

## Автоматична категоризація науковців за тематикою досліджень на основі профілей в Google Scholar

Вінницький національний технічний університет

### *Анотація*

*У роботі представлено метод багатоярликової класифікації науковців на основі їх інтересів з профілей у Google Scholar. Метод реалізовано програмно. Застосовуючи розроблену інформаційну технологію створено прототип інформаційної системи з базою українських науковців.*

**Ключові слова:** категоризація, Google Scholar, база науковців.

### *Abstract*

*Method for multi label classification of scientists based on their preferences from google scholar is depicted. The method is programmed. By using developed information technology prototype of information technology with ukrainian scientists was created.*

**Key words:** multi label classification, Google Scholar, corpus of researchers.

### Вступ

На момент написання роботи не було виявлено систем автоматичної категоризації науковців за науковими інтересами. Це і стало першопричиною написання цієї роботи. У мережі присутній сайт української бібліометрики, в якому належність науковця до галузі визначалась за суб'єктивними враженнями. Тому метою даної роботи є розробка системи автоматизованої категоризації науковців за науковими інтересами.

### Опис методу

Розглядається задача багатоярликової класифікації або категоризації. Багатоярликовий класифікатор (multi-label classifier) функціонує таким чином, що кожен об'єкт може бути віднесено не одного класу, а до кількох. Наприклад, наукова стаття може бути віднесена до кількох наукових напрямків. Токсичний коментар в соціальній мережі може бути віднесено до таких 6 класів: отруйний, дуже отруйний, непристойний, загрозливий, образливий, із зневагою до ідентичності. Одночасно коментар може належати до кількох класів, наприклад, бути і непристойний, і загрозливим, і містити зневагу до ідентичності.

Вважатимемо відомими:

$W = (w_1, w_2, \dots, w_n)$  – перелік ключових слів та словосполучень тексту, яким відповідають інтереси науковця;

$T = (t_1, t_2, \dots, t_m)$  – перелік можливих класів (тем) – галузей знань відповідно до певної класифікації наук;

$D = (d_1, d_2, \dots, d_k)$  – множина розмічених текстів;

$R(D, T) \in D \times T$  – відношення, яке описує належність розміченого тексту до тем;  $R(d_j, t_p) = 1$ , якщо документ  $d_j$  віднесено до теми  $t_p$ ,  $j = \overline{1, k}$ ,  $p = \overline{1, m}$ .

Задача полягає у знаходженні тем з  $T$ , яким відповідає множина ключових слів  $W$ . При цьому вказується не лише сам факт належності, але і ступінь належності. Таким чином на виході отримуємо нечітку множину  $\tilde{W}$  на підмножині універсальної множини  $T$ .

Пропонований метод вирішення цієї задачі базується на таких принципах:

Принцип статистичної підтримки – чим більша частка документів з певної тематики містять аналізоване ключове слово, тим більша належність ключового слова до цієї тематики.

Принцип багатоярликості – ключове слово може належати до кількох тематик.

Принцип фільтрації шумів – ігнорується незначний ступінь належності ключового слова.

Принцип ігнорування стоп-слів – ігнорується ключове слово, яке зустрічається у дуже багатьох розмічених документах.

Принцип солідарності – чим більше ключових слів віднесено до однакової тематики, тим сильніша можливість віднесення документу до цієї тематики.

Наведені принципи пропонуються реалізувати наступним алгоритмом. В алгоритмі використовуються такі нові поняття:

$p$ -та тематична колекція – підмножина з  $D$ , елементи якої віднесено до теми  $t_p$ ,  $t_p p = \overline{1, m}$ ;

$N_p$  – обсяг (кількість документів)  $p$ -ї тематичної колекції.

< Розрахувати обсяги тематичних колекцій  $N_1, N_2, \dots, N_m$  >

<  $Counter=0$  – лічильник корисних ключових слів >

**for**  $i=1:n$

< Знайти  $Q$  – кількість документів з  $D$ , в яких є ключове слово  $w_i$  >

**If**  $Q > Threshold\_SW$  **continue**; % ідентифікація стоп-слів

**end**

**If**  $Q < Threshold\_Q$  **continue**; % ідентифікація шуму

**end**

< Знайти кількість  $t_1(w_1), t_2(w_2), \dots, t_m(w_i)$  документів тематичних колекцій з ключовим словом  $w_i$  >

**for**  $p=1:m$  % Занулюємо дуже малі видачі

**if**  $t_p(w_i) < Threshold\_topic$

$t_p(w_i) = 0$

**end**

**end**

**If**  $\max t_p(w_i) == 0$  **continue**

**end**

< Розраховуємо частоти входжень ключового слова  $w_i$  в тематичні колекції  $\gamma_p(w_i) = \frac{t_p(w_i)}{Q_p}$ ,  $p = \overline{1, m}$  >

< Відбираємо мінімальну кількість найпопулярніших тематик, сумарний внесок яких по  $\gamma$  не менше за  $Tail\_l$ .

Номера невідібраних тематик запишемо у вектор

*Rejected* >

< Занулюємо частоти входжень для непопулярних тематик:

```

     $\gamma_{Rejected}(w_i) = 0 >$ 
    < Нормуємо  $\gamma(w_i)$  на 100% >
    Counter=Counter+1
     $\mu_{counter} = \gamma(w_i)$ 
end
If Counter==0
Return ('Unsuccessful')
end
If Counter>1
 $\mu = mean(\mu_{counter})$ 
%усереднюємо
< Відбираємо мінімальну кількість найпопулярніших
тематик, сумарний внесок яких по  $\mu$  не менше за
Tail_2. Номера невідібраних тематик запишемо у вектор
Rejected >
< Занулюємо частоти входжень непопулярних тематик:
 $\mu_{Rejected}=0 >$ 
< Нормуємо  $\mu$  на 100% >
end
%Відрізаємо хвости – тематики з дуже низькою належністю:
Rejected = find( $\mu(:) < Tail\_3$ )
Rejected=find( $\mu(:)<Tail\_3$ )
 $\mu_{Rejected} = 0$ 
< Нормуємо  $\mu$  на 100% >
End

```

В деяких задачах відомо, що  $W$  мають відповідати не більше ніж  $Topic_{max}$  тем з  $T$ . Для цих випадків після головного циклу додаємо процедуру вилучення з розв'язку тем з рангами, більших за  $Topic_{max}$ .

### Результати роботи

Проілюструємо роботу методу на прикладі категоризації учасника наукової мережі Google Scholar – наукового керівника цього дослідження. Фрагмент профіля Штовби С.Д. наведено на рис.2. На ньому вказано наукові інтереси, які ототожнимо з ключовими словами, що характеризують діяльність науковця. Слова, якими в профілі Google Scholar описано інтереси, науковці обирають на власний розсуд, без прив'язки до якогось словника. З іншого боку є класифікація наук, з прийнятими назвами тематик. Відповідно, ключові слова з профілю можна ототожнити з  $W$ , а галузі наук з деякою формальною класифікації через  $T$ . Тоді, виникає задача співставлення  $W$  з елементами множини  $T$ . Саме для вирішення цієї задачі і розроблено метод у попередньому пункті.



	Все	Начиная с 2012 г.
Статистика цитирования	2926	1893
h-индекс	18	15
i10-индекс	34	23

Рисунок 1 – Профіль науковця в Google Scholar

Як систему класифікації оберемо наукові галузі зі Скопусу. Уся наука в Скопусі розбита на 27 галузей. В Скопусі накопичено величезних обсяг наукових публікацій, кожна з яких категоризована за цими галузями. Крім того, Скопус видає статистику – скільки документів за кожною галуззю містить пошуковий запит. Тобто, усі процедури з отримання зовнішньої інформації для категоризації науковця можна реалізувати.

Перекладемо інтереси на англійську і отримаємо такі ключові слова:  $W=\{\text{fuzzy logic; data science; information technologies; artificial intelligence}\}$ . Здійнемо категоризацію за таких параметрів алгоритму:  $\text{threshold}_{sw}=15\%$ ,  $\text{threshold}_Q=500$ ,  $\text{threshold}_{topic}=100$ ,  $\text{Tail}_1=50\%$ ,  $\text{Tail}_2=75\%$ ,  $\text{Tail}_3=10\%$ . Пошук здійснимо за документами не старших за 10 років; зоною пошуку оберемо назву публікації, її ключові слова та анотацію. Результати пошуку наведено в табл. 1.

Усі ключові слова пройшли допуск на стоп-слова ( $\text{threshold}_{sw}$ ) та достовірність статистики ( $\text{threshold}_Q$ ). За деякими галузями видача за запитом була менше допустимою, тому її вважаємо за шум. Після видалення шумів належності до галузей кожного запиту показано на рис. 2. Подальші перетворення ілюструє рис. 3. Після відсікання хвостів за кожним запитом залишилася від 2 до 4 галузей. Після узагальнення отримаємо 5 галузей, 3 з яких утворюють хвіст розподілу – потрапляють у 4 кварталі. Після відсікання хвоста розподілу та нормування отримаємо таку нечітку класифікацію: належність до галузі Computer Science – 49.8%, належність до галузі Decision Sciences – 50.2%. Такий результат автоматичної категоризації відповідає експертним представленням про дослідження цього науковця.

Таблиця 1 – Статистика видачі Скопуса

№	Галузь Скопуса	Запит				
		Пустий	Fuzzy logic	Data science	Information technologies	Artificial intelligence
1	Agricultural and Biological Sciences	2086210	954	34	1082	1953
2	Arts and Humanities	1252662	217	27	1675	2137
3	Biochemistry Genetics and Molecular Biology	3205353	1357	89	1791	6337
4	Business Management and Accounting	763887	1186	81	11485	4784
5	Chemical Engineering	1195116	990	19	787	1446
6	Chemistry	2254800	481	29	512	1636
7	Computer Science	3327312	27103	1036	69597	145913
8	Decision Sciences	295180	1715	145	8625	6285
9	Dentistry	168773	10	1	79	19
10	Earth and Planetary Sciences	1192912	1356	32	1261	2256
11	Economics Econometrics and Finance	529007	299	20	2532	2296
12	Energy	881518	3089	20	1243	2954
13	Engineering	5556672	23434	254	42684	44292
14	Environmental Science	1373065	1960	40	2221	2979
15	Health Professions	338817	312	17	2093	1766
16	Immunology and Microbiology	764581	135	11	231	332

17	Materials Science	2679940	1708	59	2084	2977
18	Mathematics	1819513	10424	309	6246	50349
19	Medicine	7986846	2076	186	11736	10943
20	Multidisciplinary	311745	548	29	689	770
21	Neuroscience	681608	471	35	305	2665
22	Nursing	485882	35	23	1371	129
23	Pharmacology Toxicology and Pharmaceutics	911035	177	30	1245	511
24	Physics and Astronomy	3211392	1861	78	1816	4384
25	Psychology	636301	93	22	1177	846
26	Social Sciences	2451088	1637	193	15383	6081
27	Veterinary	241865	19	3	36	34
Total		28231391	46813	1675	125233	182109

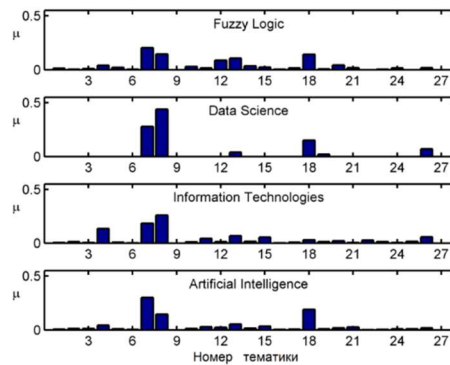


Рисунок 2 – Первина обробка видачі Скопуса

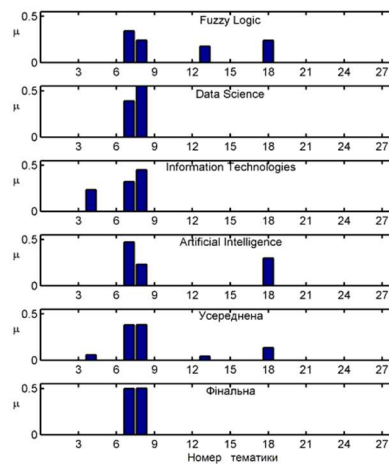


Рисунок 3 – Нечітка категоризація науковця за ключовими словами

Програмна реалізація та експериментальні дослідження

Експериментальні дослідження проведемо на базі профілей українських науковців в науковій мережі Google Scholar. Задача полягає у категоризації їх за деякими системи класифікації наук. Це розширений варіант контрольного прикладу з попереднього пункту.

Як інформаційною базою українських науковців скористуємося ресурсом «Бібліометрика української науки», на якому наведено адреси профілів біля 25000 українських науковців. Для створення відповідної бази нами розроблено парсер для цього сайту. Далі за адресами цих науковців потрібно отримати з Google Scholar інформацію про наукові інтереси цих науковців. Для цього також розроблено відповідний парсер. Таким чином, ми сформували базу ключових слів, необхідних для категоризації науковців за запропонованим алгоритмом.

Базою розмічених документів можуть бути інформаційні системи Springer та Скопус. Саме ці системи надають статистику за науковими галузями. Але ці системи наповнено переважно англійськими джерелами, тому ключові слова слід перекласти на англійську. Для цього написано відповідний парсер веб-перекладача Microsoft, який є у відкритому доступі.

Система класифікації наукових досліджень у Скопусі містить 27 галузей. Виокремлення деяких великих галузей є сумнівним, і суперечить іншим класифікаціям, наприклад, класифікації видавництва Wiley. Для підвищення презентабельності результатів та збалансування наповненості галузей ми зробили такі об'єднання:

Chemistry та Chemical Engineering;

Health Professions та Nursing;

Medicine та Dentistry.

В Шпрингері наукові галузі погано впорядковані – містять як синоніми, так і занадто дрібні екземпляри. Більш цікавим є підгалузі (subdisciplines) Шпрингера, які дозволяють достатньо вузько категоризувати дослідження. Усього таких підгалузей майже 1600. Як і у випадку галузей, деякі підгалузі мають крихітне наповнення. Є також і синонімія. Нами встановлено, що 95% наповнення припадає на 595 підгалузей. Саме це ядро підгалузей обрано нами для опрацювання. В цьому ядрі поєднано близькі підгалузі та зроблено їх категоризацію на об'єднані галузі Скопуса. Це процес відбувався в напівавтоматичному режимі. Для прикладу наведено категоризацію підгалузей, які потрапили до галузі Computer Science:

- Algorithm Analysis Problem Complexity

Complexity

Algorithms;

- Artificial Intelligence / Robotics

- Biometrics

- Mathematical Biology / Computational Biology

Bioinformatics

Computational Biology / Bioinformatics

- Computational Intelligence

- Computational Linguistics

- Computer Application / Chemistry

- Computer Hardware

- Computer Science

Computational Science and Computational Engineering

Computing Methodologies

- Computers in Education

- Data Mining / Knowledge Discovery

- Data Structures

- Geographical Information Systems / Cartography

- Information technology in Business

- Legal Aspects of Computing

- Management of Computing / Information Systems

- Multimedia / Information Systems

- Ontology

- Operating Systems

- Processor Architectures

- Statistics Programs

- System Performance / System Evaluation

- Simulation / Modeling

- User Interfaces / Human Computer Interaction

- Quantum Computing, Information and Physics
- Popular Computer Science
- Mathematical Logic / Formal Languages
- Logics / Meanings of Programs
- Numeric Computing
- Mathematics of Computing
- Computation on Abstract Devices
- Theory of Computation
- Software Engineering та Software Engineering / Programming / Operating Systems
- Data Structures / Cryptology / Information Theory
- Systems Security / Data Security
- Coding Theory / Information Theory та Data Encryption
- Pattern Recognition та Signal Processing / Image Processing / Speech Processing
- Programming Languages, Compilers, Interpreters
- Programming Techniques
- Computer Applications та Information Systems Applications / Internet
- Information Systems, Communication Service
- Special Purpose Systems, Application-Based Systems
- Computer System Implementation
- Computer Graphics
- Computer Imaging / Computer Vision / Pattern Recognition / Computer Graphics
- Image Processing / Computer Vision
- Computer Communication Networks
- Computer Systems Organization / Communication Networks
- Computer Engineering та Electronic Engineering / Computer Engineering
- Document Preparation / Text Processing
- Information Storage / Information Retrieval
- Computer-Aided Engineering.

Деякі з наведених вище підгалузей окрім Computer Science увійшли і у інші, наприклад, Computer Application in Chemistry увійшла до Chemistry та Chemical Engineering, а Algorithm Analysis Problem Complexity, та Complexity, та Algorithms – до Mathematics тощо.

### **Розробка програмного забезпечення багатоярликової класифікації**

У якості перевірки роботи методу багатоярликової класифікації було розроблено програмне забезпечення класифікації українських науковців до галузей Скопусу та піддисциплін Шпрингеру з сайту української бібліометрики за посиланням: <http://nbuviap.gov.ua/bpnu/index.php>. Програмне забезпечення написано на мові програмування PHP. Для збереження даних про науковців було використано систему управління базами даних MySQL.

Для роботи зі Скопусом та Шпрингером розроблено відповідні парсери. Модуль парсингу складається з декількох ієрархій класів. Класи парсингу описуються інтерфейсом CommonParser (рис. 4).

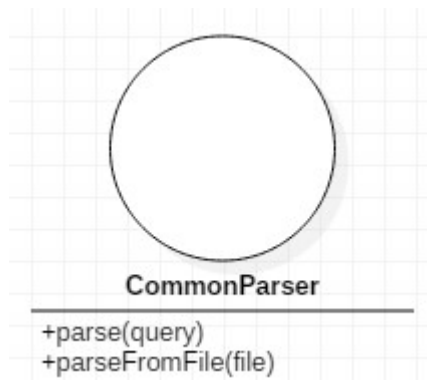


Рисунок 4 – Інтерфейс, що описує об’єкт «Парсер»

Інтерфейс описує операції, які може виконувати конкретний парсер. Всього два парсери. Алгоритм парсингу спільний для багатьох сайтів. Тому конкретні парсери наслідуються від абстрактного класу `AbstractParser`, який реалізує інтерфейс `CommonParser` (рис. 5).

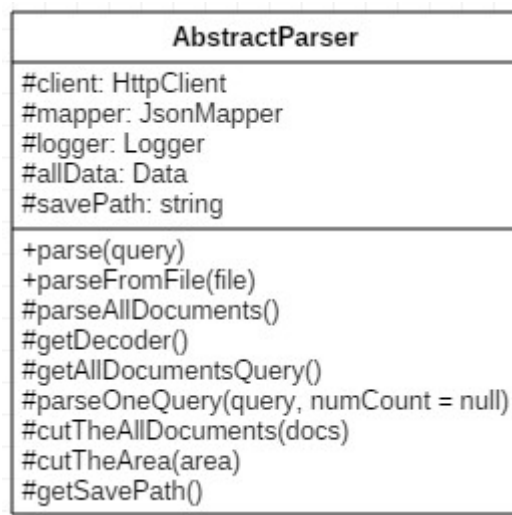


Рисунок 5 – Базовий парсер, що реалізує алгоритм та делегує конкретні завдання своїм нащадкам

В вказаному вище даному класі описується алгоритм парсингу. Алгоритм є спільним для деяких парсерів. Це так званий шаблон ООП (об’єктно-орієнтованого програмування) «`Template method`». Він реалізує операції `CommonParser` використовуючи абстрактні захищені методи показані вище. Будь-який парсер повинен реалізувати ці методи. Для отримання html сторінки використовується `Http` клієнт `Guzzle` який обгорнений у клас `HttpClient`. Для конвертування `json` рядку в об’єкт використовується `JsonMapper`. Для збереження логів використовується `Logger`. Реалізації `CommonParser` для Скопусу – зображена на рис. 6, а для Шпрингера на рис. 7.



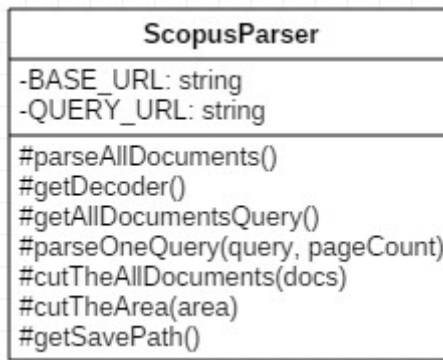


Рисунок 6 – Парсер сайту Scopus

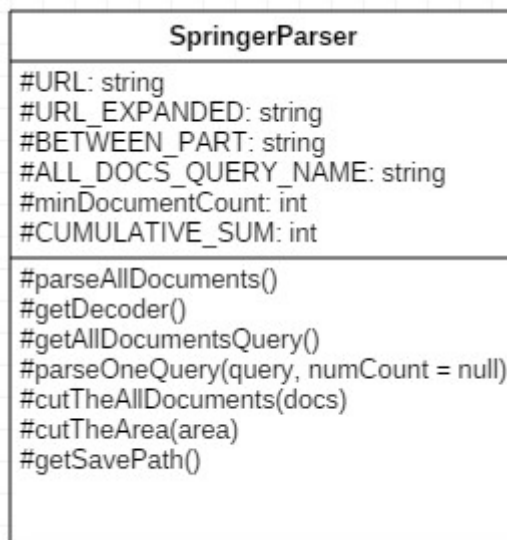


Рисунок 7 – Парсер сайту Springer

Інша ієрархія класів описує те, що повертається внаслідок парсингу. Ієрархію класів описує інтерфейс Data (рис. 8).

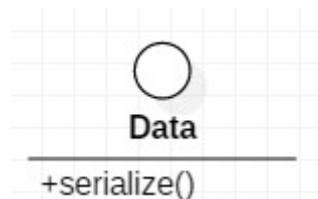


Рисунок 8 – Інтерфейс, що описує базовий тип даних, що повертається після парсингу

Після виконання парсингу конкретний парсер обгортає дані у відповідну реалізацію інтерфейсу Data (рис. 9 та 10). Загальна діаграма класів модулю парсингу наведена на рис. 11.

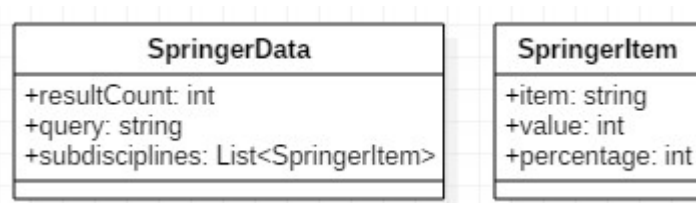


Рисунок 9 – Обгортка даних, що повертаються після парсингу об’єктом SpringerParser

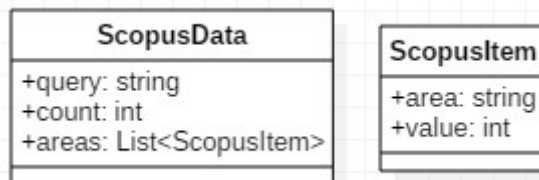


Рисунок 10 – Обгортка даних, що повертаються після парсингу об’єктом ScopusParser

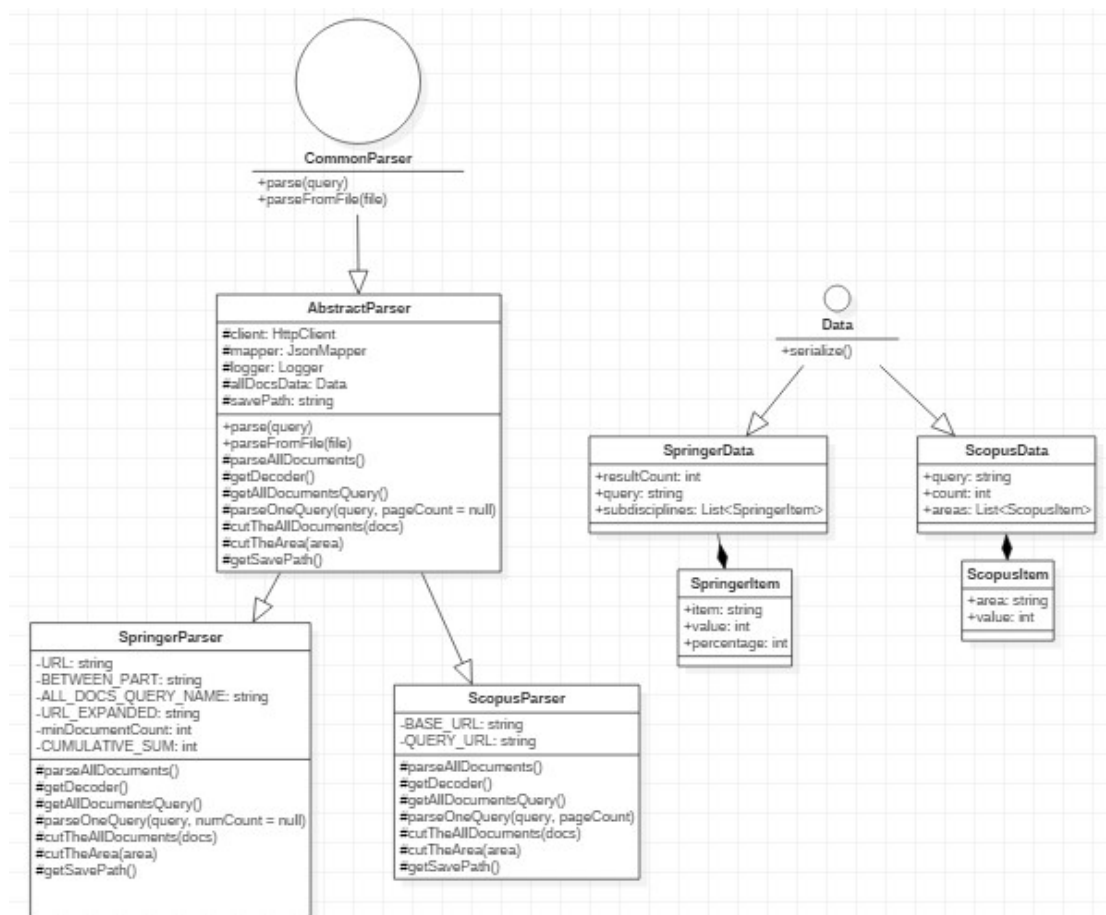


Рисунок 11– Діаграма класів модуля парсингу

Окрім модуля парсингу реалізовано модуль алгоритму, який опрацьовує список отриманої кількості документів за кожним інтересом науковця за алгоритмом з розділу 1. Реалізовано два

конкретних алгоритми: алгоритм опрацювання результатів зі Скопусу та Шпрингера. Вони мають спільні риси тому під час їх реалізації повторювані методи були винесені у абстрактний клас. Діаграма класів подана на рисунку 12. В даному модулі алгоритм описується інтерфейсом Algorithm з методом getFieldNameValue(data), що приймає параметр data з типом Data з попередньої діаграми. Для зручності увесь список піддисциплін Шпрингера та галузей Скопуса конвертується за допомогою методу prepareItems(items) у асоціативний масив, де ключем є назва галузі/піддисципліни а значенням – відповідна кількість документів.

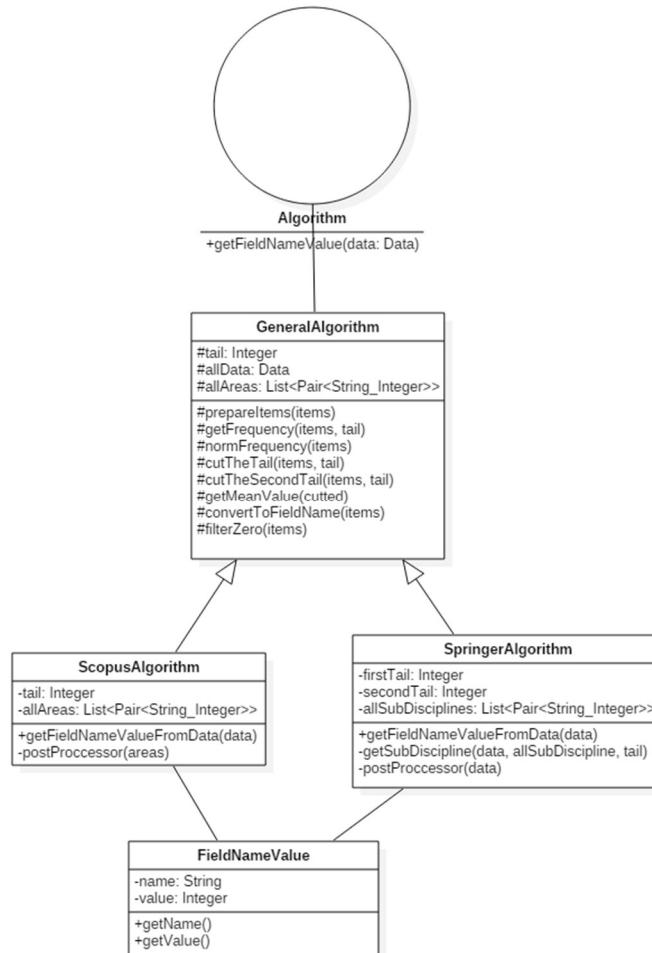


Рисунок 12 – Діаграма класів модулю алгоритму

В даній роботі ми працюємо з профілями науковців з Google Scholar. Структура такого науковця має вигляд поданий на рисунку 13.

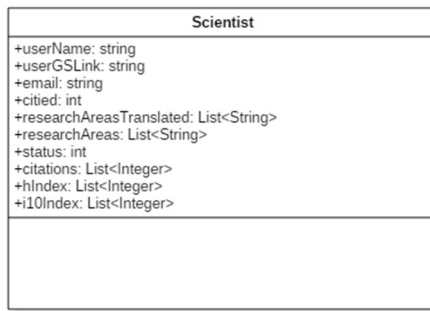


Рисунок 13 – Структура профіля науковця з Google Scholar

Модуль алгоритму працює лише з статистичними даними, тому для отримання галузей/піддисциплін по науковцю розроблено модуль класифікації вчених за науковими інтересами, що використовує модуль алгоритму описаний вище. Діаграма класів даного модулю має вигляд поданий на рисунку 14. Даний модуль не оновлює науковців з їх галузями/піддисциплінами в базі даних а лише повертає список галузей/піддисциплін характерних для даного науковця для подальшої обробки.

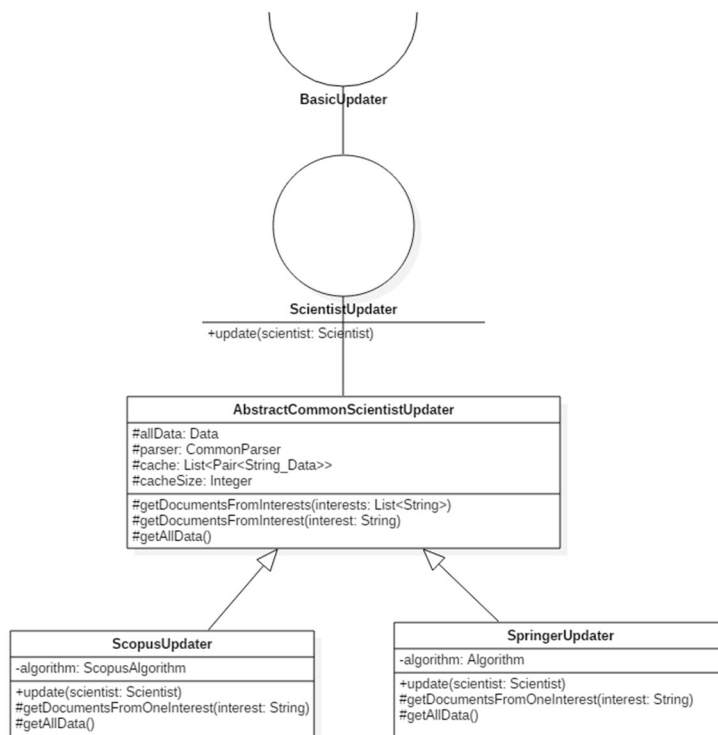


Рисунок 14 – Діаграма класів модулю категоризації вчених

Окрім цих модулів було розроблено Web інтерфейс для роботи з системою. Даний інтерфейс надає інформацію про усіх категоризованих науковців та дозволяє сортувати їх за різними критеріями. Інтерфейс поки що не дає прямого доступу до API модулів категоризації описаних вище. Web інтерфейс побудований на технології jQuery з використанням аjax для відправки асинхронних

запитів до серверу. Сервер на деякі запити генерує html сторінки а на деякі повертає json у відповідь з інформацією про науковців. Для обробки запитів на сервері було розроблено модуль маршрутизації отриманих запитів. За своєю суттю він нагадує шаблон проектування «Ланцюжок обов’язків». Діаграма класів цього модулю подана на рисунку 15. Модуль складається з декількох сутностей: Action (обробник), Router (маршрутизатор), ActionFactory (фабрика обробників). Будь-який запит потрапляє до маршрутизатора FirstRouter, який містить у собі (через батька) фабрику RouterFactory з обробниками на запит з ключем маршрутизації / та інші маршрутизатори, наприклад, AlgorithmRouter. Під ключем маршрутизації можна розуміти шлях до ресурсу в домені. Наприклад, у домені <http://localhost/profiles.com/algorithm> ключем маршрутизації є algorithm, відповідно маршрутизатор, який має такий ключ може перенаправляти будь-який наступний запит: [http://localhost/profiles.com/algorithm/\\*](http://localhost/profiles.com/algorithm/*). Усю логіку запиту виконують обробники. На діаграмі показано лише три обробники, які виконують запити домашньої сторінки (HomeAction), оновлення галузей скопусу (UpdateScientisScopusAction) та шпрингера (UpdateScientistSpringerAction) для науковців.

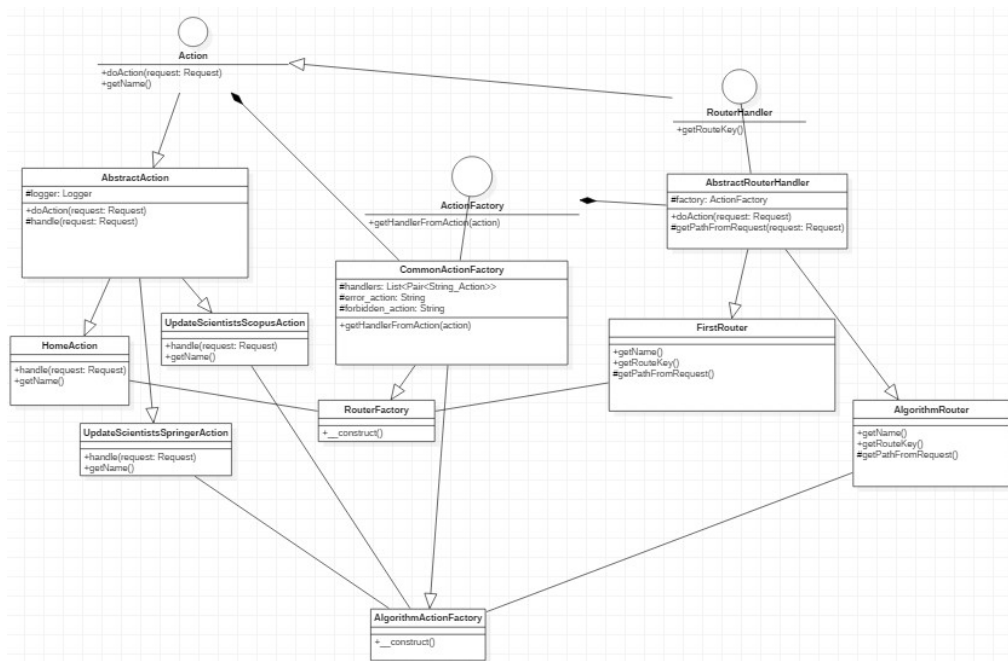


Рисунок 15 – Частина модулю маршрутизації запитів

Результати автоматичної категоризації науковців добре збігаються з експертним уявленням про напрямок наукових досліджень (рис. 16). Результати категоризації представлено на сайті <http://shtovba.vk.vntu.edu.ua/dev> з можливістю рейтингування науковців за цитованістю на множині підгалузей або галузей (рис. 17, 18).

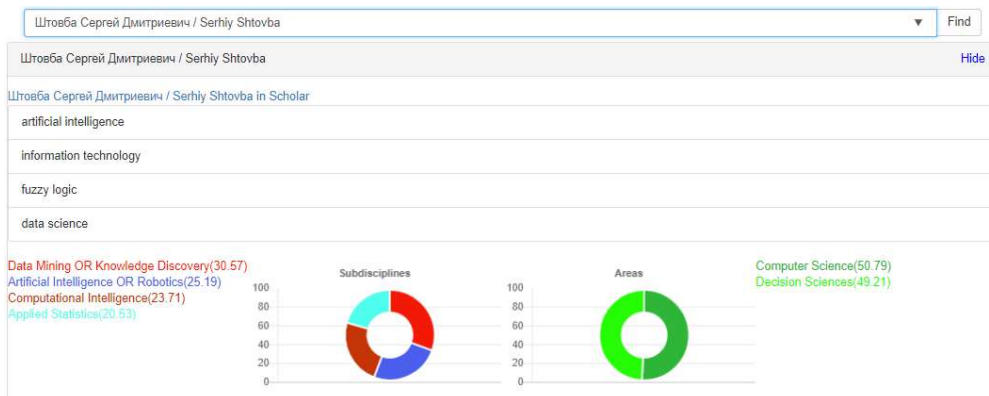


Рисунок 16 - Результати категоризації науковця за галузями та піддисциплінами з контрольного прикладу

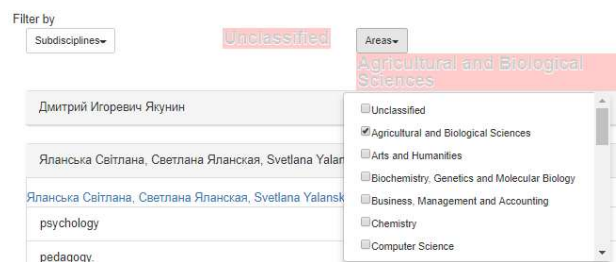


Рисунок 17 – Рейтингування науковців за галузями та піддисциплінами

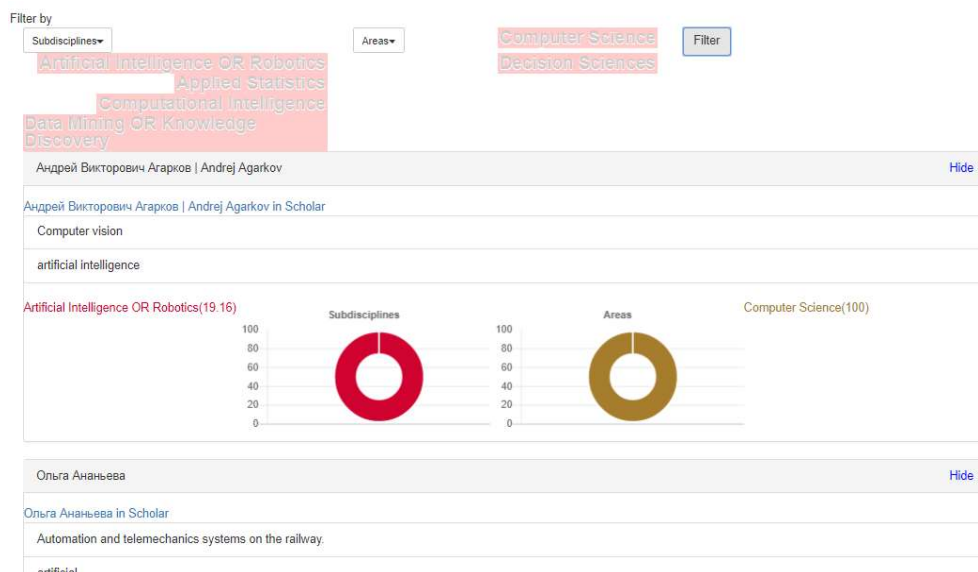


Рисунок 18 – Фільтр за вибраними галузями та піддисциплінами

В системі реалізовано пошук науковців за ім'ям з Google Scholar. При введенні частини прізвища або ім'я у полі введення з'являється список усіх науковців, які мають спільні частини з введеними (рис. 19).

Володимир | Find

Volodymyr Yagetsenko/ Яременко Володимир Васильович  
 Володимир Ятченко  
 Володимир Адаменко / Adamenko V  
 Володимир Андрієць / Volodymyr Andriets / orcid.org/0000-0003-3671-0816

Рисунок 19 – Пошук науковця за ім'ям

Деякі приклади класифікації науковців подано на рисунках 20-28.

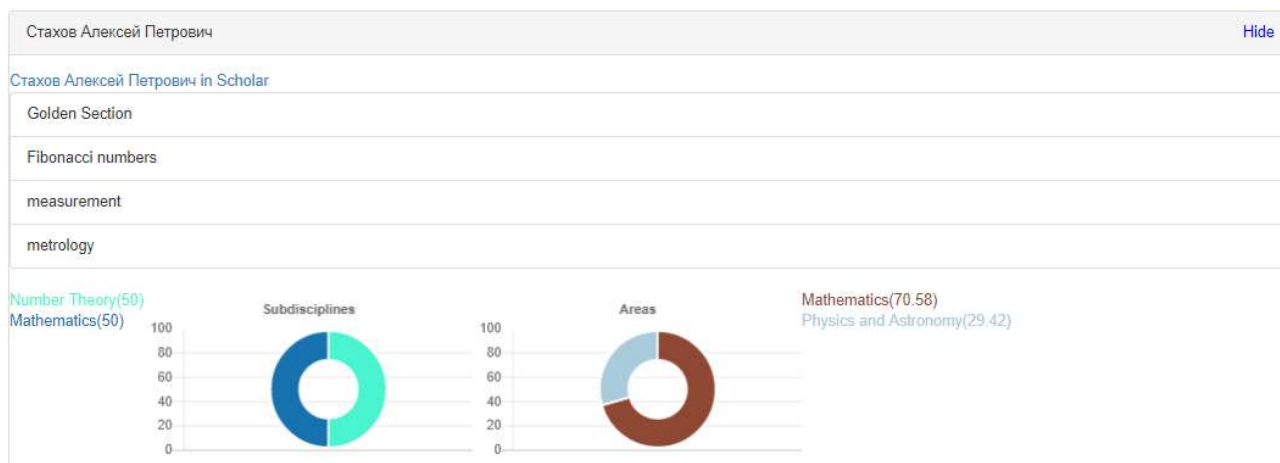


Рисунок 20 – Класифікація науковця за галузями

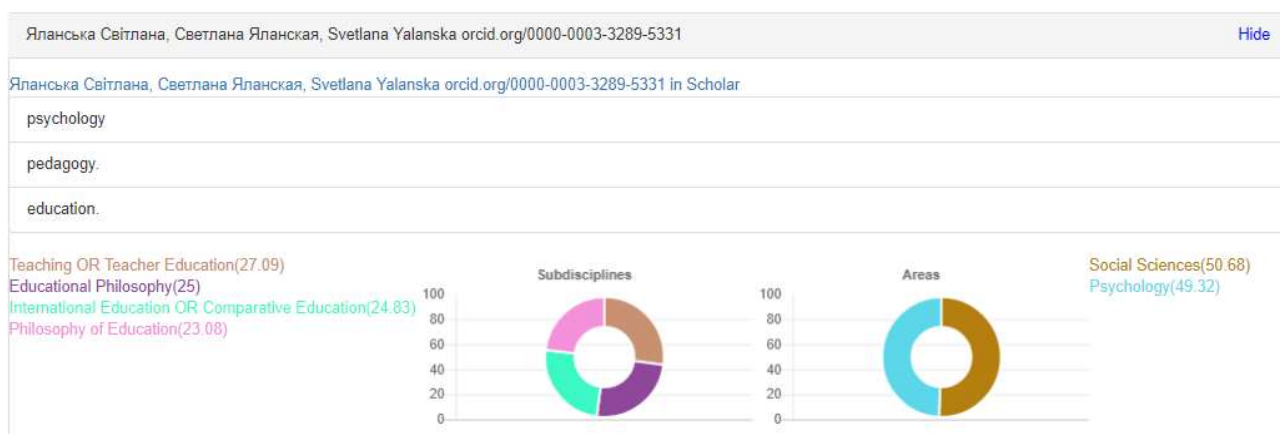


Рисунок 21 – Класифікація науковця за галузями



Рисунок 22 – Класифікація науковця за галузями

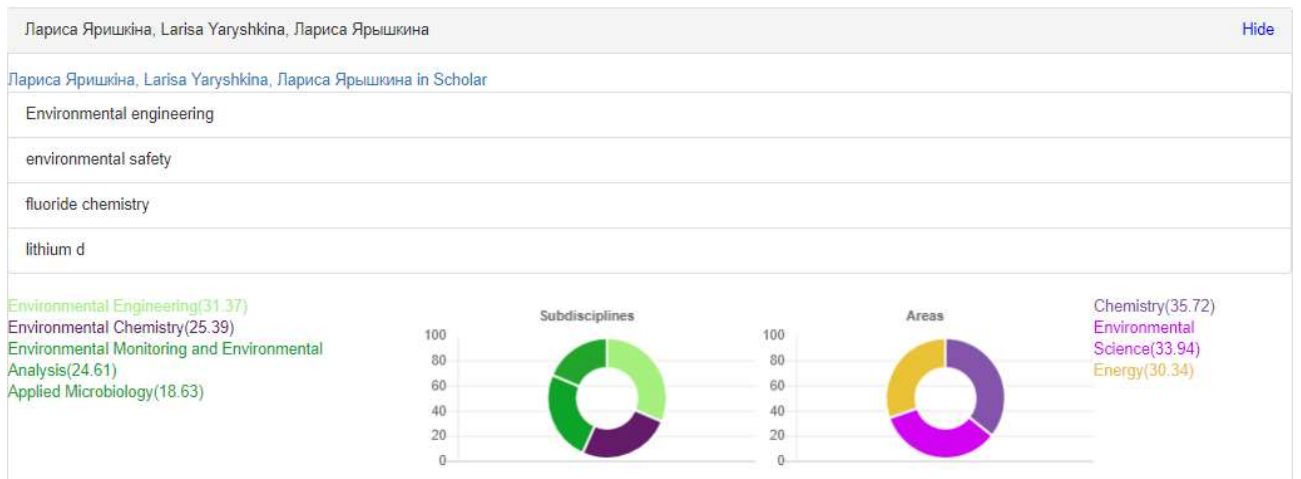


Рисунок 23 – Класифікація науковця за галузями

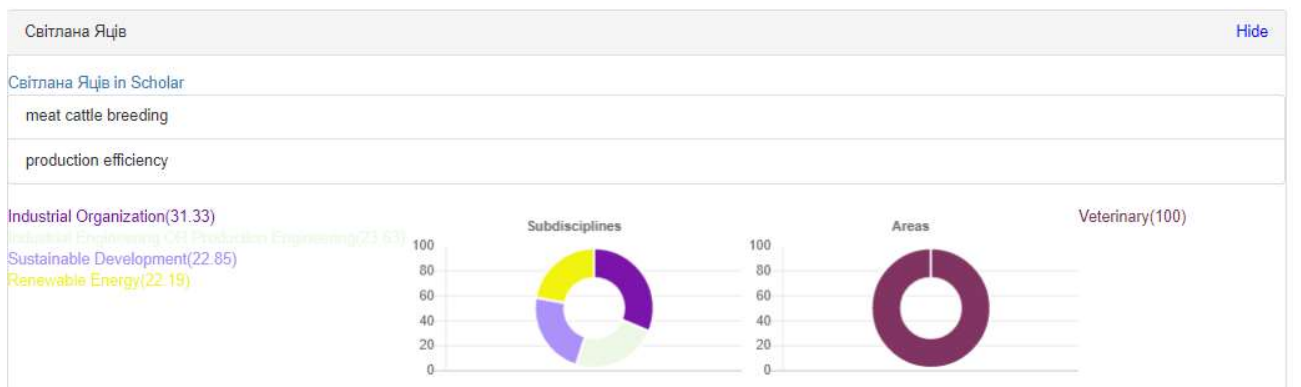


Рисунок 24 – Класифікація науковця за галузями

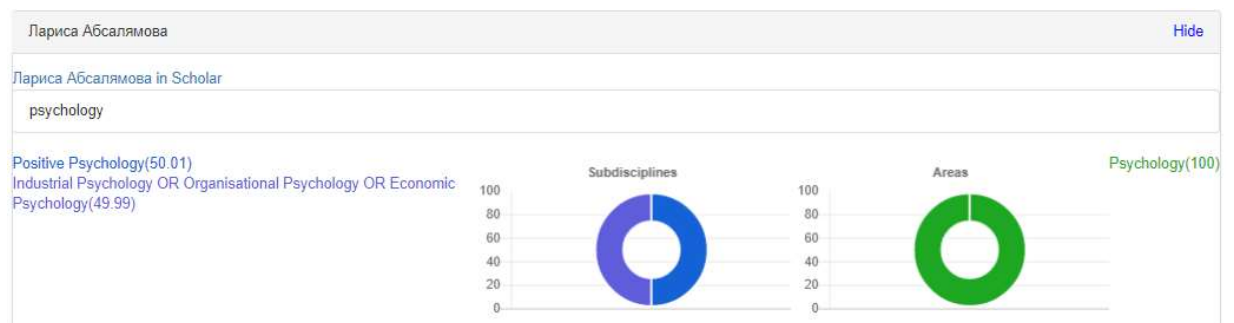


Рисунок 25 – Класифікація науковця за галузями

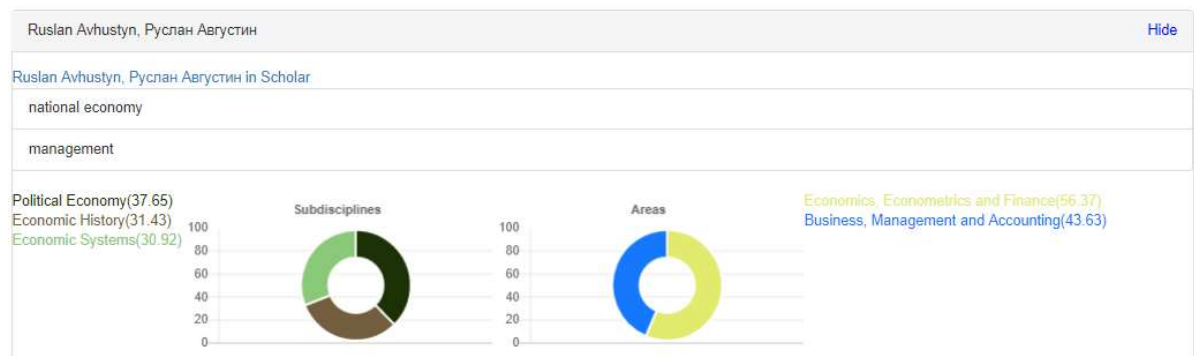


Рисунок 26 – Класифікація науковця за галузями



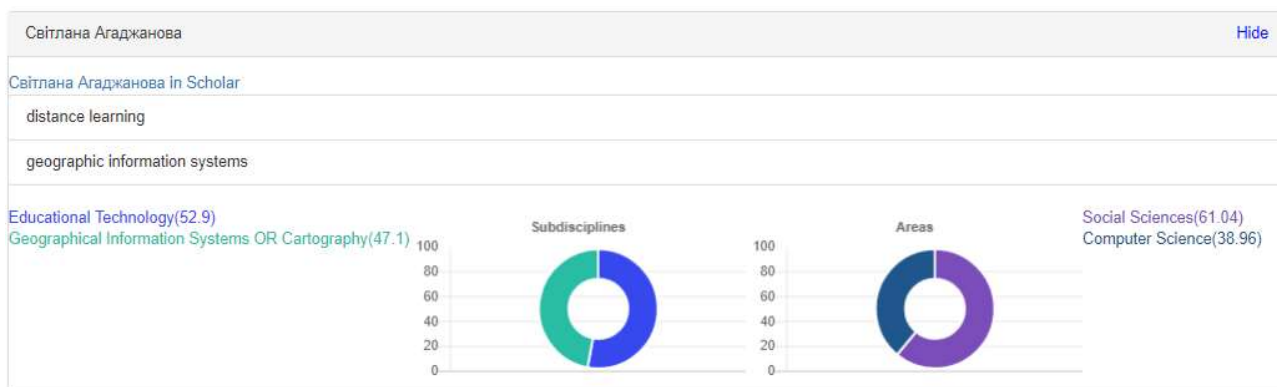


Рисунок 27 – Класифікація науковця за галузями

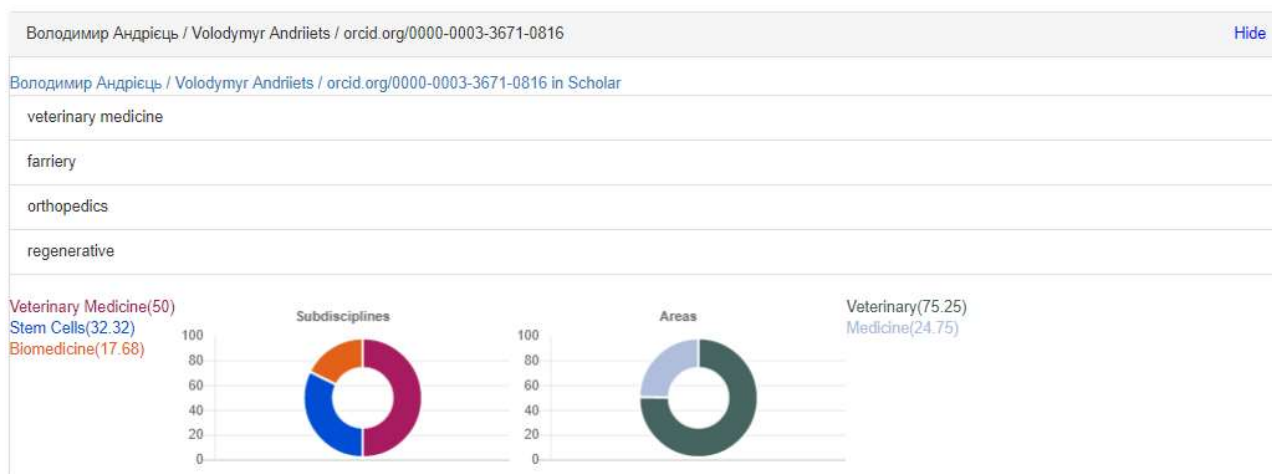


Рисунок 28 – Класифікація науковця за галузями

### Висновки

В порівнянні з бібліометрикою української науки класифікація науковців розробленим методом науково здійснюється автоматично. Окрім цього при аналізі діяльності науковців враховуватиметься не одна галузь (піддисципліна) а декілька, що дозволить більш детальніше визначити напрямки роботи науковців.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Zhang M. L., A review on multi-label learning algorithms / M.L. Zhang, Z. H. Zhou // IEEE transactions on knowledge and data engineering. – 2014. – Vol. 26. – №.8. – P. 1819-1837.
2. Городецкий В. И. Методы и алгоритмы коллективного распознавания / Городецкий В.И., Серебряков С.В. // Автоматика и телемеханика. – 2008. – №11. – С. 3-40.
3. Kuncheva L. Combining pattern classifiers: methods and algorithms / L. Kuncheva. – John Wiley & Sons, 2004. – 350 p.
4. К.В. Воронцов Модификации EM-алгоритма для вероятностного тематического моделирования / К.В. Воронцов, А.А. Потапенко – Машинное обучение и анализ данных, 2013 – с.657-686.

5. К.В. Воронцов Вероятностное тематическое моделирование / К.В. Воронцов – 2013.

**Сергій Дмитрович Штовба** – д.т.н., професор кафедри комп'ютерних систем управління, Вінницький національний технічний університет, м.Вінниця, e-mail: [shtovba@vntu.edu.ua](mailto:shtovba@vntu.edu.ua).

**Микола Володимирович Петричко** – ст.гр. 1АВ-14б, факультету комп'ютерних систем та автоматики Вінницького національного технічного університету, м. Вінниця , e-mail: [petrychko.myckola@gmail.com](mailto:petrychko.myckola@gmail.com).

**Shtovba Serhiy** —Professor of Department of Computer Control Systems, Vinnytsia National Technical University, Vinnytsia, e-mail: [shtovba@vntu.edu.ua](mailto:shtovba@vntu.edu.ua).

**Petrychko Mykola** — student, Facultyt of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia, email : [petrychko.myckola@gmail.com](mailto:petrychko.myckola@gmail.com).