УДК 004.4+004.6
DOI: https://doi.org/10.17721/1812-5409.2024/1.26

Сергій ІВАНОВ, канд. фіз.-мат. наук, доц.
ORCID ID: 0000-0002-4339-9192
e-mail: smivanov87@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Євген ІВОХІН, д-р фіз.-мат. наук, проф.
ORCID ID: 0000-0002-5826-7408
e-mail: ivohin@univ.kiev.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Михайло МАХНО, канд. техн. наук, доц.
ORCID ID: 0000-0001-5045-1705
e-mail: mykhailo.makhno@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

АВТОМАТИЗАЦІЯ ОБЛІКУ ПУБЛІКАЦІЙ З ВИКОРИСТАННЯМ ІНТЕРФЕЙСУ ПРИКЛАДНОГО ПРОГРАМУВАННЯ ORCID

Запропоновано процедуру автоматизованого обліку публікацій на основі використання Rest API бази ORCID. Описано актуальність обліку публікацій та обґрунтовано необхідність застосування різних технологій створення сховищ бібліографічних даних. Проаналізовано можливість застосування технології API у найвідоміших базах публікацій, таких як Web of science, SCOPUS, Crossref, Google Scholar та ORCID. Обґрунтовується можливість застосування бази ORCID. Наведено схему завантаження публікацій з бази ORCID за визначеними реєстраційними номерами на основі сервісів, реалізованих мовами програмування Python та MatLab. Отримані дані в JSON або XML підлягають подальшому парсингу. Наведено MatLab-функції для отримання структури з XML (JSON) форматів даних. Додатково розглянуто алгоритм пошуку дублікатів публікацій під час проведення їхнього обліку. Сформульовано підходи для уникнення дублювання публікацій у базах даних, що базуються на застосуванні алгоритму Левенштейна для оцінювання схожості. Запропоновано проводити транслітерацію кирилиці в латиницю для забезпечення однозначності та коректного порівняння текстових даних. Для накопичення та актуалізації даних щодо публікаційної активності розроблено базу даних MySql. Заголовок таблиці публікацій бази даних доповнено спеціальним атрибутом, у якому зберігаються результати перетворення назв кирилицею у відповідні назви латиницею. Рекомендується використовувати індексацію полів таблиць бази даних (INDEX) за різними атрибутами, що дало змогу суттєво підвищити ефективність пошуку, обробки та порівняння даних. Запропоновано застосування функції Soundex() в ролі засобів СУБД MySQL для визначення рівня співзвучності тем публікацій за додатковими параметрами. Практична реалізація алгоритму пошуку дублікатів публікацій та їх нумерації засвідчила конструктивність запропонованого підходу, що було підтверджено під час проведення наповнення бази даних. Пропонована стаття представляє інтерес для розробників програмного забезпечення.

Ключові слова: ***API, ORCID, облік публікацій, Python, MatLab, алгоритм, XML, пошук дублікатів.***

Класифікація відповідно до AMS 2020: 68P05, 68P10, 68P15, 68P20, 68W32.

Вступ

Публікаційна активність є основою для оцінювання дослідницької діяльності наукових або науково-педагогічних працівників, за допомогою якої формується одна з найважливіших складових звітності наукових установ (організацій) та університетів загалом. Облік публікацій супроводжується цілим спектром актуальних завдань: вчасне збирання опублікованих праць у єдиному форматі, перевірка даних і знаходження дубльованих публікацій, необхідність складання звітності з публікаційної активності тощо. З огляду на це варто зазначити, що індексування публікацій у різних базах, зокрема в Scopus або Web of Science, що є обов'язковою частиною обліку, потребує розроблення спеціальних методів і засобів автоматизації. Основним способом автоматизованого обліку публікацій є створення єдиної бази даних та безпосереднє введення даних праць авторів. Однак за наявності багатьох ресурсів, де можуть індексуватися роботи авторів, можна значно полегшити процедуру введення даних шляхом автоматизованого завантаження облікових даних публікацій.

Використанню зовнішніх джерел для обліку праць авторів присвячено лише декілька робіт, зокрема у статті (Горбачевський, 2022) розглянуто алгоритм автоматизації обліку публікацій наукових підрозділів на основі використання API (інтерфейсу прикладного програмування) ORCID (Open Researcher and Contributor IDentifier), що означає відкритий ідентифікатор дослідника та співавтора (учасника), який підтримується міжнародною некомерційною організацією, створеною дослідницьким співтовариством на користь усіх зацікавлених сторін, що займаються дослідницькою діяльністю (ORCID…, 2023). Ідея об‘єднання даних з різних джерел, а також синтез програм, розроблення нових алгоритмів на основі технології API описані в багатьох роботах (May et al., 2024; Sotiropoulos, Chaliasos, & Su, 2024; Wang, Y. et al., 2024; Wu, D. et al., 2024).

Сам алгоритм використання технології API детально представлено в документації ресурсу ORCID, де одразу пропонується декілька API (із вихідним форматом JSON або XML), які дають змогу підключатися до реєстру ORCID з можливістю зчитування та запису. Деякі функції API вільно доступні для всіх (Public API), інші – лише для організацій – членів ORCID (Member and Premium Member API), за допомогою кожного з яких можна безкоштовно провести тестування на доступному тестовому сервері ізольованого програмного середовища (Collect Authenticated ORCID…, 2023).

Іншою авторитетною та потужною базою публікаційних даних є Web of Science Core Collection (британсько-американської публічної аналітичної компанії Clarivate Analitics) (Web of science core collection…, 2023), яка підтримує API для отримання повних метаданних включно з кількістю цитувань, адресами та філіями співавторів (учасників), а також даними про фінансування. Цей API забезпечує програмний доступ на основі REST до документів Web of Science із вихідним форматом JSON або XML з розширеними можливостями пошуку та фільтрації. Для доступу до цього API потрібна платна ліцензія. API зазвичай пропонується в чотирьох різних планах й обмежується кількістю документів і запитів Web of Science за секунду (Web of science API Expanded, 2023).

Ще однією великою базою даних публікацій є Scopus видавничої корпорації Elsevier, яка також надає API доступ до власних ресурсів. Некомерційним користувачам (дослідники в академічному, державному секторі та некомерційних

ISSN 1812-5409 (Print), ISSN 2218-2055 (Online)

установах) безкоштовно доступна більшість API (крім API SciVal і Embase) для здійснення доступу, що відповідає політиці Elsevier та обмеженням щодо використання (Elsevier Research products APIs, 2023). Для комерційних користувачів (дослідники у приватному секторі та комерційних установах) API доступні з ліцензією на API та передплатою (у межах прав на комерційне використання) (Elsevier Research products APIs, 2023). API дає змогу отримати дані цитування, метадані й анотації з наукових журналів, індексовані в Scopus та бази даних цитувань Elsevier.kl, а також одержати повнотекстові журнали та книги, опубліковані Elsevier на повнотекстовій платформі ScienceDirect. API надає можливість отримати дослідницькі показники з SciVal на платформі Elsevier для порівняльного аналізу продуктивності досліджень, інженерні ресурси (з Engineering Village), дані з бази біомедичних рефератів та індексації Embase, інформацію із хімічної бази даних Reaxys, а також дані про безпеку, ефективність, фармакокінетичні, метаболізуючі ферменти та інформацію щодо транспортера з однієї з баз даних Elsevier з можливістю повного пошуку PharmaPendium. API використовує для автентифікації протокол OAuth (Elsevier Research products APIs, 2023).

Google Scholar – велика база публікацій, що індексує повнотекстові роботи, забезпечує функцію пошуку, містить велику кількість європейських й американських рецензованих журналів (Google scholar, 2023). Використання цього ресурсу для автоматизації обліку публікацій не передбачається, оскільки не враховано можливості застосування API та під час розгляду профілів помічено залучення «зайвих» робіт, які не мають відношення до автора, особливо це виникає в авторів з найбільш частими або поширеними прізвищами.

Crossref – бібліографічна база даних, що зберігає інформацію про цитованість на основі технології DOI (Digital Object Identifier) та бібліографічні дані опублікованих робіт (CrossRef…, 2023). Crossref підтримується REST API з можливостями здійснювати пошук, фільтрування або вибірку метаданих із тисяч учасників, і результати повертаються у форматі JSON. Для використання REST API у цьому ресурсі не потрібна реєстрація (CrossRef. REST API…, 2023).

Вітчизняними вченими (Шершун та ін., 2020) була запропонована база даних обліку публікацій на прикладі Одеської національної академії харчових технологій та продемонстровано інтерфейс розробленого сайту, однак, на жаль, не розглянуто проблему дублювання публікацій, що може впливати на звітність, а також не надано опису способів використання інтерфейсів прикладного програмування API у розробленій інформаційній системі.

Одним з найперспективніших ресурсів для автоматизації обліку публікацій, що дає змогу за допомогою реалізації REST API запитів завантажувати бібліографічні дані і тим самим значно спростити процедуру введення в базу даних публікацій, є ORCID, можливості якого і пропонуємо використати.

Об'єктом проведених досліджень є процес обліку публікацій наукових, науково-педагогічних й інших працівників, що містить технологію розроблення та реалізацію алгоритмів автоматичного завантаження бібліографічних даних публікацій з ресурсу ORCID на основі доступного REST API за наявними реєстраційними ORCID-номерами авторів.

1. Основні результати

Безпосереднє введення кожної публікації до бази даних є доволі витратною операцією за часом, тому завантаження з доступних джерел бібліографічних даних публікацій значно спрощує та полегшує процес обліку публікацій. Однією з доступних інформаційних баз є ORCID, що підтримує технологію API. Зважаючи на це, запропоновано використати наявні технологічні можливості у вигляді типової схеми на основі виконання запитів до API та розробити процедуру автоматичного завантаження бібліографічних даних публікацій з бази ORCID до локальної бази даних.

Послідовність етапів процедури автоматичного завантаження публікацій (рис. 1)

Процедура автоматичного завантаження бібліографічних даних передбачає виконання такої послідовності дій:

1) **Автентифікація та отримання доступу до API ORCID:** для використання API ORCID необхідно зареєструватися як користувач й отримати доступні ключі API, які надають можливість виконувати запити до системи. Отримання ключів API передбачає отримання clientId та clientSecret (ORCID. Public API…, 2023).

2) **Формування запитів до API:** формуються HTTP-запити до ORCID API для отримання інформації про публікації заданого автора за його реєстраційним номером ORCID.

3) **Парсинг отриманих даних:** унаслідок оброблення запиту від ORCID API надходить відповідь у форматі XML або JSON, яка потребує проведення парсингу та перетворення у спеціальний формат даних, що дасть змогу зберегти її в локальній базі даних з метою подальшого обліку та/або аналізу.

4) **Збереження та синхронізація даних:** розібрані дані інтегруються в локальну базу даних, де вони можуть бути синхронізовані з наявними записами. Важливо забезпечити механізм виявлення й оброблення дублікатів.

Оновлення даних: алгоритм має регулярно оновлювати дані, збережені в локальній базі, для відображення будь-яких змін або нових публікацій у профілі ORCID автора. Застосування розробленої процедури автоматизованого завантаження бібліографічних даних доповнено можливістю їх ручного введення безпосереднього у локальне сховище, що може призводити до дублювання публікацій (Zhu et al., 2016; Ahlawat, & Sagar, 2022).

Назви тем публікацій можуть містити помилки, які з'являються або внаслідок ручного введення, або через конвертації в базі ORCID, що виникають, наприклад, у випадку наявності в назві статті формул тощо. Крім можливих фразеологічних розбіжностей, в окремих словах можуть зустрічатися спеціальні символи "*,^,%,.,$,#,@" тощо. Отже, традиційні способи пошуку дублікатів у базах даних, наприклад із застосуванням group by мови SQL, не можуть урегулювати цю проблему, що й визначило необхідність розроблення нового алгоритму пошуку дублікатів на основі фонетичної співзвучності.

Методика виявлення дублікатів публікацій

Як було підкреслено на кроці 4 процедури автоматизованого завантаження публікацій для виконання коректного збору та синхронізації бібліографічних даних, необхідно провести пошук й уникнення можливих дублікатів публікацій, які виникають з різних причин, наприклад:

- публікація має співавторів;
- публікація помилково занесена двічі;
- під час внесення інформації про публікацію назва була заведена з помилковими символами;
- публікація заведена двічі іншою мовою тощо.

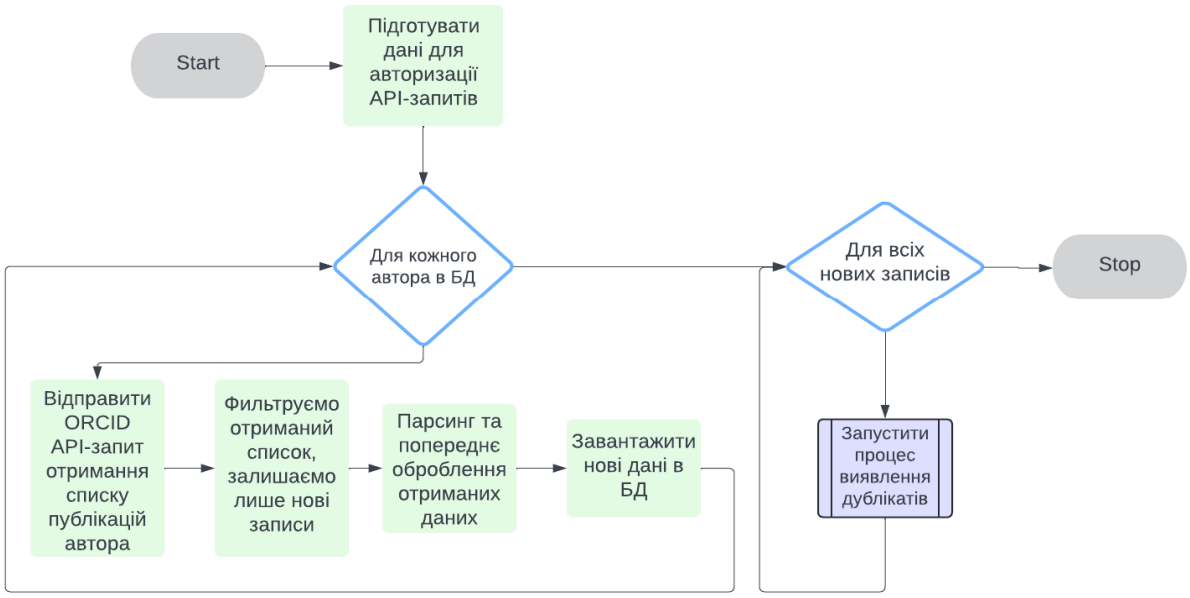

Рис. 1. Діаграма роботи процедури завантаження

Для уникнення дублювань публікацій у процесі її збереження в базі даних пропонується провести попереднє оброблення даних (Іванов, & Флакей, 2023) на основі таких дій, як:

• **Використання унікальних ідентифікаторів:** кожна публікація в ORCID зазвичай має унікальний ідентифікатор, такий як DOI (Digital Object Identifier) або інший постійний ідентифікатор. Перш ніж додавати нову публікацію до бази даних, перевіряємо, чи не існує вже запису з таким ідентифікатором.

• **Нормалізація даних:** перед порівнянням даних про публікації дані нормалізуються для забезпечення типовості формату. Це може передбачати перетворення всіх символів до нижнього регістру, видалення пробілів, символів пунктуації та інших несуттєвих елементів для подальшого технічного порівняння.

• **Використання алгоритмів виявлення схожості:** для виявлення дублікатів, що можуть мати незначні відмінності у назві або інших деталях, можна застосовувати алгоритми виявлення схожості, такі як алгоритм Левенштейна або інші способи виявлення подібності рядків. Це допоможе ідентифікувати публікації, що є дуже схожими, але не ідентичними.

• **Хешування атрибутів публікації:** запропоновано створювати хеш-суму для ключових атрибутів кожної публікації, таких як назва, автори, рік публікації тощо, і використовувати ці хеші для швидкого порівняння записів. Якщо хеш-суми збігаються, то записи потрібно додатково перевірити на потенційне дублювання.

• **Ручна перевірка та модерація:** наведені вище методики автоматичного виявлення дублікатів можуть суттєво покращити якість бази даних, але не можуть гарантувати точний результат, тому рекомендується додаткова ручна перевірка бази людиною-оператором. Для цього необхідно створити окремий користувацький інтерфейс, що дасть змогу користувачам переглядати вміст бази та виконувати фільтрацію та актуалізацію даних за різними атрибутами.

• **Маркування дублікатів:** у випадках виявлення дублікатів необхідно помітити знайдені дублікати публікацій (це може бути номер дубля), що проводиться у відповідному полі таблиці публікаційних даних.

• **Використання індексування та оптимізація пошуку:** ураховуючи велику потужність створеної бази даних, необхідно налаштувати інструменти для оптимізації роботи у вигляді індексів, щоб полегшити швидкий пошук для виявлення дублікатів та роботи користувачів із системою. Індексування основних полів, як-от назви публікацій, імен авторів і дати публікації, може спростити та суттєво пришвидшити процес ідентифікації дублікатів.

Для реалізації пошуку та виявлення дублікатів публікацій у разі використання локальної бази даних MySql розроблено та впроваджено алгоритм (Іванов, & Флакей, 2023), основними ідейними моментами якого є:

• *Транслітерація кирилиці*: для забезпечення однозначності та коректного порівняння текстових даних створюється ще одне поле у відповідній таблиці бази даних MySql, у яке записують перетворення кирилиці у відповідну латиницю.

• *Фонетичне індексування за допомогою Soundex*(): на цьому етапі використовують убудовану функцію MySQL – Soundex(), яка застосовує фонетичний алгоритм для проведення індексації літер у рядку. Функція Soundex не лише визначає фонетичну схожість між словами та фразами, а й може ефективно вилучати зайві символи, такі як #, $, <>, пробіли та інші непередбачені символи. Отже, ця функція може виконувати доволі точне порівняння за звучанням, що зумовлює високу ефективність пошуку задубльованих публікацій.

• *Створення збереженої процедури в базі даних MySQL* для пошуку та маркування дублікатів у додатковому полі таблиці публікаційних даних.

Схема розробленої процедури (рис. 2) містить послідовне виконання: пошуку та маркування дублікатів публікацій за DOI; визначення для кожної теми публікації кількості співзвучних записів із застосуванням функції Soundex; оновлення поля, що відповідає за маркування задубльованих публікацій, ураховуючи кількість виявлених дублікатів (кожній публікації присвоюється відповідна мітка для подальшого визначення порядку задубльованих записів); оновлення записів, які відповідають певним умовам, що забезпечує точне порівняння лише необхідних публікацій.

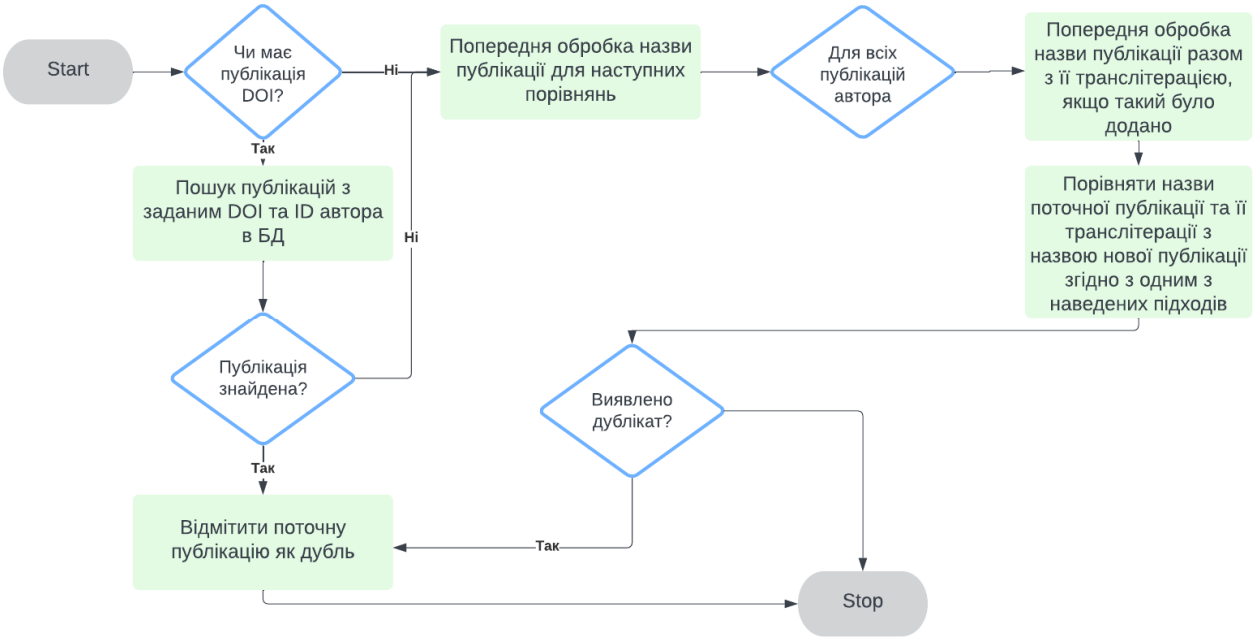

Рис. 2. Схема процедури пошуку дублікатів публікацій

2. Приклади роботи системи

У ролі прикладів роботи програми продемонструємо, як можна виконати простий запит до ORCID API мовою Python для отримання списку публікацій з ORCID. Припускаємо, що вже були отримані відповідні ключі доступу.

```
import requests
import xml.etree.ElementTree as ET
def get_orcid_publications(orcid_ids):
    for orcid_id in orcid_ids:
        print(f"Обробка ORCID ID: {orcid_id}")
        url = f"https://pub.orcid.org/v3.0/{orcid_id}/works"
        headers = { "Accept": "application/vnd.orcid+xml" }
        response = requests.get(url, headers=headers)
        if response.status_code == 200:
            root = ET.fromstring(response.content)
            ns = {
                'o': 'http://www.orcid.org/ns/orcid',
                'w': 'http://www.orcid.org/ns/work'
            }
            for work_summary in root.findall('.//o:work-summary', ns):
                # Отримання та виведення назви публікації
                title_element = work_summary.find('.//w:title', ns)
                if title_element is not None:
                    title = title_element.text
                    print(f"  Назва публікації: {title}")
                else:
                    print("  Назва публікації не знайдена")
        else:
            print(f"  Помилка при запиті для ORCID ID {orcid_id}: {response.status_code}")
orcid_ids = ["0000-0002-1825-0097", "0000-0001-2345-6789"]  # Приклади ORCID ID
get_orcid_publications(orcid_ids)
```

Цей код ініціює запит до ORCID API для отримання даних про публікації за вказаним ORCID номером та виводить назви отриманих публікацій. Важливо зауважити, що у разі використання аутентифікації для доступу до API ORCID необхідно реалізувати OAuth автентифікацію (API Tutorial: Get an Authenticated ORCID iD, 2023).

Інший приклад стосується використання сучасної версії MatLab (R2023b), що дає змогу отримувати дані публікацій з бази ORCID (Public API) одразу у вигляді структури за допомогою функції webread за наявним http-запитом.

Створимо змінну-масив W з номерами ORCID

```
W={'IvohinE', '0000-0002-5826-7408';
   'IvanovSM','0000-0002-4339-9192';
   'MakhnoM','0000-0001-5045-1705';
   };
P=[]; Y={};                  % Допоміжні змінні
for i=1:length(W(:,1))    % Отримання даних з бази orcid за http-запитом
t=webread(string(strcat('https://pub.orcid.org/v3.0/',W(i,2),'/works')));
```

```
    try  n=length(t.group);
     catch
        continue
     end
        for j=1:n
           Y(1,1)=W(i,2);
           try Y(1,2)={string(t.group(j).work_summary.put_code)}; catch Y(1,2)={"not found"}; end
           try Y(1,3)={string(t.group(j).work_summary.source.source_name.value)}; catch Y(1,3)={"not found"}; end
           try Y(1,4)={string(t.group(j).work_summary.journal_title.value)}; catch Y(1,4)={"not found"}; end
           try Y(1,5)={string(t.group(j).work_summary.type)}; catch Y(1,5)={"not found"}; end
           try Y(1,6)={string(t.group(j).work_summary.publication_date.year.value)}; catch Y(1,6)={"not found"}; end
           try Y(1,7)={string(t.group(j).work_summary.publication_date.month.value)}; catch Y(1,7)={"not found"}; end
           try Y(1,8)={string(t.group(j).work_summary.publication_date.day.value)}; catch Y(1,8)={"not found"}; end
           try Y(1,9)={string(t.group(j).work_summary.title.title.value)}; catch Y(1,9)={"not found"}; end
           P=[P; Y]; % Дозаписування у таблицю P даних з кожної ітерації циклу
        end
end
```

У підсумку у змінній-масиві P будуть перебувати деякі основні дані публікацій для трьох авторів, які в публічному доступі розміщуються в базі ORCID. У наведеному прикладі завантажуються такі дані: Номер ORCID, Номер публікації у базі ORCID, ПІБ автора, Назва журналу, Тип публікації, Рік, Місяць, День, Назва публікації.

У разі отримання XML (JSON) формату його можна зберегти у "iks.txt" файл та застосувати функцію readstruct для отримання структури, наприклад S = readstruct("iks.txt","FileType","xml").

Дискусія і висновки

У роботі було запропоновано процедуру завантаження бібліографічних публікацій авторів за їх реєстраційними номерами ORCID на основі використання інтерфейсу прикладного програмування API та сервісів, реалізованих мовами програмування Python і MatLab. Отримані дані в JSON або XML підлягають подальшому парсингу та перетворенню у спеціальний формат, що використовується для запису в локальну базу даних MySQL. Запропоновано та впроваджено процедуру пошуку дублікатів публікацій при проведенні їхнього обліку. Сформульовано підходи для уникнення дублювання публікацій у базах даних, що базуються на транслітерації кирилиці в латиницю для забезпечення однозначності та коректного порівняння текстових даних. Запроваджено використання індексації таблиць бази даних (INDEX) за різними атрибутами, що дало змогу суттєво підвищити ефективність пошуку, оброблення та порівняння даних. Запропоновано застосування функції Soundex() як засіб СУБД MySQL для визначення рівня співзвучності тем публікацій за додатковими параметрами. Практична реалізація алгоритму пошуку дублікатів публікацій та їхньої нумерації засвідчила конструктивність запропонованого підходу, що було підтверджено під час проведення наповнення бази даних бібліографічних публікацій.

Внесок авторів: Сергій Іванов – програмне забезпечення, валідація даних, написання – оригінальна чернетка; Євген Івохін – концептуалізація, методологія; Михайло Махно – написання, перегляд і редагування.

Подяка, джерела фінансування. Ця робота здійснювалася в Київському національному університеті імені Тараса Шевченка за проєктом "Інформаційна система для обліку публікацій, афілійованих з Університетом" (Наказ 299-32 від 14.04.2023 р.).

Список використаних джерел

Горбачевський, С. (2022). Автоматизація обліку публікацій наукових підрозділів на основі використання API ORCID. *Військова освіта*, 1 (45), 52–58.

Іванов, С. М,. & Флакей, Р. Р. (2023). Пошук задубльованих публікацій на основі фонетичної співзвучності тем. В Патрак та ін. (Ред.), *Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення. Інформаційні системи і технології,* 83, 43–45. ФОП Шпак В. Б. http://www.konferenciaonline.org.ua/ua/article/id-1505

Шершун, О. О., Титуренко, Ж. А., Зінченко, І. І., & Ольшевська, О. В. (2020). Розроблення автоматизованого ресурсу обробки даних науковців ОНАХТ з наукометричних баз даних. *Автоматизація технологічних і бізнес-процесів*, 12(3), 40–46.

Ahlawat, Anil, & Sagar, Kalpna. (2022). Automating Duplicate Detection for Lexical Heterogeneous Web Databases. *Recent Advances in Computer Science and Communications*, 15(4), e220322185588. https://dx.doi.org/10.2174/2666255813999200904170035

API Tutorial: Get an Authenticated ORCID iD. (2023, 20 November). https://info.orcid.org/documentation/api-tutorials/api-tutorial-get-and-authenticated-orcid-id

Collect Authenticated ORCID iDs and permissions. (2023, 20 November). https://info.orcid.org/hands-on-with-the-orcid-api/2-collect-authenticated-orcid-ids-and-permissions

CrossRef. Fact file 2018–2019 annual report. (2023, 20 November). https://www.crossref.org/pdfs/annual-report-factfile-2018-19.pdf

CrossRef. REST API. (2023, 20 November). https://www.crossref.org/documentation/retrieve-metadata/rest-api

Elsevier Research products APIs. (2023, 20 November). https://dev.elsevier.com

Google scholar. (2023, 20 November). https://scholar.google.com

May, Mahmoud, Walker, Robert J., & Denzinger, Jörg. (2024). API usage templates via structural generalization. *Journal of Systems and Software*, 210, 111974. https://doi.org/10.1016/j.jss.2024.111974

ORCID. Connecting research and researchers. (2023, 20 November). https://info.orcid.org/researchers

ORCID. Public API. (2023, 20 November). https://info.orcid.org/documentation/features/public-api

Sotiropoulos, Thodoris, Chaliasos, Stefanos, & Su, Zhendong. (2024). API-Driven Program Synthesis for Testing Static Typing Implementations. *Proc. ACM Program. Lang,* 8 (POPL), 62 (January 2024), 1850–1881. https://doi.org/10.1145/3632904

Wang, Y., Chen, L., Gao, C., Fang, Y., & Li, Y. (2024). Prompt enhance API recommendation: visualize the user's real intention behind this query. *Automated Software Engineering*, 31, 27. https://doi.org/10.1007/s10515-024-00425-0

Web of science core collection. (2023, 20 November). https://clarivate.com/cis/solutions/web-of-science-core-collection

Web of science API Expanded. (2023, 20 November). https://developer.clarivate.com/apis/wos

Welcome to MatLab. (2024, 20 March). https://matlab.mathworks.com

Wu, D., Feng, Y., Zhang, H., & Xu, B. (2024) Automatic recognizing relevant fragments of APIs using API references. *Automated Software Engineering* 31. Article 3. https://doi.org/10.1007/s10515-023-00401-0

Zhu, Weiheng, Yin, Jian, Deng, Yuhui, Long, Shun, & Qiu, Shiding. (2016). Efficient Duplicate Detection Approach for High Dimensional Big Data[J]. *Journal of Computer Research and Development*, 53(3), 559–570. DOI: 10.7544/issn1000-1239.2016.20148218

References

Ahlawat, Anil & Sagar, Kalpna. (2022). Automating Duplicate Detection for Lexical Heterogeneous Web Databases. *Recent Advances in Computer Science and Communications*, 15(4), e220322185588. https://dx.doi.org/10.2174/2666255813999200904170035

API Tutorial: Get an Authenticated ORCID iD. (2023, 20 November). https://info.orcid.org/documentation/api-tutorials/api-tutorial-get-and-authenticated-orcid-id

Collect Authenticated ORCID iDs and permissions. (2023, 20 November). https://info.orcid.org/hands-on-with-the-orcid-api/2-collect-authenticated-orcid-ids-and-permissions

CrossRef. Fact file 2018–2019 annual report. (2023, 20 November). https://www.crossref.org/pdfs/annual-report-factfile-2018-19.pdf

CrossRef. REST API. (2023, 20 November). https://www.crossref.org/documentation/retrieve-metadata/rest-api

Elsevier Research products APIs. (2023, 20 November). https://dev.elsevier.com

Google scholar. (2023, 20 November). https://scholar.google.com

Horbachevskyi, S. (2022). Automation of the accounting of publications of scientific units based on the use of the ORCID API. *Military education*, 1 (45), 52–58 [in Ukrainian].

Ivanov, S. M., & Flakei, R. R. (2023). Search for duplicate publications based on phonetic consonance of topics. In Patrak et al. (Eds.), *Information society: technological, economic and technical aspects of formation/ Information systems and technologies,* 83, 43–45. FOP Shpak V. B. http://www.konferenciaonline.org.ua/ua/article/id-1505 [in Ukrainian].

May, Mahmoud, Walker, Robert J., & Denzinger, Jörg. (2024). API usage templates via structural generalization. *Journal of Systems and Software*, 210, 111974. https://doi.org/10.1016/j.jss.2024.111974

ORCID. Connecting research and researchers. (2023, 20 November). https://info.orcid.org/researchers/

ORCID. Public API. (2023, 20 November). https://info.orcid.org/documentation/features/public-api/

Shershun, O. O., Tyturenko, Zh. A., Zinchenko, I. I., & Olshevska, O. V. (2020). Development of an automated data processing resource of ONAKHT scientists from scientometric databases. *Automation of technological and business processes*, 12(3), 40–46 [in Ukrainian].

Sotiropoulos, Thodoris, Chaliasos, Stefanos, & Su, Zhendong. (2024). API-Driven Program Synthesis for Testing Static Typing Implementations. *Proc. ACM Program. Lang,* 8 (POPL), 62 (January 2024), 1850–1881. https://doi.org/10.1145/3632904

Wang, Y., Chen, L., Gao, C., Fang, Y., & Li, Y. (2024). Prompt enhance API recommendation: visualize the user's real intention behind this query. *Automated Software Engineering*, 31, 27. https://doi.org/10.1007/s10515-024-00425-0

Web of science API Expanded. (2023, 20 November). https://developer.clarivate.com/apis/wos

Web of science core collection. (2023, 20 November). https://clarivate.com/cis/solutions/web-of-science-core-collection

Welcome to MatLab. (2024, 20 March). https://matlab.mathworks.com

Wu, D., Feng, Y., Zhang, H., & Xu, B. (2024) Automatic recognizing relevant fragments of APIs using API references. *Automated Software Engineering* 31. Article 3. https://doi.org/10.1007/s10515-023-00401-0

Zhu, Weiheng, Yin, Jian, Deng, Yuhui, Long, Shun, & Qiu, Shiding. (2016). Efficient Duplicate Detection Approach for High Dimensional Big Data[J]. *Journal of Computer Research and Development*, 53(3), 559–570. DOI: 10.7544/issn1000-1239.2016.20148218

Отримано редакцією журналу: 20.03.24
Прорецензовано: 15.04.24
Схвалено до друку: 20.04.24

Serhii IVANOV, PhD (Phys. & Math.), Assoc. Prof.
ORCID ID: 0000-0002-4339-9192
e-mail: smivanov87@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Eugene IVOHIN, DSc (Phys. & Math.), Prof.
ORCID ID: 0000-0002-5826-7408
e-mail: ivohin@univ.kiev.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Mykhailo MAKHNO, PhD (Engin.), Assoc. Prof.
ORCID ID: 0000-0001-5045-1705
e-mail: mykhailo.makhno@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

AUTOMATION OF ACCOUNTING OF PUBLICATIONS USING THE ORCID APPLICATION PROGRAMMING INTERFACE

The procedure for automated accounting of publications based on the use of Rest API of the ORCID database is proposed. The relevance of publication accounting is described. The importance of using various technologies for creating bibliographic data repositories is substantiated. The possibility of using API technology in the most famous publication databases such as Web of science, SCOPUS, Crossref, Google Scholar, and ORCID was analyzed. The possibility of using the ORCID database is substantiated. The scheme for downloading publications from the ORCID database by specified registration numbers based on services implemented in the Python and MatLab programming languages is given. The received data in JSON or XML is subject to further parsing. MatLab functions for obtaining a structure from XML (JSON) data formats are provided. In addition, the algorithm for finding duplicate publications during their accounting is considered. Approaches to avoid duplication of publications in databases based on the application of the Levenstein algorithm for similarity assessment are formulated. It is proposed to transliterate the Cyrillic alphabet into the Latin alphabet to ensure clarity and correct comparison of textual data. A MySql database was developed to collect and update data on publishing activity. The title of the publication table of the database is supplemented with a special attribute, which stores the results of the conversion of Cyrillic names into corresponding Latin names. It is recommended to use indexing of database table fields (INDEX) by various attributes, which allowed to significantly increase the efficiency of searching, processing and comparing data. It is proposed to use the Soundex() function as a MySQL DBMS tool to determine the level of consonance of publication topics by additional parameters. The practical implementation of the algorithm for finding duplicate publications and their numbering confirmed the constructiveness of the proposed approach which was confirmed when filling the database. This article is of interest to software developers.

K e y w o r d s : ***APIs, ORCID, accounting of publications, Python, MatLab, algorithm, XML, duplicate search.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.