https://doi.org/10.31891/2307-5732-2025-353-9
УДК 004.912:004.42

БЕРЕЗЬКИЙ ОЛЕГ
Західноукраїнський національний університет
https://orcid.org/0000-0001-9931-4154
e-mail: ob@wunu.edu.ua
МЕЛЬНИК ГРИГОРІЙ
Західноукраїнський національний університет
https://orcid.org/0000-0003-0646-7448
e-mail: mgm@wunu.edu.ua
КУДІНОВ ФЕДІР
Західноукраїнський національний університет
e-mail: grade77@ukr.net
ПОВОРОЗНИК ВІТАЛІЙ
Західноукраїнський національний університет
e-mail: maunder257@gmail.com

МОДЕЛІ ТА ПРОГРАМНИЙ ЗАСІБ ПОШУКУ НАУКОВО-ТЕХНІЧНОЇ ІНФОРМАЦІЇ

У роботі розглянуто актуальну проблему ефективного пошуку науково-технічної інформації в умовах стрімкого зростання обсягів публікацій та цифрових ресурсів. Проведено аналіз характеристик наукометричних баз, таких як Scopus, Web of Science та інших, з урахуванням їхніх особливостей індексації, структури метаданих та релевантності результатів пошуку. Описано відомі моделі інформаційного пошуку — булеву, векторну та імовірнісну, а також сучасні підходи, зокрема BM25, кластеризацію та тематичне моделювання. Основну увагу приділено розробленню веб-програмного засобу для автоматизованого пошуку наукових публікацій із відкритих джерел. Система реалізована на основі стеку технологій PHP, Laravel, MySQL та REST API. Забезпечено підтримку збирання, актуалізації, пошуку та експорту статей у форматі BibTeX. Представлено архітектуру, інтерфейс і функціональні компоненти: збір даних, оновлення бази та пошук. Результати тестування підтвердили ефективність системи у забезпеченні релевантного тематичного пошуку. Запропоноване рішення може стати корисним інструментом для дослідників, викладачів і студентів при роботі з науковими джерелами.

Ключові слова: наукометрична база, інформаційний пошук, релевантність документів.

BEREZSKY OLEH
MELNYK GRYGORIY
KUDINOV FEDIR
POVOROZNYK VITALIY
West Ukrainian National University

MODELS AND SOFTWARE FOR SCIENTIFIC AND TECHNICAL INFORMATION SEARCH

This article addresses the problem of efficient search, filtering, and systematization of scientific and technical information by developing an integrated software tool based on modern information retrieval models. The work explores the structure and functionality of leading scientometric databases such as Scopus, Web of Science and discusses their indexing capabilities, metadata handling, and citation-based metrics. Classical information retrieval models, including Boolean, vector space, and probabilistic models, are reviewed along with advanced ranking methods such as BM25. The system also integrates techniques such as topic modeling, clustering, and citation graph analysis to enhance search quality and result relevance. The core contribution of the paper is the design and implementation of a modular web-based information retrieval system that collects metadata and full-text content from major academic publishers Elsevier and IEEE. The system allows for keyword-based searching with filtering options (e.g., date range, journal name, content type), and supports exporting results in BibTeX format for use in reference management. The backend is implemented using PHP and the Laravel framework. The frontend leverages HTML, CSS, and JavaScript for user interaction. A MySQL database ensures structured data storage and fast query execution. The system features three functional modules: data collection (up to five years of publications), periodic data updating via scheduled tasks, and a flexible article search and export interface. Testing confirms the system's ability to streamline the discovery of relevant scientific content and improve the efficiency of research workflows. The proposed tool is particularly useful for researchers, graduate students, and librarians involved in literature reviews, citation analysis, and scientific writing.

Keywords: scientometric database, information search, document relevance.

Стаття надійшла до редакції / Received 17.04.2025
Прийнята до друку / Accepted 06.05.2025

Постановка проблеми

У сучасних умовах стрімкого зростання обсягів науково-технічної інформації особливої актуальності набуває проблема її ефективного пошуку, фільтрації, систематизації та подальшого використання. Відсутність єдиного інструменту, здатного інтегрувати джерела відкритого доступу (такі як бази даних IEEE, Elsevier, CrossRef) та забезпечити гнучкий механізм пошуку релевантних

наукових публікацій за ключовими словами, датами, назвами журналів тощо, значно ускладнює наукову та аналітичну діяльність дослідників, аспірантів і студентів.

Більшість існуючих інформаційно-пошукових систем орієнтовані на конкретні платформи або вузькі предметні області, не забезпечують належного рівня автоматизації збору, актуалізації даних та їх експорту для подальшої обробки. При цьому, необхідність швидкого й точного виявлення наукових публікацій за визначеними критеріями є ключовим етапом при підготовці наукової статті чи патентного пошуку.

Наукометричні бази даних відіграють ключову роль у сучасному цифровому середовищі наукової комунікації. Вони є централізованими системами для збору, зберігання та аналітичної обробки наукової інформації, зокрема публікацій, цитувань та супутніх метаданих. Основним призначенням таких баз є оцінювання наукового внеску, моніторинг актуальних тенденцій досліджень і підтримка ухвалення стратегічних рішень в науковій сфері.

До складу наукометричних баз входять різноманітні типи документів: статті у наукових журналах, матеріали конференцій, патенти, дисертаційні роботи, технічні звіти тощо. Ключовими властивостями наукометричних баз є цитування документів, індексація документів, показники впливовості та інші метрики.

Цитування демонструє, наскільки активно конкретна робота використовується та визнається іншими дослідниками. Інформація про цитування використовується для оцінювання впливовості авторів, окремих робіт і наукових колективів. Для перевірки індексів цитування найчастіше застосовуються платформи Scopus, Web of Science і Google Scholar.

Величина індексу цитування є критично важливою для авторів, оскільки може впливати на їх академічну репутацію та кар'єрні перспективи. Для університетів і науково-дослідних установ рівень цитованості робіт співробітників впливає на позиції у міжнародних рейтингах і можливості отримання фінансування. Водночас слід пам'ятати про певні обмеження: нові статті можуть мати низькі показники через недостатній час для цитування, а роботи у вузьких чи специфічних напрямках науки можуть бути недооцінені, попри високу якість.

До популярних аналітичних інструментів також належать h-індекс, i10-індекс, аналіз цитування і графи співавторства. Графи співавторства представляють собою спеціалізовані мережеві моделі, що ілюструють співпрацю між вченими. Ці графи складаються з вузлів-авторів та ребер - спільних публікацій. Графи співавторства дозволяють проводити кластеризацію авторів за частотою та інтенсивністю співпраці

Метадані котрі описують та класифікують статті в наукометричних базах: назва роботи, автори та їхні афіліації, анотація, ключові слова, назва журналу або конференції; дата публікації, ідентифікатор DOI, інформація щодо цитувань. Індексація за цими метаданими забезпечує ефективність пошуку інформації, що відповідає потребам користувачів.

Класифікація наукометричних баз може здійснюватися за типом джерел, науковими напрямками та типами доступу (відкритий чи комерційний). Типова архітектура наукометричних систем включає такі компоненти:

- сховище даних (бази публікацій та цитувань);
- інтерфейс для користувачів;
- аналітичні модулі, що надають доступ до метрик, графів цитування та інших засобів оцінювання наукового впливу.

Основними платформами для пошуку наукових текстів та їх метаданих є Web of Science, Scopus, IEEE Xplore, Google Scholar, CiteSeerX, DBLP, Semantic Scholar. Серед агрегаторів лідирують Internet Archive Scholar (понад 35 млн. записів), CORE і CiteSeerX. Google Scholar охоплює найбільшу кількість метаданих (близько 390 млн.), далі йдуть Dimensions і AMiner. Web of Science включає понад 92 млн. публікацій, Scopus включає близько 80 млн.

Scopus містить близько 40 тис. видань і пропонує:

- пошук за документами, авторами, організаціями;
- розширений пошук з використанням операторів та візуальних елементів;
- фільтрацію за роками, типами доступу, предметами.

Web of Science підтримує пошук у вибраних базах та надає можливість комбінувати пошукові поля з булевими операторами (AND, OR, NOT) та операторами близькості (NEAR, SAME). Результати можна деталізувати за тематикою, авторами та періодами.

Спільні риси платформ: доступ до широкого спектру наукових публікацій (статті, конференції, книги), складні пошукові інструменти, орієнтація на академічну спільноту. Відмінні риси платформ такі:

- Scopus, Web of Science мають деталізовані бібліометричні аналізи, комерційний доступ ;
- IEEE Xplore спеціалізується на електроніці та IT;
- Google Scholar володіє вільним доступом, забезпечує різноманітність дисциплін;
- CORE, CiteSeerX, Internet Archive Scholar є безкоштовними, з великим обсягом відкритих ресурсів;
- Semantic Scholar використовує алгоритми штучного інтелекту для аналізу даних;

- PubMed Central, Europe PMC індексують праці з медичних і біологічних дисциплін;
- DBLP спеціалізується у галузі комп'ютерних наук.

Internet Archive Scholar охоплює понад 35 млн статей, включаючи історичні та новітні матеріали конференцій. Google Scholar забезпечує пошук у широкому академічному полі. IEEE Xplore спеціалізується на наукових текстах у галузях інформатики, електротехніки та суміжних напрямках, надаючи гнучкі інструменти фільтрації пошуку для зручності користувачів. Платформа Scopus дозволяє виконувати простий та розширений пошук за ключовими словами, назвами документів, авторами, включаючи пошук за ідентифікатором ORCID. У розширеному пошуку доступні булеві оператори (AND, OR, NOT) та оператори близькості (W/n). Також результати можна деталізувати за роками, типами документів, предметними галузями та типами доступу. Web of Science пропонує базовий та розширений пошук з можливістю поєднання пошукових полів за темою, автором, назвою із застосуванням булевих операторів та операторів близькості (наприклад, NEAR/X). Пошук можна деталізувати за датами, авторами, джерелами публікацій та тематичними категоріями.

IEEE Xplore надає можливість пошуку документів за назвами статей, ключовими словами, іменами авторів, а також ідентифікаторами DOI чи ISBN. Користувачі можуть звузити результати за типом документа, роком публікації та тематичною спрямованістю.

Google Scholar забезпечує простий та розширений пошук по ключових словах, авторах, назвах журналів та періоду публікації, однак має обмежені можливості деталізації результатів і не підтримує оператори близькості.

Semantic Scholar застосовує технології штучного інтелекту для класифікації та аналізу результатів пошуку, пропонуючи розширений пошук за ключовими словами, авторами, цитуваннями, пов’язаними темами, а також можливість аналізувати тенденції у дослідженнях і зв’язки між статтями та авторами.

CORE, CiteSeerX та Internet Archive Scholar забезпечують базовий пошук за ключовими словами, назвами та авторами, переважно пропонуючи відкритий доступ до повних текстів і метаданих, однак мають обмежений функціонал розширеного пошуку.

Платформи Europe PMC і PubMed Central спеціалізуються на пошуку медичної та біологічної літератури, використовуючи розширений пошук з медичними термінами (MeSH), іменами авторів, типами досліджень, а також пропонують деталізацію результатів за типами статей, датою та джерелом публікації.

База даних DBLP орієнтована на пошук публікацій у сфері інформатики, пропонуючи пошук за назвами статей, авторами та конференціями з простим та зручним інтерфейсом.

Постановка завдання

Проблема полягає у відсутності інструменту, що дозволяє централізовано збирати, оновлювати та ефективно здійснювати тематичний пошук по великій кількості джерел, а також представляти результати у форматах, зручних для наукової роботи (наприклад, BibTeX). У зв’язку з цим виникає необхідність розробки універсальної програмної системи, яка б поєднувала переваги сучасних моделей пошуку інформації та відповідала вимогам користувачів у сфері наукових досліджень.

Метою роботи є: дослідження і створення моделей та програмного засобу пошуку науково-технічної інформації.

Виклад основного матеріалу

Інформаційний пошук (Information retrieval або IR) охоплює процеси представлення, збереження, організації та доступу до інформаційних ресурсів. У таких системах пошук ґрунтується на ключових словах природномовного запиту користувача. Проблема полягає в тому, що ці запити часто неструктуровані й семантично неоднозначні, на відміну від структурованих даних у базах даних.

Системи баз даних ефективні для точного пошуку даних, але не вирішують завдання семантичного пошуку за змістом. IR-система має вміти аналізувати зміст документів і визначати ступінь їх релевантності запиту, що передбачає витягнення лексичних та семантичних характеристик тексту й використання їх для ранжування.

Центральним поняттям IR є релевантність — ступінь відповідності документа запиту. Мета IR-системи полягає у знаходженні всіх релевантних документів із мінімальною кількістю нерелевантних. Для цього традиційно використовуються *індексні терміни* — значущі слова, які дозволяють індексувати зміст.

Передбачення релевантності документів вирішують *алгоритми ранжування*, які формують впорядкований список результатів, де верхні позиції займають найбільш релевантні документи.

Класичні моделі IR такі:

1. булева модель — представляє документи й запити як множини індексних термінів (теоретико-множинний підхід);
2. векторна модель — трактує документи як вектори в *n*-вимірному просторі (алгебраїчний підхід);
3. імовірнісна модель — базується на теорії ймовірності для оцінки відповідності.

Існують також альтернативи:

- нечіткі булеві моделі;
- латентно-семантичне індексування (LSI);
- нейромережеві моделі (в т.ч. імовірнісні).

Для формалізації поняття IR-системи введемо такі позначення:

- k_i — *i*-те ключове слово документу;
- d_j — j-й документ;
- w_{ij} — вага k_i у d_j, що відображає його значущість.

Множину термінів представимо так:

$$K = \{k_1, k_2, \dots, k_t\}.$$

Документ d_j описується вектором:

$$\overrightarrow{d_j} = \{w_{1j}, w_{2j}, \dots, w_{tj}\}.$$

Функцію, що повертає вагу терміна k_i в документі d_j опишемо так:

$$g_i(\overrightarrow{d_j}) = w_{ij}.$$

Можлива некорельованість пар:

$$(k_{i+1}, d_j) \neq (k_i, d_j).$$

Формальне визначення IR-системи таке:

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

де:

- $\boldsymbol{D}$ — множина документів,
- $\boldsymbol{Q}$ — множина запитів,
- $\boldsymbol{F}$ — структуроване представлення документів та зв'язків,
- $R(q_i, d_j)$ — функція ранжування для оцінки відповідності документа d_j запиту q_i.

Після обчислення $R(q_i, d_j)$ система формує впорядкований список документів за релевантністю.

Булева модель пошуку — це базова модель інформаційного пошуку, що спирається на теорію множин і булеву алгебру. В булевій моделі пошуку використовуються логічні операції заперечення, кон'юнкції, диз'юнкції. Для множини $K = \{k_1, k_2, \dots, k_t\}$ ключових слів ваги $w_{ij} \in \{0,1\}$, а запити формулюються у вигляді булевих виразів, наприклад:

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

Такий запит можна перетворити у диз'юнктивну нормальну форму:

$$q_{dnf} = (1,0,0) \vee (1,0,1).$$

Приймемо що $\vec{q}_{cc}$ є кон'юнктивною нормальною формою для $\vec{q}_{dnf}$. Оцінка подібності між документом *dj* і запитом *q*:

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

Якщо $sim = 1$, документ є релевантним.

Булева модель формальна й точна, тому її широко використовують у комерційних системах. Проте вона має такі недоліки:

1. результат є лише бінарним (релевантний/нерелевантний);
2. перетворення інформаційної потреби на логічний вираз може бути складним.

Попри це, булева модель залишається однією з основних моделей пошуку.

Векторна модель простору використовує зважування за частотою терміна та оберненою частотою документа (TF-IDF). У цій моделі документи і запити подаються у вигляді векторів високої розмірності, елементами яких є ваги термінів, а релевантність визначається за косинусною подібністю між цими векторами [1]. TF-IDF надає вищі ваги тим термінам, які часто зустрічаються в документі, але є рідкісними в усьому корпусі, що підсилює терміни, які найкраще характеризують зміст документа. Варіації TF-IDF широко використовуються в пошукових системах як основний компонент ранжування [2].

Векторна модель може використати і двійкові ваги. Проте, головна перевага — підтримка часткової релевантності. Вона призначає небінарні ваги ключовим словам у запитах і документах, дозволяючи оцінювати ступінь подібності між ними. Це забезпечує ранжування результатів — більш релевантні документи розміщуються вище у списку.

Коефіцієнти ваг розраховуються різними методами. Наприклад у роботі [3] пропонують використовувати частоту терміну t_f (як внутрішньодокументну характеристику) і зворотну частоту документів IDF (для міждокументної відмінності). Частота t_f відображає, наскільки термін описує документ, а IDF — наскільки термін рідкісний у всій колекції.

Формула для розрахунку ваги слова k_i в документі d_j така:

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

де f_{ij} нормалізована частота ключового слова k_i в документі d_j.

Основні переваги векторної моделі: зважування ключових слів покращує продуктивність пошуку, сортування знайдених документів за ступенем їх схожості із запитом.

Побудовані на основі TF-IDF, ймовірнісні моделі, такі як Okapi BM25, стали фактичним стандартом функцій ранжування в сучасних пошукових системах. BM25 — це алгоритм пошуку, що працює за принципом "мішка слів" і оцінює документи на основі частоти терміна, довжини документа та оберненої частоти документа [2]. Він часто перевершує звичайний TF-IDF завдяки врахуванню ефекту спадної віддачі від частоти терміна та нормалізації за довжиною документа. Основні наукові пошукові системи (наприклад, Semantic Scholar та інші) використовують інверсний індекс із зважуванням BM25 на першому етапі пошуку завдяки його високій швидкодії та ефективності [4].

Машинне навчання застосовується для побудови оптимальних функцій ранжування на основі даних. Такий підхід відомий як навчання ранжуванню (learning to rank, LTR). Замість ручного створення формул релевантності, алгоритми LTR навчаються на кліках користувачів або оцінках релевантності, щоб поєднати багато ознак у модель, яка прогнозує релевантність документа [5]. У контексті академічного пошуку LTR може враховувати такі ознаки, як текстова подібність, кількість цитувань, давність публікації тощо, щоб краще впорядковувати результати.

Кластеризаційні методи застосовуються для організації результатів пошуку або попередньої обробки простору документів з метою покращення пошуку. Групуючи подібні документи, кластеризація допомагає користувачам орієнтуватися в результатах запиту або забезпечує диверсифікацію (наприклад, показ по одному результату з кожного кластера для охоплення різних аспектів). Традиційна кластеризація (наприклад, k-середніх або ієрархічна кластеризація за TF-IDF-векторами документів) може групувати наукові роботи за тематичними напрямками. Сучасніші підходи реалізують семантичну кластеризацію, яка базується на змісті документів, а не лише на ключових словах [6]. Наприклад, один із методів будує граф тем та кластеризує результати пошуку шляхом віднесення кожного документа до вузла теми у Вікіпедії, створюючи узгоджені тематичні кластери [7]. В академічних пошукових порталах кластеризація може виступати як "науковий помічник": після початкового пошуку статті можуть бути згруповані за темами, такими як експериментальні дослідження, оглядові роботи, окремі підзадачі тощо.

Тематичне моделювання — це метод без учителя, що дозволяє виявити приховані теми у колекції документів. Популярна модель — Latent Dirichlet Allocation (LDA), яка представляє документи як суміші тем, а кожну тему — як ймовірнісний розподіл слів. У платформах академічного пошуку LDA та пов’язані з нею моделі використовуються для покращення пошуку й навігації. Наприклад, у [8] було застосовано тематичне моделювання на основі LDA поверх індексу ElasticSearch для класифікації та реорганізації результатів пошуку за науковими темами. Завдяки виділенню прихованих тем із заголовків і анотацій статей, система могла групувати результати пошуку за змістовими категоріями. Тематичні моделі також можуть підтримувати системи рекомендацій або розширення запитів — наприклад, пропонуючи додаткові ключові слова котрі пов’язані з темою запиту. Перевага LDA полягає в тому, що вона надає високорівневий семантичний огляд і здатна знаходити пов’язані документи навіть без спільних ключових слів.

Сучасні академічні пошукові системи часто поєднують декілька стратегій — лексичних, семантичних, на основі цитувань тощо [9-12]. У [4] інверсний індекс із ранжуванням BM25 поєднано з аналізом мережі цитувань для повторного ранжування та покращення результатів. Система спочатку відбирає документи кандидати на основі збігу ключових слів, а потім враховує структуру графа цитувань — наприклад, підвищуючи оцінку документам, які активно цитуються або тісно пов’язані з початковими результатами. Така гібридизація дозволяє враховувати не лише текстовий вміст документів, а й науковий контекст (які документи цитуються та ким), що також впливає на оцінку релевантності.

У Західноукраїнському національному університеті на протязі двадцяти років у співпраці з Тернопільським національним медичним університетом імені Івана Пулюя працює наукова група з аналізу біомедичних зображень. Співкерівниками цієї наукової групи є д.т.н., професор Березький О.М., і д.м.н., професор Сельський П.Р. Деякі публікації цієї групи приведені у посиланнях на використану літературу [13-15].

Розроблена програмна система реалізує пошук за ключовими словами з можливістю фільтрації за окремими полями (наприклад, заголовок або зміст), а також із урахуванням декількох критеріїв (дати, назви журналів).

Основним інструментом розробки серверної частини виступало інтегроване середовище PhpStorm 2023 від компанії JetBrains, яке забезпечує повний цикл розробки програмного коду на PHP та супутніх веб-технологіях. Фронтенд-розробка здійснювалась із залученням таких інструментів, як Node.js і npm — для керування залежностями, а також Webpack і Babel, які забезпечують трансформацію сучасного JavaScript-коду (зокрема, з використанням ES6+) до формату, сумісного з різними середовищами виконання.

Для організації залежностей на стороні PHP застосовувався менеджер Composer, який дозволяє легко інтегрувати зовнішні бібліотеки у проєкт. Контроль версій програмного коду здійснювався за допомогою системи Git з використанням віддаленого репозиторію GitHub як централізованого середовища зберігання та керування історією змін.

Розроблена програмна система пошуку передбачає використання класичної трирівневої моделі, яка включає фронтенд, бекенд, базу даних та модулі для зовнішньої інтеграції через API, а також механізми забезпечення безпеки. Клієнтська частина (фронтенд) реалізована за допомогою стандартного стеку веб-технологій. Розмітка структури сторінок виконується мовою HTML, яка забезпечує логічну організацію контенту. Візуальне оформлення інтерфейсу користувача здійснюється за допомогою каскадних таблиць стилів CSS, що відповідають за зовнішній вигляд елементів. Для додавання інтерактивності, обробки подій та реалізації динамічних компонентів застосовується мова JavaScript. Серверна частина (бекенд) побудована на основі мови PHP версії 8.0. Веб-сервер Apache версії 2.4 використовується як посередник для обробки HTTP-запитів і відповідає за передачу даних між клієнтом і сервером. У якості основи для розробки серверної логіки обрано фреймворк Laravel версії 9.x, що забезпечує підтримку архітектурної моделі Model-View-Controller, дозволяє організувати модульну структуру коду і сприяє масштабованості системи. Система управління базами даних MySQL версії 8.0 використовується для зберігання, запиту та управління структурованою інформацією. Запити до бази реалізуються засобами ORM (об'єктно-реляційного відображення), які підтримує Laravel.

Інтеграція з зовнішніми ресурсами забезпечується через RESTful API. Такий інтерфейс дозволяє встановлювати з'єднання з сервісами сторонніх постачальників, зокрема таких, як Elsevier або IEEE, отримуючи структуровану інформацію у форматі JSON.

Окрему увагу приділено механізмам безпеки. У систему впроваджено методи аутентифікації та авторизації, що дозволяють здійснювати контроль доступу до функціоналу залежно від ролі користувача. Таким чином, структура програмного забезпечення забезпечує надійну роботу в середовищі сучасного веб-додатку, підтримуючи гнучкість, масштабованість і взаємодію з зовнішніми джерелами даних.

Програмний засіб складається з трьох компонент:

- збір інформації;
- актуалізація даних;
- пошук статей.

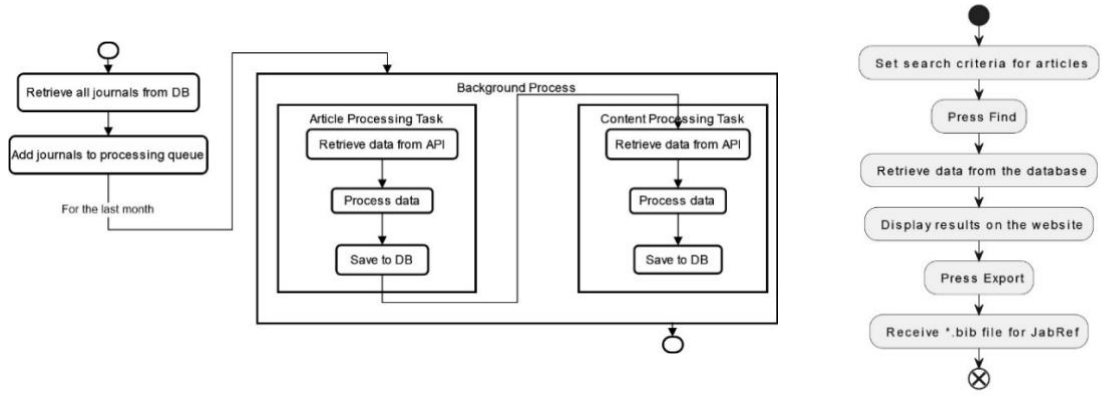
Компонент для збору інформації реалізує процес завантаження за 5 років . Процес ініціюється на клієнтській стороні шляхом введення назв наукових журналів. Після цього формується запит до серверної частини, яка, у свою чергу, додає відповідні журнали до черги обробки. Далі відбувається послідовне виконання трьох основних задач у фоні:

1. Етап обробки журналів система звертається до зовнішнього API з метою отримання метаданих про журнали. Після отримання інформації дані проходять попередню обробку, нормалізацію або фільтрацію та зберігаються до бази даних.
2. Після збереження інформації про журнали, система формує окреме завдання для збору статей, опублікованих у цих журналах за останні п'ять років. Аналогічно до попереднього етапу, дані отримуються через API, обробляються і додаються до сховища.
3. Заключний етап полягає у зборі повного текстового контенту або розширених метаданих кожної статті. Цей процес також передбачає запити до зовнішнього API, обробку отриманої інформації та її збереження до бази даних.

Компонент для актуалізації даних відповідає за періодичне оновлення даних про статті з наукових журналів за останній місяць (рис. 1,а). Процес починається з вибірки усіх наявних журналів із бази даних, після чого кожен з них додається до черги обробки. У фоні запускається послідовність завдань, зокрема:

- обробка статей: отримання метаданих через API, подальша обробка отриманої інформації та її збереження в базу даних;
- обробка контенту статей: аналогічний цикл запиту, обробки та збереження повного тексту або розширених описів публікацій.

Компонент пошуку статей призначений для пошуку та експорту наукових статей у форматі, сумісному з бібліографічним менеджером JabRef (рис. 1,б). Користувач задає критерії пошуку, після чого ініціює процес за допомогою кнопки Find. Система виконує запит до бази даних, отримує відповідні записи та відображає результати на вебінтерфейсі. За бажанням, користувач може експортувати знайдені публікації, натиснувши кнопку Export, унаслідок чого формується файл у форматі .bib, придатний для подальшого використання в JabRef або інших інструментах для управління бібліографією.



а) **б)**

Рис. 1. UML діаграми програмної системи: а)– компонент для актуалізації даних; б) – компонент пошуку статей

На зображеній на рис. 2 ER-схемі представлено структуру бази даних системи збору та управління науковими публікаціями. Схема охоплює основні функціональні модулі — зберігання метаданих, повного контенту статей, користувачів, фонових задач і механізмів автентифікації. Нижче наведено доповнений опис таблиць.

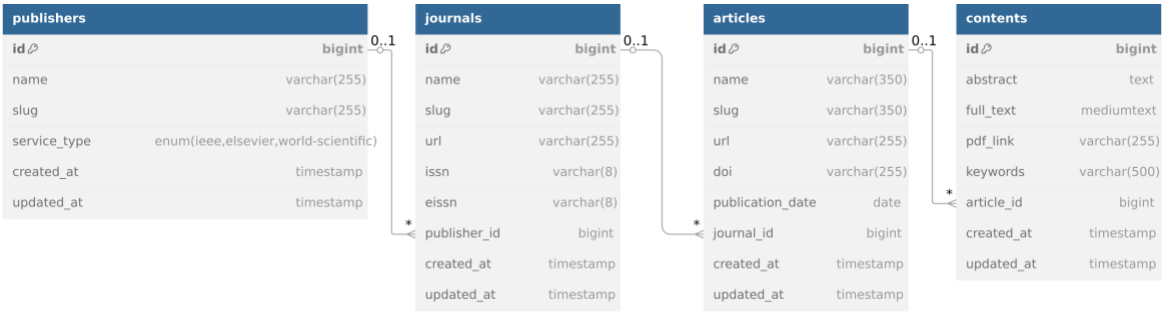


Рис. 2. Фрагмент структури бази даних

Таблиця publishers містить інформацію про наукові видавництва. Поле service_type обмежене значеннями переліченого типу (наприклад elsevier, ieee), що дозволяє адаптувати систему до особливостей API різних провайдерів.

Таблиця journals Зберігає інформацію про журнали, прив'язані до відповідного видавництва через зовнішній ключ publisher_id. Містить назву, унікальний ідентифікатор (slug), URL, а також стандартизовані міжнародні номери — ISSN та EISSN.

Таблиця articles призначена для збереження статей, які було знайдено у відкритому доступі. Кожна стаття пов'язана з конкретним журналом (journal_id) і має унікальні атрибути: назву, DOI, дату публікації.

У таблиці articles фіксуються статті, виявлені у відкритому доступі для кожного журналу. Зберігається назва статті, DOI (унікальний цифровий ідентифікатор публікації), вебпосилання на сторінку публікації та дата її оприлюднення.

Для зберігання змістової частини публікацій використовується таблиця contents. Вона включає такі поля, як abstract (резюме або анотація статті), full_content — текстовий вміст, отриманий через API від відповідного видавництва, pdf_link — посилання на повну версію статті у форматі PDF, а також keywords, які відповідають ключовим словам, визначеним авторами.

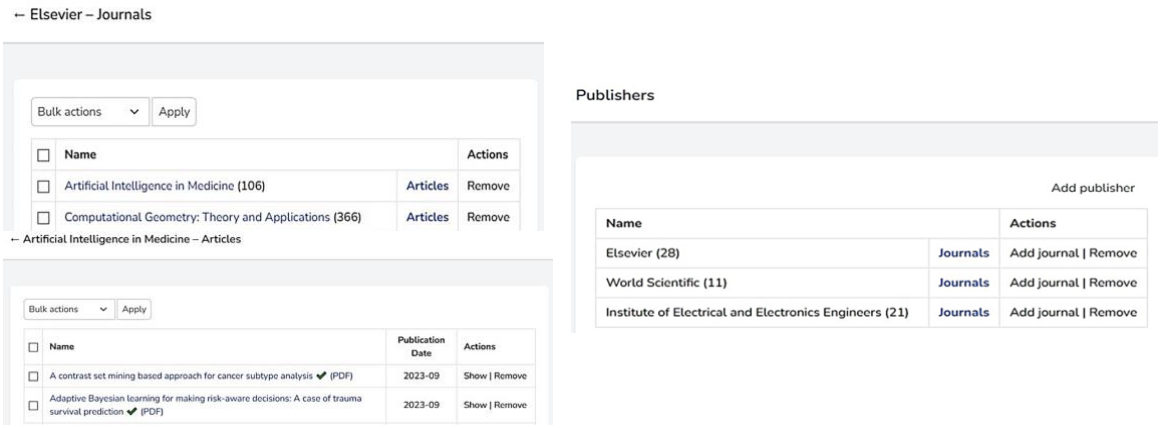
Таблиця contents містить повний контент кожної статті, включаючи анотацію (abstract), текстову частину (full_text), посилання на PDF (pdf_link) і ключові слова. Ця таблиця зв'язана із таблицею diploma_articles через article_id.

Таблиця users відповідає за зберігання даних зареєстрованих користувачів системи: ім'я, email, хеш пароля. Поле email_verified_at дає змогу реалізувати верифікацію електронної пошти.

Підсистема автентифікації та авторизації складається із таблиць personal_access_tokens та password_resets. Перша служить для реалізації механізму доступу через API з використанням персональних токенів. Вона забезпечує підтримку типів токенів, можливостей (abilities) і терміну дії (expires_at). Друга зберігає тимчасові токени для відновлення паролів користувачами за email.

Для початку взаємодії з функціоналом веб-ресурсу користувач має пройти процедуру реєстрації, після чого здійснити авторизацію для доступу до закритої частини сайту. Після входу відкривається головна сторінка внутрішнього розділу сайту, що містить навігаційне меню. Основні функції зосереджені у пунктах Publishers та Search. При переході до сторінки Publishers користувач

отримує перелік видавництв, які наразі інтегровані з системою. При цьому виводиться назва видавництва та кількість пов'язаних журналів. Функціональність сторінки дозволяє як видаляти наявні видавництва, так і додавати нові журнали за допомогою кнопки «Add journal». Після натискання на «Add journal» ініціюється запит до сервера, де в асинхронному режимі поетапно виконується обробка: спершу отримуються метадані журналу, далі – перелік публікацій, далі для статті витягується резюме, посилання на PDF, повний текст (якщо є) тощо.



а) **б)**

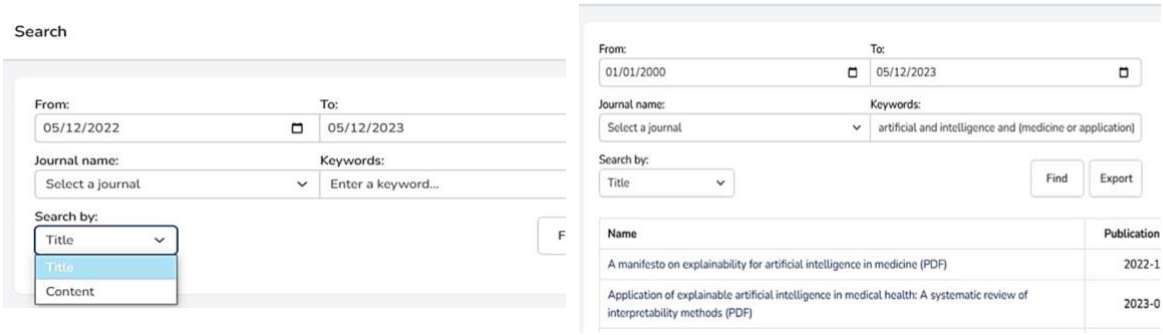
Рис. 3. Інтерфейс завантаження журналів в базу даних: а) – журнали; б) – видавництва

На сторінці видавництва можна перейти до переліку журналів і переглянути кількість знайдених статей. Для перегляду статей, пов'язаних із журналом, натискається посилання «Articles» (рис. 3,а). Відображаються ключові параметри: назва статті (з посиланням), PDF-лінк та дата публікації.

Функціонал пошуку реалізовано на основі ключових слів. Фільтрація можлива за такими параметрами: початкова дата («From»), – кінцева дата («To»), назва журналу («Journal name»), ключові слова («Keywords»), тип поля пошуку («Search by» заголовок або вміст статті). Результати пошуку можна експортувати у форматі BibTeX (.bib), що дозволяє надалі імпортувати їх у системи бібліографічного управління JabRef. На рис. 4 показано основне вікно пошуку документів котре забезпечує три сценарії пошуку:

- пошук за ключовими словами у назві;
- пошук за ключовими словами у змісті;
- комбінація фільтрації за назвою журналу та ключовими словами в заголовку/тексті.

На рис. 4, а показано результат булевого пошуку по назві статті.



а) **б)**

Рис. 4. Пошук статті в базі даних: а) – задання атрибутів пошуку; б) – результати пошуку

Для реалізації системи пошуку наукових публікацій було використано публічні програмні інтерфейси видавництв IEEE та Elsevier, що надають доступ до метаданих журналів і статей. Отримані дані, такі як назви журналів, ISSN-коди, назви статей, анотації та повні тексти, використовуються як вхідні параметри для побудови пошукових запитів.

Висновки

У результаті реалізації програмного засобу пошуку науково-технічної інформації було створено ефективну веб-систему, здатну здійснювати автоматизований збір, обробку, зберігання та тематичний пошук наукових публікацій з відкритих джерел. Розроблений програмний продукт забезпечує інтеграцію з API провідних наукових видавництв (Elsevier, IEEE), дозволяє формувати структуровану базу даних наукових статей та підтримує зручний механізм фільтрації результатів за ключовими словами, журналами, датами публікацій тощо. Програмна система реалізована на основі

сучасного стеку веб-технологій, зокрема мови PHP у зв'язці з фреймворком Laravel, архітектурою MVC та базою даних MySQL. Для зручності користувача передбачено графічний інтерфейс, механізм авторизації, а також функціонал експорту результатів у форматі BibTeX. Запропоноване рішення дозволяє значно спростити доступ до релевантної інформації, що є важливим чинником при підготовці літературних оглядів, аналітичних звітів та написанні наукових публікацій.

Література

1. Patel M. TinySearch - Semantics based Search Engine using Bert Embeddings // CoRR. — 2019. — Vol. abs/1908.02451.

2. Rivas A. R. Study of Query Expansion Techniques and Their Application in the Biomedical Information Retrieval / A. R. Rivas, E. L. Iglesias, L. Borrajo // The Scientific World Journal. — 2014. — Vol. 2014. — P. 1–10.

3. Grossman D. A. Information Retrieval: Algorithms and Heuristics / D. A. Grossman, O. Frieder. — Springer US, 1998. — 262 p.

4. Khalid S. An Effective Scholarly Search by Combining Inverted Indices and Structured Search With Citation Networks Analysis / S. Khalid, S. Wu, A. Wahid, et al. // IEEE Access. — 2021. — Vol. 9. — P. 120210–120226.

5. Wang J. Utilizing BERT for Information Retrieval: Survey, Applications, Resources, and Challenges / J. Wang, J. X. Huang, X. Tu, et al. // ACM Comput. Surv. — 2024. — Vol. 56, Num 7. — P. 185:1-185:33.

6. Soliman S. S. Semantic Clustering of Search Engine Results / S. S. Soliman, M. F. El-Sayed, Y. F. Hassan // The Scientific World Journal. — 2015. — Vol. 2015, Num 1. — P.1-9.

7. Scaiella U. Topical clustering of search results / U. Scaiella, P. Ferragina, A. Marino, M. Ciaramita. — New York, NY, USA,2012. — P. 223-232.

8. Kim M. A Suggestion on the LDA-Based Topic Modeling Technique Based on ElasticSearch for Indexing Academic Research Results / M. Kim, D. Kim // Applied Sciences. — 2022. — Vol. 12, Num 6. — P. 3118.

9. Paiva S. GSSP–A Generic Semantic Search Platform / S. Paiva, M. Ramos-Cabrer, A. Gil-Solla // Procedia Technology. — 2012. — Vol. 5. — P. 388–396.

10. Huang C.-Q. EIIS: An Educational Information Intelligent Search Engine Supported by Semantic Services /C.-Q. Huang, R.-L. Duan, Y. Tang, et al. // International Journal of Distance Education Technologies. — 2011. — Vol. 9, Num 1. — P. 21–43.

11. A Semantic Search Engine for Historical Handwritten Document Images. / V. M. Ngo, G. Munnelly, F. Orlandi et al. // Linking Theory and Practice of Digital Libraries - 25th International Conference on Theory and Practice of Digital Libraries, TPDL 2021, Virtual Event, September 13-17, 2021. — Springer, 2021. — P. 60–65.

12. Properties and Structure of Fast Text Search Engine in Context of Semantic Image Analysis. / J. Rygal, P. Najgebauer, T. Nowak et al. // Artificial Intelligence and Soft Computing - 11th International Conference, ICAISC 2012, Zakopane, Poland, April 29-May 3, 2012. — Springer, 2012. — P. 592–599.

13. Berezsky O. N. Topological Methods and Algorithms of Transform of the Contours and Regions of Flat Images // Journal of Automation and Information Sciences. — 2010. — Vol. 42, № 10. — P. 49–59.

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

15. Berezsky O. Access Distribution in Automated Microscopy System / O. Berezsky, L. Dubchak, Oleh Pitsun // Proceedings of the 14 th International Conference «The Experience of Designing and Application of CAD Systems in Microelectronics» CADSM 2017, 21-25 February 2017, Lviv, Ukraine. – Lviv, 2017. – P. 241-243.

References

1. Patel M. TinySearch - Semantics based Search Engine using Bert Embeddings // CoRR. — 2019. — Vol. abs/1908.02451.

2. Rivas A. R. Study of Query Expansion Techniques and Their Application in the Biomedical Information Retrieval / A. R. Rivas, E. L. Iglesias, L. Borrajo // The Scientific World Journal. — 2014. — Vol. 2014. — P. 1–10.

3. Grossman D. A. Information Retrieval: Algorithms and Heuristics / D. A. Grossman, O. Frieder. — Springer US, 1998. — 262 p.

4. Khalid S. An Effective Scholarly Search by Combining Inverted Indices and Structured Search With Citation Networks Analysis / S. Khalid, S. Wu, A. Wahid, et al. // IEEE Access. — 2021. — Vol. 9. — P. 120210–120226.

5. Wang J. Utilizing BERT for Information Retrieval: Survey, Applications, Resources, and Challenges / J. Wang, J. X. Huang, X. Tu, et al. // ACM Comput. Surv. — 2024. — Vol. 56, Num 7. — P. 185:1-185:33.

6. Soliman S. S. Semantic Clustering of Search Engine Results / S. S. Soliman, M. F. El-Sayed, Y. F. Hassan // The Scientific World Journal. — 2015. — Vol. 2015, Num 1. — P.1-9.

7. Scaiella U. Topical clustering of search results / U. Scaiella, P. Ferragina, A. Marino, M. Ciaramita. — New York, NY, USA,2012. - P. 223–232.

8. Kim M. A Suggestion on the LDA-Based Topic Modeling Technique Based on ElasticSearch for Indexing Academic

Research Results / M. Kim, D. Kim // Applied Sciences. — 2022. — Vol. 12, Num 6. — P. 3118.

9. Paiva S. GSSP–A Generic Semantic Search Platform / S. Paiva, M. Ramos-Cabrer, A. Gil-Solla // Procedia Technology. — 2012. — Vol. 5. — P. 388–396.

10. Huang C.-Q. EIIS: An Educational Information Intelligent Search Engine Supported by Semantic Services / C.-Q. Huang, R.-L. Duan, Y. Tang, et al. // International Journal of Distance Education Technologies. — 2011. — Vol. 9, Num 1. — P. 21–43.

11. A Semantic Search Engine for Historical Handwritten Document Images. / V. M. Ngo, G. Munnelly, F. Orlandi et al. // Linking Theory and Practice of Digital Libraries - 25th International Conference on Theory and Practice of Digital Libraries, TPDL 2021, Virtual Event, September 13-17, 2021. — Springer, 2021. — P. 60–65.

12. Properties and Structure of Fast Text Search Engine in Context of Semantic Image Analysis. / J. Rygal, P. Najgebauer, T. Nowak et al. // Artificial Intelligence and Soft Computing - 11th International Conference, ICAISC 2012, Zakopane, Poland, April 29-May 3, 2012. — Springer, 2012. — P. 592–599.

13. Berezsky O. N. Topological Methods and Algorithms of Transform of the Contours and Regions of Flat Images // Journal of Automation and Information Sciences. — 2010. — Vol. 42, Num 10. — P. 49–59.

14. Berezsky O. M. Intelektualna systema dlia diahnostuvannia riznykh form raku molochnoi zalozy na osnovi analizu histolohichnykh i tsytolohichnykh zobrazhen / O. M. Berezsky, G. M. Melnyk, Yu. M. Batko, T. V. Datsko // Naukovyi visnyk NLTU Ukrainy: zb. nauk.-tekhn. prats. — 2013. — Vol. 23.13. — P. 357–367.

15. Berezsky O. Access Distribution in Automated Microscopy System / O. Berezsky, L. Dubchak, Oleh Pitsun // Proceedings of the 14 th International Conference «The Experience of Designing and Application of CAD Systems in Microelectronics» CADSM 2017, 21-25 February 2017, Lviv, Ukraine. – Lviv, 2017. – P. 241-243.