

СИМАК АНДРІЙ
Західноукраїнський національний університет
<https://orcid.org/0009-0009-9873-2878>
e-mail: simak.chik.chirik@gmail.com
ЮШКО АНДРІЙ
Західноукраїнський національний університет
<https://orcid.org/0009-0003-6431-3479>
e-mail: a.yushko@wunu.edu.ua

ІНТЕЛЕКТУАЛЬНИЙ МОДУЛЬ ЗБОРУ ВІДКРИТИХ ДАНИХ В СИСТЕМІ РЕЙТИНГУВАННЯ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНОГО ПЕРСОНАЛУ

У науковій роботі описано архітектуру модуля збору та обробки відкритих даних про наукові роботи працівників закладу вищої освіти, інтегрованого в рейтингову систему університету. Модуль працює в рамках системи рейтингування науково-педагогічних працівників, яка реалізована на фреймворку Laravel. Він включає графічний інтерфейс для користувачів (науковців, завідувачів кафедр, рецензентів, адміністраторів) та серверну частину на PHP з JavaScript. Для зберігання даних використовується MySQL сервер.

Модуль автоматизує збір інформації про наукову діяльність, зокрема статті, конференційні публікації та інші роботи з джерел, як-от Web of Science, Google Scholar і Scopus. Розпізнавання та класифікація робіт відповідно до критеріїв рейтингової системи здійснюється за допомогою штучного інтелекту, що використовує трансформерну архітектуру із механізмом самоуваги (self-attention). Ця технологія дозволяє моделі аналізувати текстові дані, розуміти контекст і семантичні зв'язки між словами, та цей процес інтегрований через API OpenAI.

Для збору даних використовується сервер на Node.js з Express.js і хмарне сховище MongoDB через GraphQL API. Сервер інтегрується з джерелами, як-от DSpace та Електронна бібліотека НАПН України. Багаторівнева архітектура підвищує надійність і масштабованість системи. Взаємодія з модулем відбувається через JWT Tokens, що забезпечує захист даних і доступ до них тільки авторизованим користувачам.

Ключові слова: веб-система, система оцінювання діяльності, науково-педагогічні працівники, заклад вищої освіти, рейтинг, автоматизація, інтеграція, штучний інтелект, ChatGPT.

SIMAK Andriy, YUSHKO Andriy
West Ukrainian National University

INTELLIGENT OPEN DATA COLLECTION MODULE IN THE ACADEMIC STAFF PERFORMANCE APPRAISAL SYSTEM

The scientific work implements the structure of a module designed for the collection and processing of open data related to scientific works of faculty members at a university, which is incorporated into the performance evaluation system of academic staff. This module serves as part of the current rating system, which is developed using the Laravel framework. The system includes a graphical presentation for users (researchers, heads of departments, reviewers and administrators) and a server-side coded in PHP using JavaScript. A MySQL server that is deployed is utilized for storing the data.

The suggested module automates the gathering of information related to the scientific activity of employees, such as articles, conference papers and other publications, from public sources, including connected external web platforms and publication databases, such as Web of Science, Google Scholar and Scopus. The recognition and categorization of scientific works based on the rating system's criteria are performed through artificial intelligence, which employs a transformer architecture. This architecture processes and analyzes textual data using a self-attention mechanism, enabling the model to comprehend context, relationships between words, and perform semantic analysis and the process is integrated through the OpenAI API.

For data collection and processing, a Node.js server with Express.js and MongoDB cloud storage via GraphQL API are used. The server integrates with external sources, such as Electronic Library of the National Academy of Educational Sciences of Ukraine and the DSpace. The multi-tiered architecture increases the reliability and scalability of the system. Interaction with the module is provided via JWT Tokens, which guarantees data security and access only to authorized users.

Key words: web system, academic staff performance appraisal system, scientific and pedagogical workers, higher education institution, rating, automation, integration, artificial intelligence, ChatGPT.

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

У сучасному вищому освітньому середовищі, яке постійно змінюється і стає все більш конкурентним, систематичний збір та аналіз наукової і науково-педагогічної діяльності стають критично важливими для забезпечення успіху та розвитку університету [1]. Це має важливе значення не лише для самого закладу, а й для його науково-педагогічного складу. Ефективне управління результатами праці викладачів та науковців сприяє покращенню якості освітніх послуг, підвищенню престижу установи на національному та міжнародному рівнях, залученню додаткового фінансування для проведення наукових досліджень, а також зміцненню позицій університету в рейтингах конкурентоспроможності [2].

Водночас, для самих науково-педагогічних працівників такі системи дозволяють покращити умови для професійного розвитку, стимулюючи до кар'єрного зростання та наукової активності. Важливими

заходами для досягнення цих цілей є впровадження технологій, спрямованих на автоматизацію процесів збору та оцінки наукових робіт, підвищення мотивації персоналу через систему оцінювання досягнень, організація навчальних заходів для підвищення кваліфікації та активне залучення викладачів до участі в науковій роботі. Застосування сучасних технологій в навчальному процесі сприяє покращенню досвіду студентів, що, у свою чергу, має позитивний вплив на імідж та ефективність університету.

Використання різноманітних систем рейтингування, підготовка звітної документації та інші адміністративні завдання є невід'ємною частиною наукової діяльності. Автоматизація цих процесів звільнить час для більш цінних активностей. На сьогодні існують різні системи та платформи, призначені для автоматизації наукової та науково-педагогічної діяльності. Штучний інтелект і нові технології відкривають шляхи для покращення систем, що аналізують академічну працю. Таким чином, створення інтелектуальної системи для автоматизації звітів наукових працівників є важливим кроком для підвищення ефективності роботи університетів.

Основною метою є створення модуля та інтеграція його з існуючою системою рейтингування, який здатний автоматично збирати дані з внутрішніх акаунтів користувачів, а також з науково-метричних ресурсів, зокрема таких, як Web of Science, Google Scholar і Scopus. Окрім того, модуль повинен здатний здійснювати аналіз інформації з різноманітних джерел, таких як вебсайти бібліотечних ресурсів, застосовуючи методи семантичного аналізу та кластеризації даних. Це сприятиме підвищенню ефективності функціонування університету, зменшенню адміністративного навантаження на співробітників та покращенню якості навчального процесу.

В даній роботі процес автоматизації рейтингування реалізовано з допомогою інтеграції рейтингової системи університету з інтелектуальним модулем, для автоматизації збору та заповнення користувацьких даних у рейтинговий звіт для формування оцінки наукової діяльності працівників. Результатом є автоматичне заповнення частини критеріїв звіту та зменшення часу і зусиль, необхідних для виконання цих завдань працівниками університету.

Для створення модуля збору даних та його інтеграції в систему рейтингування діяльності науково-педагогічного персоналу необхідно вирішити такі завдання:

1. Забезпечити модульність та інтеграцію системи для автоматизації збору наукових робіт з внутрішніх та зовнішніх джерел.
2. Спроекувати архітектуру модуля збору відкритих даних, із ефективним управлінням запитами та оптимізацією передачі даних.
3. Реалізувати механізм парсингу даних з веб-ресурсів (Web of Science, Google Scholar, Scopus) з подальшою обробкою даних.
4. Забезпечити взаємодію модуля збору даних із системою штучного інтелекту для аналізу та групування наукових публікацій.
5. Інтегрувати модуль збору даних із рейтинговою системою університету, забезпечивши автоматичне наповнення звітів користувачів, на основі зібраної інформації.
6. Забезпечити масштабованість модуля для можливості подальшого розширення функцій та підтримки нових джерел відкритих наукових даних.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Зважаючи на те, що чинна рейтингова система Західноукраїнського національного університету (ЗУНУ) активно працює, архітектура нової додаткової системи повинна бути розроблена з урахуванням модульного підходу. Це дозволить інтегрувати її в поточний процес формування рейтингів, забезпечуючи при цьому гнучкість і здатність до масштабування. Цей підхід дозволяє системі швидко адаптуватися до нових вимог та підтримувати її постійний розвиток без серйозних перебоїв у роботі основної системи. Для спрощення архітектури рейтингової системи, інтелектуалізований модуль збору даних буде автономною системою, здатною працювати незалежно. Така організація дозволяє модулям і основній системі функціонувати окремо, що позитивно впливає на їхню надійність і ефективність. Оскільки процес збору даних може займати деякий час, відокремлення цього процесу від основної системи допоможе мінімізувати вплив на продуктивність рейтингової системи.

Загальну архітектуру системи рейтингування з новим модулем зобразимо на (рис. 1).

Система рейтингування, яка функціонує в ЗУНУ, розроблена на основі фреймворку Laravel, вибір якого обумовлений його високою гнучкістю та здатністю до масштабування (1.a). Laravel пропонує модульну структуру, що спрощує інтеграцію нових функцій без необхідності суттєвих змін у основному коді. Застосування цього фреймворку також скорочує час розробки завдяки наявності великої кількості вбудованих компонентів і бібліотек. Серверна частина системи реалізована на PHP, який є популярним інструментом для створення веб-додатків завдяки зручності у використанні та широкому колу підтримки з боку розробників. PHP, у поєднанні з JavaScript для клієнтської логіки (1.b), забезпечує інтерактивність і швидкість у роботі з користувацьким інтерфейсом, що є важливим для зручності взаємодії науково-педагогічних працівників, завідувачів кафедр, рецензентів і адміністраторів із системою. Для зберігання даних використовується MySQL

сервер (1.c), що є одним з найпоширеніших рішень для реляційних баз даних завдяки його високій надійності, продуктивності та сумісності з PHP. Використання MySQL також дає можливість оптимізувати роботу з великими обсягами даних, що є важливим для збереження інформації про наукову діяльність.

Сервер для збору інформації про наукову діяльність користувачів побудований на основі Node.js та Express.js (2.a), що є сучасними технологіями для серверної розробки. Node.js забезпечує асинхронну обробку запитів, що значно підвищує продуктивність системи за умови великої кількості користувачів і запитів. Express.js, у свою чергу, є мінімалістичним веб-фреймворком для Node.js, що полегшує розробку серверної логіки та управління маршрутизацією. Для зберігання даних у цьому сервері використовується MongoDB, розміщена у хмарному середовищі (2.b). MongoDB обрана через її здатність зберігати дані в неструктурованому або напівструктурованому вигляді, що забезпечує гнучкість у зберіганні наукових робіт та метаданих, які не завжди можуть мати чітку структуру. Важливим компонентом системи є GraphQL API, який використовується для взаємодії між сервером та базою даних (2.b). GraphQL було обрано замість традиційного REST API через його здатність точно контролювати обсяги даних, що передаються між клієнтом і сервером. Це дозволяє зменшити обсяг непотрібних запитів і оптимізувати використання ресурсів.

Збір, розпізнавання та класифікація наукових робіт. Цей елемент системи використовує штучний інтелект, зокрема API OpenAI [3-4], для автоматичного розпізнавання критеріїв та відповідної їм наукової роботи з подальшим автоматичним включенням її до звіту (3). Також він виконує роль мосту між модулем збору даних та зовнішніми джерелами інформації. Застосування штучного інтелекту на даному етапі дає змогу значно автоматизувати етапи збору та обробки даних, що сприяє зменшенню людського втручання та підвищує точність отриманої інформації.

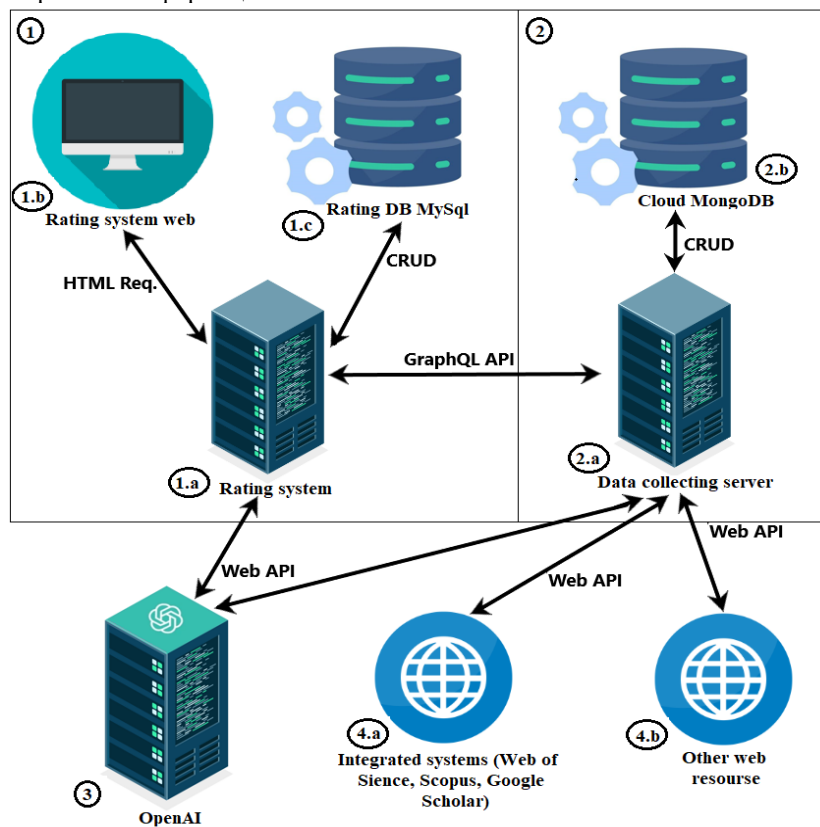


Рис. 1. Загальна архітектура системи рейтингування з новим модулем

Зовнішні веб-джерела. Система інтегрується з такими зовнішніми веб-джерелами, як DSpace або Електронна бібліотека НАПН України, та іншими (4.a) [5]. Це забезпечує доступ до додаткового широкого спектра наукових ресурсів, що дозволяє забезпечити більшу повноту даних, а також для автоматизації процесу збору інформації.

Інтеграція з відомими базами публікацій. Система має пряму інтеграцію з такими авторитетними базами публікацій, як Web of Science, Google Scholar, Scopus (4.b) [6-8]. Ці бази є провідними у світі наукових досліджень, тому інтеграція з ними дозволяє автоматично отримувати інформацію про наукові публікації викладачів. Це також підвищує точність та повноту даних, що використовуються для рейтингування науково-педагогічних працівників, сприяючи більш об'єктивній оцінці їхньої наукової діяльності.

Таким чином, обрана технологічна архітектура системи забезпечує високу продуктивність, гнучкість та масштабованість, що дозволяє ефективно збирати, аналізувати та інтегрувати дані про наукову діяльність

у процес створення звітів викладачів. Кожен елемент системи виконує свої певні функції, і вони можуть працювати паралельно, що додатково підвищує загальну ефективність та швидкість обробки даних.

Рейтингова система забезпечує процес оцінювання діяльності науково-педагогічних працівників, формування звітів, а також рейтингів співробітників, кафедр та факультетів (рис. 2). Система підтримує чотири основні ролі: адміністратор, завідувач кафедри, викладач та рецензент. Кожна з цих ролей авторизується через спільний інтерфейс, але має доступ до різних функціональних можливостей залежно від рівня прав. Адміністратор несе відповідальність за створення критеріїв, необхідних для формування звітів, і сам звіт для подальшого заповнення викладачами. Після того як звіт активується, викладачі можуть почати його заповнювати [9].

Ця система функціонує на основі фреймворку Laravel для клієнтської частини, тоді як серверна частина написана на PHP. Під час заповнення звітів викладачами, система інтегрується з новим модулем збору даних, що дозволяє автоматично отримувати інформацію про наукові публікації. Штучний інтелект виконує фільтрацію та категоризацію цих даних, розподіляючи наукові роботи працівників за відповідними критеріями у звіті. Це дозволяє частково автоматизувати процес заповнення звіту, і викладач отримує вже попередньо заповнений звіт, що містить зібрані основні дані.

Викладач може за необхідності редагувати ці дані або додавати детальнішу інформацію, та заповнити інші специфічні критерії звіту. Після завершення заповнення, звіт відправляється на перевірку завідувачу кафедри, а далі за умови коректності – рецензентам для остаточного затвердження. Такий підхід значно спрощує та прискорює процес формування звітів, знижуючи навантаження на викладачів і мінімізуючи ймовірність помилок.

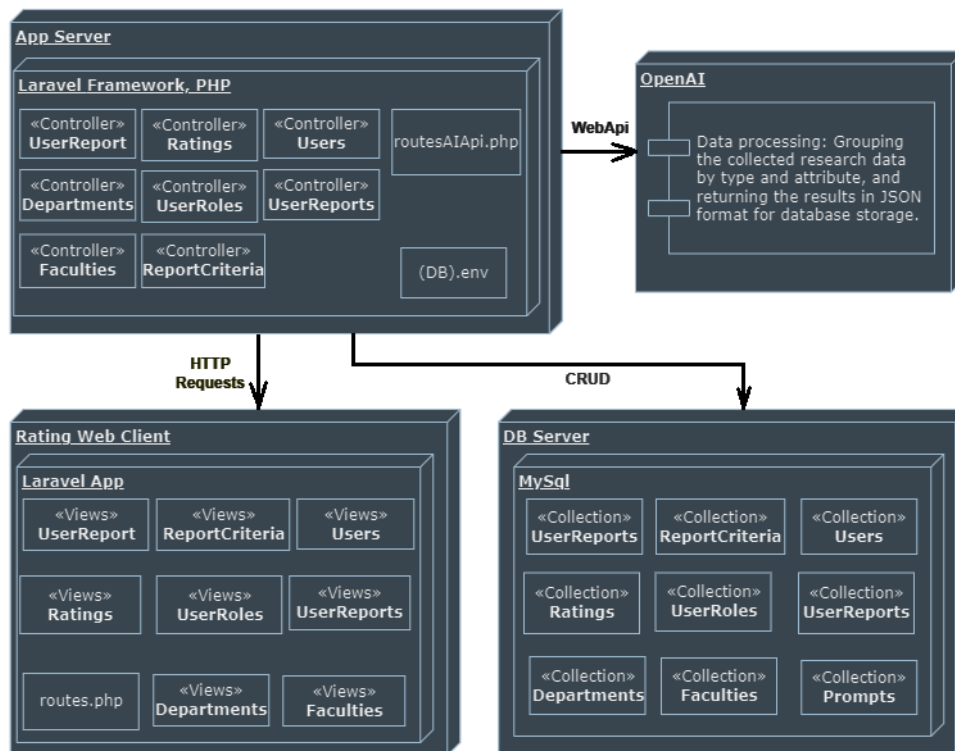


Рис. 2. Діаграма компонентів веб-системи рейтингування

Модуль збору даних (рис. 3) може функціонувати без веб-інтерфейсу, оскільки ключовим компонентом для нього є Node.js. Це середовище дозволяє виконувати серверний код і забезпечує стабільну роботу нового модуля.

Для побудови веб-сервера використано веб-фреймворк Express, що відомий своєю гнучкістю та мінімалістичністю. Він дозволяє швидко розгорнути веб-сервіси, що обробляють запити клієнтів. Інтеграція GraphQL в цей стек технологій дає можливість точно керувати запитом до бази даних і надавати користувачам саме ті дані, які їм потрібні, оптимізуючи тим самим взаємодію між клієнтами й сервером. Ця комбінація технологій знижує навантаження на мережу, зменшує обсяг передаваних даних та підвищує загальну продуктивність системи.

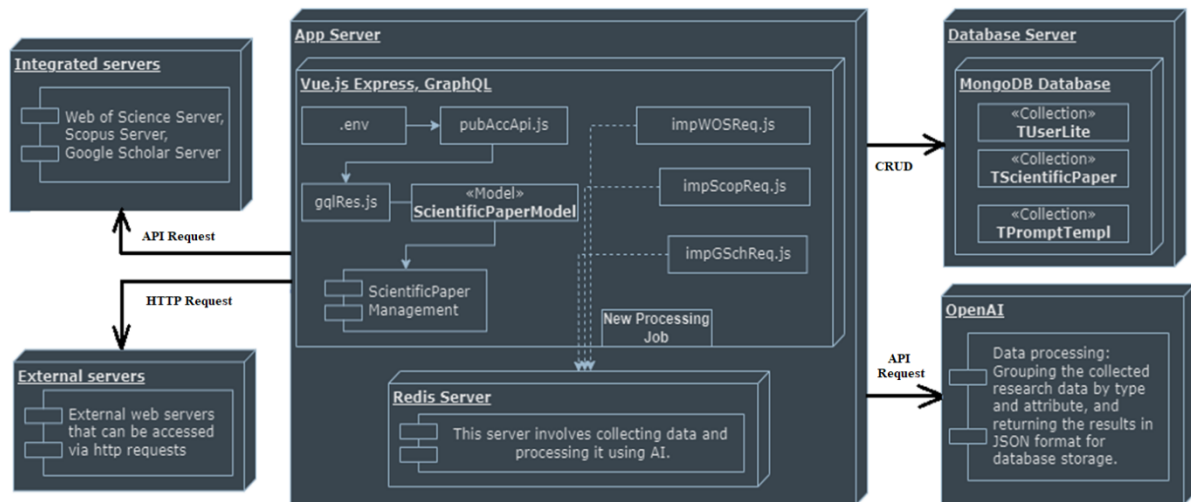


Рис. 3. Діаграма компонентів інтелектуального модуля збору даних

Конфігурація сервера зберігається у файлі .env, що є стандартною практикою для безпечного зберігання налаштувань. Точки входу для API визначені у файлі pubAccApi.js, а бізнес-логіка моделі описана у відповідних моделях. Обробка запитів та мутацій GraphQL здійснюється через налаштування, які зберігаються у файлі gqlRes.js.

Для обробки публікацій, які імпортуються з внутрішніх та зовнішніх джерел, таких як Web of Science, Google Scholar, Scopus, використовується Redis Server. Redis забезпечує швидке та ефективне керування чергами завдань завдяки тому, що зберігає дані в оперативній пам'яті. Це особливо важливо для інтеграції з інтелектуальною обробкою даних, що здійснюється через OpenAI. Такий підхід дозволяє швидко обробляти великий обсяг даних і забезпечує їх правильну класифікацію та зберігання через API.

Процес збору інформації виглядає так: спочатку сервер робить API запит до системи рейтингування, щоб отримати дані про корисувача (ППБ, посада, сфера наукової діяльності, ORCID, ScopusId, GoogleScholarID) для пошуку його наукових публікацій. Надалі ці дані відправляються в окрему, створену "задачу" в Redis (яка опрацьовується паралельно). Потім відбувається перебір цільових веб ресурсів, щоб парсити веб сторінки кожного з них. Під час парсингу [10-11] визначаються основні структури і елементи на веб ресурсі, і збираються основні метадані публікацій з html елементів (headers, titles, <body>, <context>, <label>, <div> eth) [12]. Після первинного збору даних з різних наукових ресурсів, наступним кроком є застосування алгоритмів штучного інтелекту (AI) для їх аналізу (видаляються всі зайві дані з контексту респонсу від веб-ресурсів, а саме – теги, щоб працювати лише з самим текстом та звільнити більше tokenів для самого запиту до AI). Застосовуючи сучасні алгоритми обробки тексту, штучний інтелект здатен з високою точністю виявляти важливі елементи тексту, такі як назви статей, авторів, дати публікацій та короткі описи. Це досягається через розподіл тексту на логічні частини, аналіз контексту кожного елемента та класифікацію відповідно до визначених категорій, таких як тип наукової роботи, журнал або тема дослідження [13].

Основний механізм, за допомогою якого AI здійснює цю обробку, полягає у використанні спеціально підготовлених промптів (prompt engineering) [14]. Промпти є інструкціями або шаблонами запитів, що задають моделі точну структуру та контекст, в яких вона повинна працювати. Наприклад, промпти можуть містити вказівки на те, які саме дані слід виділити із тексту та як їх організувати у відповідний формат. Залежно від складності аналізованих даних, промпти можуть варіюватися за своєю деталізацією: від базових інструкцій для витягу авторів і назв статей до більш складних, багаторівневих запитів, що включають аналіз ключових слів та виявлення цитувань. Ключовою особливістю роботи з промптами є їх адаптивність. Шаплони можуть бути оновлені та оптимізовані на основі типу даних або специфічних вимог до класифікації інформації [15]. Наприклад, якщо дані з певного наукового ресурсу мають специфічну структуру або формат, промпти можуть бути відповідним чином змінені, щоб врахувати ці особливості (рис. 4). Це дозволяє змінювати і покращувати промпти (при потребі), щоб підвищити ефективність і точність аналізу AI, мінімізуючи можливі помилки при обробці масивних обсягів не структурованих даних. Завдяки такому підходу до автоматизації обробки інформації, системи на основі AI здатні не тільки швидко аналізувати наукові дані, але й отримувати чітко структуровані результати, готові до подальшого збереження і використання в системі рейтингування.

Промпт

Активний

Так

Назва:

Розпізнавання та групування статей користувача

Prompt

В тексті подано інформацію про різні статті та їх авторів, але формат представлення даних не визначений, тобто може бути різним. Знайди і виведи список статей, автором яких є {author}.

Текст:

{test_tp_processing}

Завдання:

1. Знайти всі згадки про статті та їх авторів у тексті.
2. Визначити, які статті написані автором {author}.
3. Скласти список назв цих статей по шаблону (author, co_authors, title, publication_type(optional), description(optional), publication_date(optional), number_pages(optional)) і поверни у форматі json.
4. Якщо якісь значення з шаблону відсутні, або їх не вдалося визначити, залиши їх пустими (null).
5. Перевірити список на наявність дублікатів (при наявності - видалити).
6. Якщо отримані назви статей написані різними мовами, треба перекласти їх на одну із мов. Якщо назви статей схожі (мають однакові або майже однакові назви, з невеликими змінами або синонімами), залишити лише одну унікальну версію статті в списку.
7. Повернути результат у форматі JSON.

Рис. 4. Шаблон промпта у системі для обробки датасету статей

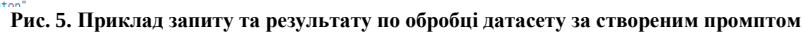
Моделі обробки природної мови (NLP), такі як ChatGPT, використовують сучасні методи для розуміння та аналізу тексту. Основні аспекти, які дозволяють цим моделям розрізняти і аналізувати текст, що включає:

- Контекстуальне розпізнавання. ChatGPT базується на трансформерах, які навчені на великій кількості текстових даних і можуть ефективно розпізнавати контекст та значення слів і фраз. Моделі навчаються на текстах, що містять різні формати написання слів, що дозволяє моделі зв'язувати різні форми написання певного слова або терміна.
- Синонімія і варіації. Моделі NLP навчені розпізнавати синонімічні вирази та варіації. Це досягається шляхом аналізу великого корпусу текстів, що дозволяє моделям визначати, що різні форми написання одного слова (включаючи кирилицю) є еквівалентними. Тобто, ChatGPT може визнати, що "Мельник Андрій" і "Andrii Melnyk" вказують на одну і ту ж особу, незважаючи на різні мовні форми.
- Контекстуальне розуміння. Моделі NLP здатні розуміти текст в контексті його використання. Це дозволяє ChatGPT зв'язувати різні варіанти написання слів і синонімічні вирази, забезпечуючи точність і відповідність у розпізнаванні та обробці тексту в межах контексту цілого датасету [16].

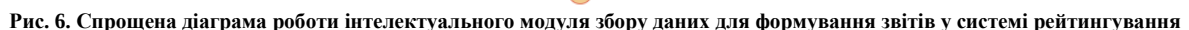
Отриманий з веб-ресурсів контент, включаючи всі основні елементи сторінки, спочатку очищується на сервері. Цей крок є необхідним, оскільки структурні HTML-елементи не мають значної аналітичної вартості та можуть збільшити обсяг переданих даних. Зайве навантаження може погіршити продуктивність запитів до ChatGPT API, адже збільшується кількість токенів. Видалення непотрібних HTML-елементів дозволяє значно оптимізувати процес і зменшити використання токенів, що особливо критично при обробці великих обсягів даних.

Очищений контент, що вже підготовлений для подальшого аналізу, передається до штучного інтелекту через API-запит для обробки. Такий підхід дозволяє зосередитися на основному контексті даних, що має значення для звіту користувача, а не на технічних деталях/елементах структури веб-сторінок. На рисунку 5 показано приклад запиту по API та відповідь від AI. Цей приклад демонструє високу ефективність використання кастомних шаблонів, які дають змогу чітко формувати запит і збирати всю необхідну інформацію з текстових даних.

Під час створення промпту описуються конкретні завдання, які потрібно виконати штучним інтелектом над заданим датасетом, а також очікуваний формат результату. Шаблон промпту кастомний та містить динамічні параметри, які заповнюються на сервері, безпосередньо перед самим реквестом. У цьому прикладі, сформовано результат у вигляді масиву елементів (статей) із такими параметрами: author, co_authors, title, publication_type(optional), description(optional), publication_date(optional), number_pages(optional). Очікуваний результат отримується у форматі JSON, що дозволяє зручно працювати з результатуючими даними і ефективно використовувати їх у подальших процесах (рис. 5).



Основні етапи роботи інтелектуального модуля збору даних для формування звітів у системі рейтингування можна візуально представити у вигляді діаграми (рис. 6).



International Scientific-technical journal
«Measuring and computing devices in technological processes» 2025, Issue 1

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

У даній роботі проведено дослідження та спроектовано модуль для збирання та опрацювання даних про наукові роботи науково-педагогічних працівників закладу вищої освіти, що інтегрований в існуючу систему рейтингування. На основі архітектури фреймворку Laravel, система дозволяє автоматизувати процес збору та обробки інформації про наукову діяльність, що сприяє підвищенню точності та швидкості обчислення рейтингових показників.

Основні результати цієї роботи полягають у наступному:

1. Реалізовано автоматизований модуль збирання наукових робіт, яка дозволяє ефективно отримувати та систематизувати дані від науково-педагогічних працівників для подальшого використання у рейтинговій системі.

2. Впроваджено процес збору та аналізу наукових робіт, побудований на автоматизованій системі парсингу веб-ресурсів. Дані про наукові публікації збираються шляхом аналізу HTML-елементів веб-сторінок та зовнішніх сервісів і передаються через API до штучного інтелекту. AI здійснює первинну класифікацію, розбір тексту, групування за категоріями та виявлення ключових даних по створеному промпту. Після цього проаналізовані дані зберігаються в базі даних для подальшого використання при заповненні звіту науково-педагогічних працівників.

3. Реалізовано інтеграцію з системою штучного інтелекту для автоматичного аналізу і збору наукових робіт. За допомогою AI система здатна опрацювати несистематизовані датасети (отримані із зовнішніх сервісів або після парсингу даних) на основі спеціально налаштованих промптів. Адміністратори системи можуть змінювати та додавати нові промпти, що дозволяє гнучко налаштовувати запити до AI.

4. На основі вдосконаленої архітектури системи рейтингування забезпечено автоматичне наповнення рейтингових показників, що зменшує людський фактор та можливість виникнення помилок, а також прискорює ці процеси.

5. Реалізована система побудована таким чином, що може легко масштабуватися, адаптуватися до нових вимог та розширюватися за рахунок додавання нових функцій або інтеграції з іншими модулями. Для прикладу, в майбутньому заплановано розгорнути можливість використання комплексу моделей AI та комунікувати з ними, наприклад, з допомогою API, щоб збільшити ліміти використання, або побудувати свою модель для використання в системі рейтингування.

Впроваджений модуль дозволяє не тільки підвищити ефективність процесів оцінки наукової діяльності працівників, але й сприяє загальному покращенню функціонування систем рейтингування у закладах вищої освіти, забезпечуючи прозорість, точність та об'єктивність оцінок та спрощує процес заповнення та перегляду звітів користувачами.

Література

1. Y. Labzhynskiy, A. Kilchenko, V. Kovalenko. The role of information and digital technologies for evaluating the effectiveness of scientific and pedagogical activities. Report of the scientific-practical conference of the Institute of Information Technologies and Teaching Aids of the National Academy of Sciences of Ukraine: materials of the scientific-practical conference, February 11, 2021, Kyiv / editor: O.P. Pinchuk, N.V. Yaskova – Kyiv: IITZN National Academy of Sciences of Ukraine, 2021. – 163 p.
2. V. Y. Bykov, O. M. Spirin, A. O. Biloshchytskyi, A. Y. Kuchansky, O. V. Dikhtiarenko, and O. V. Novytskyi, “Open Digital Systems For Assessment Of Pedagogical Research Results”, ITLT, vol. 75, no. 1, pp. 294–315, Feb. 2020, DOI: 10.33407/itlt.v75i1.3589.
3. Chatgpt. [Електронний ресурс]. Режим доступу: <https://chat.openai.com/>, Access Date: 01.12.2024.
4. OpenAI. [Електронний ресурс]. Режим доступу: <https://platform.openai.com/docs/overview/>, Access Date: 10.12.2024.
5. DSpace at West Ukrainian National University. [Електронний ресурс]. Режим доступу: <http://dspace.wunu.edu.ua/>, Access Date: 08.12.2024.
6. Elsevier Developer Portal, Research Products APIs. [Електронний ресурс]. Режим доступу: <https://dev.elsevier.com/>, Access Date: 05.12.2024.
7. Web of Science API Expanded. [Електронний ресурс]. Режим доступу: <https://developer.clarivate.com/apis/wos/>, Access Date: 20.12.2024.
8. Google Scholar Author Articles API – SerpApi. [Online] Available: <https://serpapi.com/google-scholar-author-articles/>.
9. A. Pukas, A. Simak, S. Shandruk, L. Bilovus, V. Stepanenko and A. Demianiuk, “Features of Implementation the Academic Staff Performance Appraisal System”, Proceedings of the 12th International Conference on Advanced Computer Information Technologies, ACIT’2022, Ruzomberok, Slovakia, 26-28 September 2022, pp. 568 – 571.

10. A. Pukas, A. Simak, T. Goncharuk-Cholach, O. Konoplińska, V. Zabchuk, "Intelligent Analyzing Module in the Academic Staff Performance Appraisal System" 2023, X International Scientific Conference "Information Technology and Implementation" (IT&I-2023). Conference Proceedings, Kyiv, Ukraine, pp. 423-429.
11. A. Kovbasistyi, A. Melnyk, M. Dyvak, V. Brych and I. Spivak, "Method for detection of non-relevant and wrong information based on content analysis of web resources," 2017 XIIIth International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), Lviv, Ukraine, 2017, pp. 154-156, doi: 10.1109/MEMSTECH.2017.7937555.
12. M. Radilova, P. Kamencay, R. Hudec, M. Benco and R. Radil, "Tool for Parsing Important Data from Web Pages," Applied Sciences. 12, 2022, 12031, doi: 10.3390/app122312031.
13. Rahib I., "The Rise of GPT-3: Implications for Natural Language Processing and Beyond", International Journal of Research Publication and Reviews, Vol 4, no 3, pp 4893-4903 March 2023, DOI:10.55248/gengpi.2023.4.33987.
14. K. Hooper, "Use OpenAI (GPT-3) to Create an AI Email Parser in JavaScript", 2023 [Electronic resource] – Available: <https://ai.plainenglish.io/using-openai-gpt-3-to-create-an-ai-email-parser-in-javascript-480029ee873f>.
15. J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, D. Schmidt, "A prompt pattern catalog to enhance prompt engineering with chatgpt", 2023. doi:10.48550/arXiv.2302.11382.
16. R. van Dinter, B. Tekinerdogan, C. Catal, Automation of systematic literature reviews: A systematic literature review, Information and Software Technology 136 (2021) 106589, DOI: 10.1016/j.infsof.2021.106589.

References

1. Y. Labzhynskyi, A. Kilchenko, V. Kovalenko. The role of information and digital technologies for evaluating the effectiveness of scientific and pedagogical activities. Report of the scientific-practical conference of the Institute of Information Technologies and Teaching Aids of the National Academy of Sciences of Ukraine: materials of the scientific-practical conference, February 11, 2021, Kyiv / editor: O.P. Pinchuk, N.V. Yaskova – Kyiv: IITZN National Academy of Sciences of Ukraine, 2021. – 163 p.
2. V. Y. Bykov, O. M. Spirin, A. O. Biloshchytskyi, A. Y. Kuchansky, O. V. Dikhtiarenko, and O. V. Novytskyi, "Open Digital Systems For Assessment Of Pedagogical Research Results", IJLT, vol. 75, no. 1, pp. 294–315, Feb. 2020, DOI: 10.33407/ijlt.v75i1.3589.
3. ChatGPT. [Online]. Available: <https://chat.openai.com/>, Access Date: 01.12.2024.
4. OpenAI. [Online]. Available: <https://platform.openai.com/docs/overview/>, Access Date: 10.12.2024.
5. DSpace at West Ukrainian National University. [Online] Available: <http://dspace.wunu.edu.ua/>, Access Date: 08.12.2024.
6. Elsevier Developer Portal, Research Products APIs. [Online] Available: <https://dev.elsevier.com/>, Access Date: 05.12.2024.
7. Web of Science API Expanded. [Online] Available: <https://developer.clarivate.com/apis/wos/>, Access Date: 20.12.2024.
8. Google Scholar Author Articles API – SerpApi. [Online] Available: <https://serpapi.com/google-scholar-author-articles/>.
9. A. Pukas, A. Simak, S. Shandruk, L. Bilovus, V. Stepanenko and A. Demianiuk, "Features of Implementation the Academic Staff Performance Appraisal System", Proceedings of the 12th International Conference on Advanced Computer Information Technologies, ACIT'2022, Ruzomberok, Slovakia, 26-28 September 2022, pp. 568 – 571.
10. A. Pukas, A. Simak, T. Goncharuk-Cholach, O. Konoplińska, V. Zabchuk, "Intelligent Analyzing Module in the Academic Staff Performance Appraisal System" 2023, X International Scientific Conference "Information Technology and Implementation" (IT&I-2023). Conference Proceedings, Kyiv, Ukraine, pp. 423-429.
11. A. Kovbasistyi, A. Melnyk, M. Dyvak, V. Brych and I. Spivak, "Method for detection of non-relevant and wrong information based on content analysis of web resources," 2017 XIIIth International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), Lviv, Ukraine, 2017, pp. 154-156, doi: 10.1109/MEMSTECH.2017.7937555.
12. M. Radilova, P. Kamencay, R. Hudec, M. Benco and R. Radil, "Tool for Parsing Important Data from Web Pages," Applied Sciences. 12, 2022, 12031, doi: 10.3390/app122312031.
13. Rahib I., "The Rise of GPT-3: Implications for Natural Language Processing and Beyond", International Journal of Research Publication and Reviews, Vol 4, no 3, pp 4893-4903 March 2023, DOI:10.55248/gengpi.2023.4.33987.
14. K. Hooper, "Use OpenAI (GPT-3) to Create an AI Email Parser in JavaScript", 2023 [Electronic resource] Available: <https://ai.plainenglish.io/using-openai-gpt-3-to-create-an-ai-email-parser-in-javascript-480029ee873f>.
15. J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, D. Schmidt, "A prompt pattern catalog to enhance prompt engineering with chatgpt", 2023. doi:10.48550/arXiv.2302.11382.
16. R. van Dinter, B. Tekinerdogan, C. Catal, Automation of systematic literature reviews: A systematic literature review, Information and Software Technology 136 (2021) 106589, DOI: 10.1016/j.infsof.2021.106589.