018. P. 420-423.

4. Березький О. М., Мельник Г.М., Батько Ю.М., Дацко Т. В. Інтелектуальна система для діагностування різних форм раку молочної залози на основі аналізу гістологічних і цитологічних зображень. Науковий вісник НЛТУ України: зб. наук.-техн. праць. Львів: РВВ НЛТУ України. 2013. Вип. 23.13. С. 357-367.

5. Березский О. Н. Топологические методы и алгоритмы преобразования контуров и областей плоских изображений. Проблемы информатики и управления. 2010. № 5. С.123-131.

6. Березький О. М. Методи та алгоритми перетворення контурів зображень в афінному просторі. Вісник Національного університету «Львівська політехніка». Комп'ютерні науки та інформаційні технології. 2009. № 638. С. 185-189.

7. Березький О. М., Батько Ю.М., Мельник Г.М. Комп'ютерна система аналізу біомедичних зображень. Вісник Національного університету «Львівська політехніка». Комп'ютерні науки та інформаційні технології. 2009. № 570. С. 84-89.

8. Berezsky O., Melnyk G., Batko Yu. Biomedical image search and retrieval algorithms. International Journal of Computing. 2008. Vol. 7, Issues 1. P.108-113.

9. Березький О. М., Березька К. М., Попіна С. Ю. Статистичне оброблення цитологічних зображень. Вісник Хмельницького національного університету: зб. наук.-техн. праць. Сер.: Технічні науки. 2012. № 5. С. 161–164.

10. Грицик В.В., Березька К.М., Березький О. М. Моделювання та синтез складних зображень симетричної структури. Львів: УАД – ДНДІІІ, 2005. 140 с.

11. Кудінов Ф.В., Каліновський Р.М. Алгоритми пошуку інформації. *Інтелектуальні комп'ютерні системи та мережі* : тези доп. VIII Наук.-практ. конф. молодих вчених і студентів (5 грудня 2023 р.). Тернопіль : ЗУНУ, 2023. С. 71.

12. Кудінов Ф.В., Каліновський Р.М. Призначення і функції наукометричних баз. *Інтелектуальні комп'ютерні системи та мережі* : тези доп. VIII Наук.-практ. конф. молодих вчених і студентів (5 грудня 2023 р.). Тернопіль : ЗУНУ, 2023. С. 72.

13. Березький О. М., Дубчак Л. О., Мельник Г. М. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня "Магістр". Спеціальність: 123 – Комп'ютерна інженерія. Магістерська програма – «Комп'ютерна інженерія». Тернопіль : ЗУНУ, 2021. 32 с.

14. Shi X., Leskovec J., McFarland D. A. Citing for high impact. Proceedings of the 10th Annual Joint Conference on Digital Libraries, Gold Coast, Queensland, Australia. New York, NY, USA, 2010. P. 49–58.

15. Saha S., Saint S., Christakis D. A. Impact factor: a valid measure of journal quality?. Journal of the Medical Library Association: JMLA. 2003. Vol. 91, no. 1. P. 42-46.

16. List of academic databases and search engines. URL: https://en.wikipedia.org/wiki/List_of_academic_databases_and_search_engines.

17. Scopus. wikipedia.org. URL: https://en.wikipedia.org/wiki/Scopus.

18. Scopus Search. Elsevier. URL: https://www.elsevier.com/products/scopus/search.

19. Scopus® Quick reference guide. URL: https://supportcontent.elsevier.com/RightNow%20Next%20Gen/Scopus/Files/Scopus_User_Guide.pdf.

20. Internet Archive Scholar. URL: https://scholar.archive.org.

21. Command Search. IEEE Xplore Resources and Help. IEEE Xplore. URL: https://ieeexplore.ieee.org/Xplorehelp/searching-ieee-xplore/command-search.

22. International Classification for Standards. wikipedia.org. URL: https://en.wikipedia.org/wiki/International_Classification_for_Standards.

23. Valenzuela-Escarcega M. A., Ha V. A., Etzioni O. Identifying Meaningful Citations. AAAI Workshop: Scholarly Big Data. 2015. URL: https://api.semanticscholar.org/CorpusID:2538517.

24. Reese J. P. Walker Library: Effective Search Strategies: Boolean Operators. 2015.

25. Computer aided document indexing system / M. Kolar et al. 27th International Conference on Information Technology Interfaces. 2005. P. 323-328.

26. Sadat M., Caragea C. Hierarchical Multi-Label Classification of Scientific Documents. 2022. URL: Режим%20доступу:%20ArXiv%20abs/2211.02810.

27. Gottardi T., Medeiros C. B., dos Reis J. C. Semantic Search on Scientific Repositories: A Systematic Literature Review. Brazilian Symposium on Databases. 2020.

28. The correlation between scientific collaboration and citation count at the paper level: a meta-analysis / H. Shen et al. Scientometrics. 2021. Vol. 126. P. 3443-3470.

29. A Deep Learning Approach for Scientific Paper Semantic Ranking / F. Gargiulo et al. International Conference on Intelligent Interactive Multimedia Systems and Services. 2018.

30. Wang X., Harmelen F. V., Huang Z. Ontology-based Methods for Classifying Scientific Datasets into Research Domains: Much Harder than Expected. International Conference on Knowledge Discovery and Information Retrieval. 2020.

31. Ahmed R. F. M., Salama C. R., Mahdi H. M. K. Clustering Research Papers Using Genetic Algorithm Optimized Self-Organizing Maps. 2020 15th International Conference on Computer Engineering and Systems (ICCES). 2020. P. 1-6.

32. Marcos S., García-Peñalvo F. J. Information retrieval methodology for aiding scientific database search. Soft Computing. 2018. Vol. 24. P. 5551-5560.

33. Kovalchuk O., Banakh S., Masonkova M., Berezka K., Mokhun S., Fedchyshyn O. Text Mining for the Analysis of Legal Texts. Proceedings of the 12th International Conference on Advanced Computer Information Technologies (ACIT), Ruzomberok, Slovakia, 26-28 September, 2022. P. 502-505. https://ieeexplore.ieee.org/abstract/document/9913169.

34. Sierepeklis O., Cole J. M. A thermoelectric materials database auto-generated from the scientific literature using ChemDataExtractor. Scientific Data. 2022. Vol. 9. URL: https://api.semanticscholar.org/CorpusID:253064769.

35. Xiang H. X. Query optimization over a heterogeneously distributed scientific database. 2013 IEEE International Conference on Big Data. 2013. P. 58-64. URL: https://api.semanticscholar.org/CorpusID:7207720.

36. Comparative Toxicogenomics Database (CTD): update 2023 / A. P. Davis et al. Nucleic Acids Research. 2022. Vol. 51. P. D1257 - D1262. URL: https://api.semanticscholar.org/CorpusID:252566721.

37. A standardized citation metrics author database annotated for scientific field / J. P. A. Ioannidis et al. PLoS Biology. 2019. Vol. 17. URL: https://api.semanticscholar.org/CorpusID:199548006.

38. A fast method for identifying worldwide scientific collaborations using the Scopus database / F. G. Montoya et al. Telematics Informatics. 2018. Vol. 35. P. 168-185. URL: https://api.semanticscholar.org/CorpusID:46815490.

39. Гогунський В. Д., Коляда А. С., Оборський Г. О. Наукометричні бази: характеристика, можливості і завдання. Матеріали науково-методичного семінару "Шляхи реалізації кредитно-модульної системи". Одеса, 2014. С. 3–12.

40. Кільченко А. В. Бібліометричні та наукометричні системи у науково-педагогічній діяльності. Черкаський національний університет імені Богдана Хмельницького, Черкаський інститут банківської справи, Чорноморський державний університет імені Петра Могили, 2019. 231 с.

41. Відкриті цифрові системи в оцінюванні результатів науково-педагогічних досліджень / В. Ю. Биков та ін. Information Technologies and Learning Tools. 2020. Т. 75, № 1. С. 294-315.

42. Мінтій І., Вакалюк Т., Іванова С. Огляд можливостей онлайн-сервісу Lens. Тези доповідей VI Міжнародної науково-практичної конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2022), м. Черкаси, 22 черв. 2022 р. С. 96–97.

43. Open Ukrainian Citation Index. OUCI. URL: http://ouci.dntb.gov.ua (date of access: 01.11.2023).

44. All Science Journal Classification Codes -. -. URL: https://scientificresearch.in/asjc-all-science-journal-classification-codes/ (date of access: 01.11.2023).

45. Cited-by - crossref. www.crossref.org. URL: https://www.crossref.org/services/cited-by/ (date of access: 01.11.2023).

46. I4OC: Initiative for Open Citations. I4OC: Initiative for Open Citations. URL: https://i4oc.org (date of access: 01.11.2023).

47. Introduction to Information Retrieval: Scoring, term weighting, and the vector space model / Manning, Christopher D., Prabhakar Raghavan, Hinrich Schütze. Cambridge: Cambridge University Press., 2008. 482 p.

48. Search Engines - Information Retrieval in Practice. / Croft W. Bruce, Donald Metzler, Trevor Strohman. 2009.

49. Information Retrieval – Algorithms and Heuristics, Second Edition. / Grossman, David A. And Ophir Frieder.// The Kluwer International Series on Information Retrieval. 2004.

50. A data science-based framework to categorize academic / Z. Halim, Shafaq Khan // Scientometrics.2019, 119. P. 393-423.

51. Scopus 1900–2020: Growth in articles, abstracts, countries, fields, and journals / M. Thelwall, Pardeep Sud // Quantitative Science Studies.2021. 3. P.37-50.

52. Knowledge Repository Structure of an Experimental Software Engineering Environment / Vitor Pires Lopes, G. Travassos // 2009 XXIII Brazilian Symposium on Software Engineering. 2009. P. 32-42.