

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Автоматизована інформаційна система тематичної класифікації наукових статей

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студентка 4 курсу, група КН-17-2
Курс, група виконавця
Підпис А.В. Шпичко
Ініціали, прізвище
Керівник: к.т.н., доцент кафедри КНІТ
Науковий ступінь, посада
Підпис О.В. Мазурець
Ініціали, прізвище
Нормоконтроль: к.т.н., доцент кафедри КНІТ
Науковий ступінь, посада
Підпис Р.О. Багрій
Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КНІТ, д.т.н., професор

08 червня 2021 р.

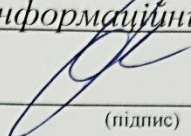
Підпис

О.В. Бармак

Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра комп'ютерних наук та інформаційних технологій
Освітній ступінь бакалавр
Галузь знань 12 – Інформаційні технології
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук та інформаційних технологій


(підпис)
д.т.н., професор О.В. Бармак
« 08 » лютого 2021 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Автоматизована інформаційна система тематичної класифікації наукових статей»

2. Завдання видано студентці Шпичко Анастасії Валентинівні
(прізвище, ім'я, по батькові)

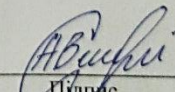
3. Керівник роботи доцент кафедри КНІТ Мазурець Олександр Вікторович
(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від « 05 » од 2021 р. № 11

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

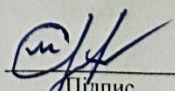
Мета роботи – розробка інформаційної системи тематичної класифікації наукових статей. Слід забезпечити виконання основних функцій системи: вибір кількості класів для класифікації, вибір кількості ключових слів у класах, обробка тексту статей, формування послідовних множин та унікальних множин слів статей, обрахунок значень TF-IDF унікальних слів класів, пошук збігів тестової статті за ключовими словами класів, обрахунок оцінки приналежності тестової статті до класів, візуалізація результатів роботи системи.

Виконавець: студентка 4 курсу, група КН-17-2
Курс, група виконавця


Підпис

А.В. Шпичко
Ініціали, прізвище

Керівник: к.т.н., доцент кафедри КНІТ
Науковий ступінь, посада


Підпис

О.В. Мазурець
Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: «Автоматизована інформаційна система тематичної класифікації наукових статей»

Виконавець кваліфікаційної роботи бакалавра: студентка групи КН-17-2 Шпичко Анастасія Валентинівна

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КНІТ Мазурець Олександр Вікторович

Кваліфікаційна робота бакалавра містить:

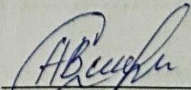
Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
55	30	10	43	4

Метою кваліфікаційної роботи бакалавра є створення автоматизованої інформаційної системи тематичної класифікації наукових статей.

Результатом виконання кваліфікаційної роботи є система, що дозволяє спростити та автоматизувати визначення тематичної класифікації для обраної наукової статті. Для розробки програмного продукту було обрано СКБД SQL Server 2012, мова програмування C# та фреймворк .NET. Вхідними даними автоматизованої інформаційної системи тематичної класифікації наукових статей є: параметри роботи (кількість класів для класифікації та кількість ключових слів у класах), множина статей для кожного з класів та тестова стаття для аналізу. Вихідними даними є: числова оцінка приналежності обраної статті до класів та таблиця збігів тестової статті за ключовими словами класів.

Ключові слова: наукова стаття, класифікація, інформаційна система, стаття, TF, IDF.

Виконавець: студентка 4 курсу, група КН-17-2
Курс, група виконавця


Підпис

А.В. Шпичко
Ініціали, прізвище

Зміст

Перелік скорочень	3
Вступ.....	4
Розділ 1	
Характеристика предметної області та постановка задачі	6
1.1 Аналіз предметної області	6
1.2 Аналіз існуючого програмного забезпечення предметної області	11
1.3 Аналіз сучасних засобів створення програмного забезпечення	15
1.4 Постановка задачі та вимоги до розробки інформаційної системи.....	17
Розділ 2	
Проектування інформаційної системи	18
2.1 Моделі, методи, інформаційна технологія системи	18
2.1.1 Схема інформаційної технології	18
2.1.2 Метод пошуку ключових слів	20
2.2 Інформаційна структура системи	22
2.2.1 Структура компонентів інформаційної системи	22
2.2.2 Структура бази даних інформаційної системи	24
2.3 Вибір засобів розробки інформаційної системи	28
2.3.1 Вибір мови програмування	28
2.3.2 Вибір фреймворку	29
2.3.3 Вибір СКБД	30
Розділ 3	
Програмна реалізація інформаційної системи	33
3.1 Структура та функціональне призначення складових системи	33
3.2 Особливості реалізації складових системи	35
3.3 Тестування інформаційної системи	38
3.4 Інструкція користувача.....	43
3.5 Вимоги до розгортання інформаційної системи.....	49
Висновки	50
Перелік посилань.....	52
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
БД	База даних
ІС	Інформаційна система
ІТ	Інформаційні технології
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
КНІТ	Комп'ютерні науки та інформаційні технології
ПІБ	Прізвище ім'я по-батькові
ПЗ	Пояснювальна записка
ПП	Програмний продукт
СКБД	Система керування базами даних
ХНУ	Хмельницький національний університет.
MS	Microsoft
SQL	Structured Query Language
TF	Term Frequency
TF-IDF	Term Frequency – Inverse Document Frequency

Вступ

Наукові дослідження у вигляді статей відіграють важливу роль у формуванні майбутніх спеціалістів. Завдяки дослідженням фахівці отримують досвід та підвищують свої шанси на працевлаштування у тій чи іншій сфері.

Результатом процесу мислення, в якому поєднуються певний аналіз, структурування, формулювання та висловлення думок є стаття. Вона висвітлюється як науковий або публіцистичний твір невеликого розміру у збірнику, журналі чи газеті [1]. Отже, стаття слугує візуальним вираженням результатів мисленнєвої діяльності. Саме наукова стаття є результатом наукового дослідження.

Кожна наукова стаття має відноситись до певної науки, тобто повинна мати свою класифікацію. Для більш зручної та чіткої класифікації використовуються ключові слова кожної статті. До ключових слів відносять окремі слова чи словосполучення, які визначають тему і зміст статті [2].

За допомогою ІТ, пошук цих слів відбувається автоматично, що значно економить час та відносить статтю до правильної класифікації. На сьогодні можна виділити такі основні класифікації: УДК, класифікація за галуззю знань МОН, сучасна класифікація наук та класифікація по 5 напрямкам.

Кожна з класифікацій має свій перелік класів за якими відбувається класифікація статей, тому, якщо взяти одну і ту ж статтю та провести класифікацію різними способами, результат буде у вигляді різних але суміжних класів класифікації.

Досить визначальну роль у класифікації відіграє тематика статті, оскільки, в темі статті дуже часто зустрічаються слова, які є ключовими у тому чи іншому класі. Завдяки чому, стаття вже на початковому етапі може отримати клас класифікації.

Метою класифікації можна назвати створення для всіх запропонованих наукових статей значення за яким вона буде розпізнаватись при пошуку статті в бібліотеках, електронних ресурсах та ін.

Під час класифікації, в статті визначаються ключові слова, які в подальшому використовуються для співставлення їх із ключовими словами статей, які вже відносяться до певних класів класифікації. Тобто, якщо більшість ключових слів нової статті підходять до ключових слів статей з класифікацією, то нова стаття автоматично стає статтею відповідної їй класифікації. Таким чином, у даній сфері класифікації, використовуючи ІТ можна швидко віднести будь-яку статтю до відповідного їй класу класифікації.

Розділ 1

Характеристика предметної області та постановка задачі

1.1 Аналіз предметної області

Завдяки наукометричним базам можемо помітити, що кожного року наукові статті збільшуються в кількості, це дає поштовх до розробки певних автоматизованих систем для їх класифікації. Наукометрична база містить певні проаналізовані ресурси, придатні для моніторингу, організації та адміністрування науки [3]. Для написання низки наукових публікацій необхідно мати певний перелік книг та статей, за допомогою яких буде проводитись аналіз. До переліку книг, журналів і статей зазначається місце, рік, видавництво та інші данні про публікацію [4]. Таким переліком прийнято називати бібліографію.

Бібліографія є галуззю знань про методи опису друкованих та електронних видань, складання їх покажчиків і оглядів для наукового і практичного використання [5]. У періодичних виданнях, бібліографія слугує відділом присвяченим короткому огляду книг, що вийшли з друку [4].

Велике значення має стандартизація бібліографічних записів документів. Це пов'язано з швидким розвитком пошукових систем, які забезпечують ефективність пошуку та використання документів всіх видів та типів.

Для покращення структури публікацій та кращого пошуку використовують рубрикацію. Рубрикація є однією з складових успішної статті, це розмежування тексту на складові частини, використання заголовків, нумерації тощо. Найпростішою рубрикацією є поділ на абзаци [6].

Доведення інформації до суспільства за допомогою преси, радіо, телебачення, розміщення в різних виданнях (газетах, книгах, підручниках) називається науковою публікацією [7]. Публікації відображають основний зміст, новизну наукового дослідження і фіксують завершення певного етапу дослідження або роботи в цілому.

Результати наукових досліджень оприлюднюються у вигляді різних видів публікацій, таких як: монографія, стаття, автореферат, препринт, тези доповідей, наукова доповідь, збірник наукових праць [8].

Стаття – результати дослідження конкретного питання, що мають певне наукове й практичне значення і опубліковані в науковому журналі чи збірнику [8]. Наприклад, в одному з випадків для наукових статей обов'язково мають бути такі елементи [9]:

1. Назва журналу.
2. Розділ журналу, розташовується у верхньому правому куті сторінки, шрифт Times New Roman (Сур), розмір шрифту 14.
3. Індекс УДК - верхній лівий кут сторінки, шрифт Times New Roman (Сур), розмір шрифту 14.
4. Прізвище, ім'я, по батькові автора (-ів), місце роботи (навчання), науковий ступінь, вчене звання, посада - на трьох мовах (українській, російській, англійській) (Якщо стаття написана не українською мовою, то український переклад необов'язковий) - правий кут сторінки, шрифт Times New Roman (Сур), розмір шрифту 14, напівжирний.
5. Назва статті - на трьох мовах (українській, російській, англійській) (Якщо стаття написана не українською мовою, то український переклад необов'язковий) розташування по центру сторінки, шрифт Times New Roman (Сур), розмір шрифту 14, напівжирний, всі прописні, до десяти слів.
6. Анотації - обсяг 100-250 слів, на різних мовах.
7. Ключові слова (мінімум 3 слова або словосполучень з відокремленням їх один від одного комами) - на трьох мовах (українській, російській, англійській) (Якщо стаття написана не українською мовою, то український переклад необов'язковий).
8. Текст статті, який може бути українською, російською, англійською мовами.
9. Перелік джерел.

Науковими зазвичай вважаються статті, що мають такі необхідні елементи [10]:

- вступ - постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями;
- аналіз останніх досліджень і публікацій, в яких започатковано розв'язання даної проблеми і на які спирається автор, виділення невирішених раніше частин загальної проблеми, яким присвячується стаття;
- формулювання мети статті (постановка завдання);
- виклад основного матеріалу дослідження з повним обґрунтуванням отриманих наукових результатів – основна частина статті;
- висновки з даного дослідження і перспективи подальших досліджень з даної теми;
- список літератури, в якому вміщені бібліографічні описи тих джерел і літератури, на які є посилання у тексті статті.

На сьогодні немає однозначної загальноприйнятої класифікації для наукових статей. Дана проблема викликає ряд утруднень при визначенні напрямку до якого відноситься стаття. Це стається через те, що існують різні класифікації і в кожній з них одна і та ж стаття може належати до кількох класифікацій.

До основних класифікацій наукових статей можна віднести:

- 1) УДК;
- 2) класифікація за галуззю знань МОН;
- 3) сучасна класифікація наук;
- 4) класифікація по 5 напрямкам.

Універсальна десяткова класифікація (УДК), класифікація яка широко використовується у всьому світі для систематизації творів науки, літератури і мистецтва, різних видів документів і організації картотек [11].

УДК ділить наукові статті за такими основними розділами:

- 0 Загальний відділ
- 1 Філософія. Психологія

2 Релігія. Геологія

3 Суспільні науки

4 (вільний)

5 Математика. Природничі науки

6 Прикладні науки. Медицина. Техніка. Сільське господарство

7 Мистецтво. Архітектура. Ігри. Спорт

8 Мова. Мовознавство. Художня література. Літературознавство

9 Географія. Біографії. Історія

Наприклад, шифр УДК 004.8 відноситься до загального відділу та розшифровується як «Штучний інтелект».

УДК має кілька недоліків:

1. База УДК занадто вузька, а це призводить до подовження індексів, їхньої нерівномірності в різних відділах, труднощів у введенні нових ділень, коли всі десять певного ступеня вже зайняті [12];

2. Стаття може належати до кількох розділів УДК.

Класифікація за галуззю знань МОН. Галузь знань є основною предметною областю освіти і науки, що включає групу споріднених спеціальностей, за якими здійснюється професійна підготовка [13]. Поділ на галузі зображено в Додатку А.

Для даної класифікації шифр 12-125 буде відноситись до галузі «Інформаційні технології» та розшифровуватись як «Кібербезпека».

Сучасна класифікація наук. Однією з класифікацій статей є сучасна класифікація, що класифікує науки та наукові статті за 25-ма галуззями [14].

Класифікація по 5 напрямкам, в даній класифікації функції науки виділяються за основними видами діяльності дослідників, їх основним завданням, а також сферою застосування отриманих знань [15].

Напрямки за якими здійснюється класифікація:

- природничі науки (математика, фізика, хімія, біологія та ін.);
- технічні науки - система знань про цілеспрямоване перетворення природних сил і процесів у технічні об'єкти;

- медичні науки;
- суспільні науки (економіка, соціологія, політологія, правові науки, демографія та ін.);
- гуманітарні науки (історія держави, історія мистецтва, церкви, теологія, мовознавство і літературознавство, філософія, логіка, психологія та ін.).

Таблиця 1.1 – Сучасна класифікація наук [14]

Шифр	Основні галузі науки	Шифр	Основні галузі науки
01	Фізико-математичні науки	14	Медичні науки
02	Хімічні науки	15	Фармацевтичні науки
03	Біологічні науки	16	Ветеринарні науки
04	Гносеологічні науки	17	Мистецтвознавство
05	Технічні науки	18	Архітектура
06	Сільськогосподарські науки	19	Психологічні науки
07	Історичні науки	20	Військові науки
08	Економічні науки	21	Національна безпека
09	Філософські науки	22	Соціологічні науки
10	Філологічні науки	23	Політичні науки
11	Географічні науки	24	Фізичне виховання та спорт
12	Юридичні науки	25	Державне управління
13	Педагогічні науки		

Класифікація статей використовується для впорядкування досліджень по певних параметрах. Це дозволяє покращити пошук статей, визначити їх напрям, ключові слова та ін.

Для більш точної і конкретнішої класифікації буде використовуватись класифікація за 5 напрямками. Завдяки цьому зменшуються для відповідної статті можливі варіанти класифікації.

Для того, щоб віднести статтю до однієї з класифікацій, додаємо її в базу та автоматично, за допомогою методів аналізу текстів та ключових слів, визначаємо до якої класифікації вона відноситься. Все це відбувається за рахунок порівняння ключових слів даної статті та ключових слів декількох статей кожної з класифікацій.

Мета класифікації – створити для всіх запропонованих наукових статей індикатор розпізнання, що використовується при пошуку та розповсюдженні статті в бібліотеках, електронних ресурсах та ін. Щоб мінімізувати масштаби розрахунків та автоматизувати їх, доцільно використовувати ІТ для даної проблеми.

1.2 Аналіз існуючого програмного забезпечення предметної області

На сучасному етапі кожне видання (стаття, журнал та ін.) має свою класифікацію. Так, для класифікації новин створюються класифікатори, кожен з яких обробляє запроповану новину та виділяє її класифікацію. Прикладом такого додатку є програма під назвою «Методи тематичного моделювання», що використовує методи PLSA та LSA для аналізу (рисунок 1.1) [16].

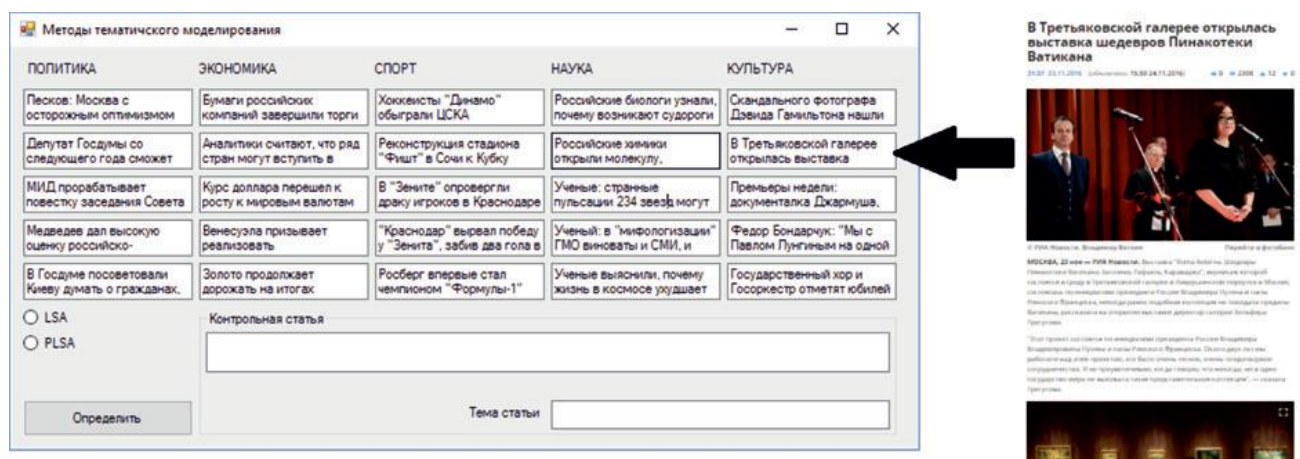


Рисунок 1.1 – Графічний інтерфейс програми [16]

Спочатку в програмі відображаються новини по категоріях, які беруться з сайтів новин. Далі обирається метод за допомогою якого буде обраховуватись тема контрольної статті, та натискається кнопка «Визначити». Результат роботи програми для метода LSA зображено на рисунку 1.2, для метода PLSA на рисунку 1.3.

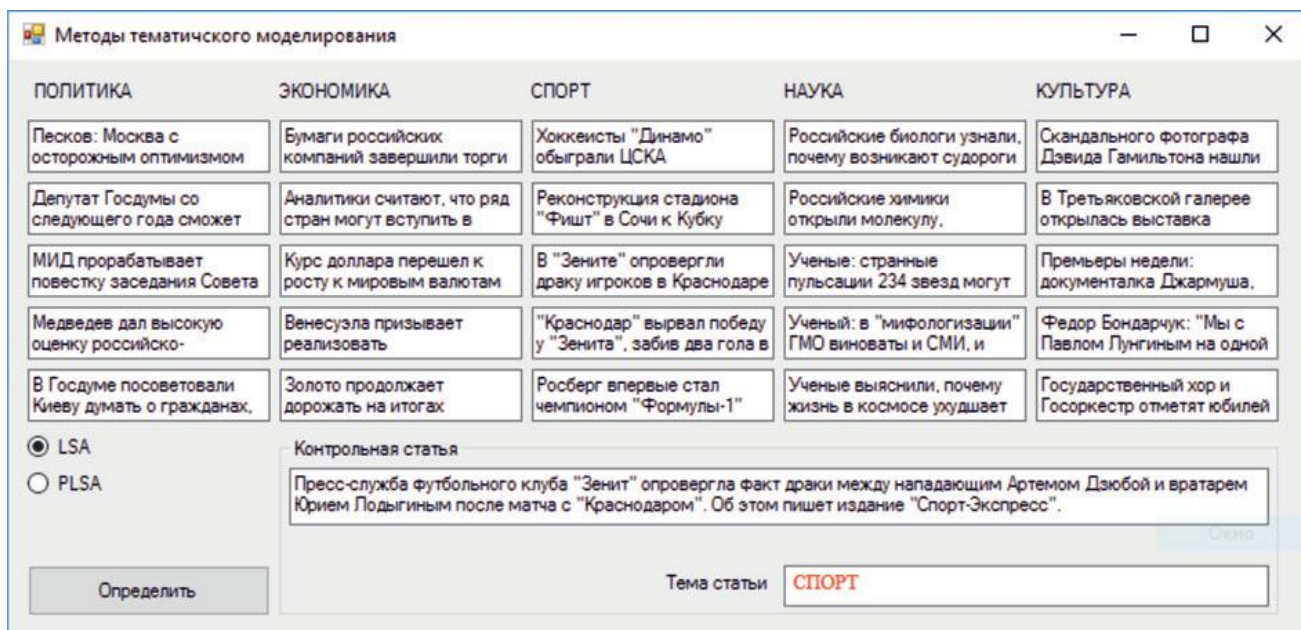


Рисунок 1.2 – Результат работы программы методом LSA [16]



Рисунок 1.3 – Результат работы программы методом PLSA [16]

У даному додатку тексти новин беруться з інтернет ресурсів - сайтів новин. Програма виконує завдання визначення класифікації контрольної статті.

Програма працює коректно, оскільки у приклад були взяті статті з визначеною класифікацією і після обробки даних додатком, класифікації співпали (для методу PLSA). Для методу LSA іноді класифікації не відповідали заявленим. Недоліком даної програми, з технічної точки зору, є неточність одного з методів. Також, до недоліка можна віднести не зручний інтерфейс, оскільки, не видно увесь текст новин з якими порівнюється контрольна.

Для більш наглядного розуміння даної тематики, можна розглянути наукометричні бази, системи індексації та електронні бібліотеки.

До наукометричних баз даних відносяться бібліографічні та реферативні бази, що включають в себе інструменти для пошуку цитованості статей, які опубліковані у наукових виданнях [17]. Наприклад, українська система – «Український індекс наукового цитування», дана система моніторить дані лише по Україні. Сайт даної системи зображено на рисунку 1.4 [18].

The image shows a screenshot of the website 'uincit.urau.ua'. The page title is 'Український індекс наукового цитування' (Ukrainian Index of Scientific Citation) and the subtitle is 'Система наукометричного моніторингу суб'єктів наукової діяльності України' (System of scientometric monitoring of subjects of scientific activity in Ukraine). The navigation menu includes 'Пошук' (Search), 'Аналітика' (Analytics), 'Порівняння' (Comparison), and 'Про проект' (About the project). There are also flags for the UK, Russia, and Ukraine. The main search area has two tabs: 'Пошук вчених' (Search scientists) and 'Пошук установ' (Search institutions). The search form contains the following fields: 'Прізвище:' (Surname), 'Ім'я та/або по батькові:' (Name and/or patronymic), 'Місце роботи:' (Place of work), 'Галузь досліджень:' (Field of research) with a dropdown menu set to 'вибрати все' (select all), and 'ORCID:'. A green 'Пошук' button is located at the bottom right of the form. At the bottom of the page, there are links for 'Політика приватності' (Privacy policy), 'Допомога' (Help), and 'Адміністративна зона' (Administrative area). The Uran logo is in the bottom right corner.

Рисунок 1.4 – Сайт системи «Український індекс наукового цитування» [18]

Для роботи з даними не лише вітчизняних, а ще й закордонних наукових видань існують міжнародні системи цитування, такі як: Web of Science, Scopus, Index Copernicus, Astrophysics та ін [19].

До прикладу, міжнародна система «Web of Science». Дана платформа дозволяє переглядати бази з науковою літературою та патентами, які відносяться до природничих, технічних, біологічних, суспільних, гуманітарних наук і мистецтва.

Головним концептом наукометричної системи є імпакт-фактор, тобто індекс впливовості наукового видання. Сайт даної платформи представлений на рисунку 1.5 [20].

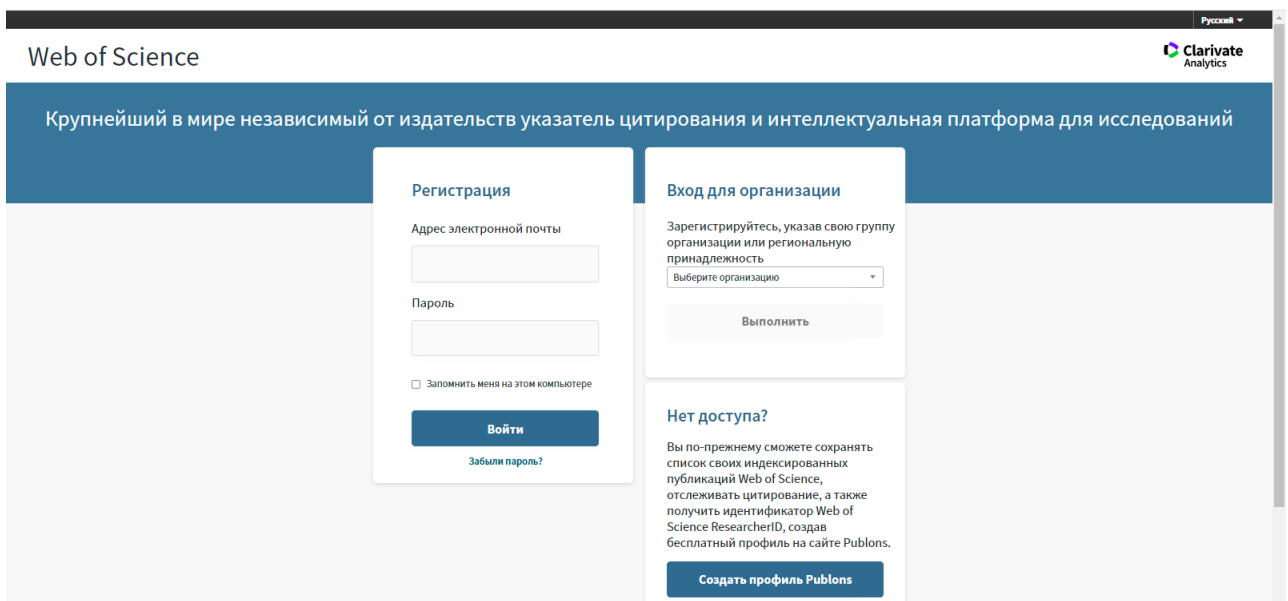


Рисунок 1.5 – Сайт системи «Web of Science» [20]

Ще одним з прикладів системи є електронна бібліотека імені В. І. Вернадського. Дана бібліотека включає в себе книги та статті науково-дослідницького, науково-методичного та культурно-просвітницького характеру, які написані в межах України. Бібліотека відноситься до однієї з найбільших у світі [21]. Сайт бібліотеки зображено на рисунку 1.6 [22].

The screenshot shows the website of the National Library of Ukraine (NBUL) at nbuv.gov.ua. The header includes the library's name and a language selection dropdown. The main navigation menu contains links for 'ПРО БІБЛІОТЕКУ', 'НОВИНИ', 'ЧИТАЧАМ', 'РЕСУРСИ', and 'БІБЛІОТЕКАРЯМ'. The main content area features a large banner with a photograph of the library building and a text box that reads: 'Найбільша бібліотека України. Головний науково-інформаційний центр держави...'. Below the banner is a horizontal menu with categories: 'НАЦІОНАЛЬНИЙ ІНФОРМАЦІЙНИЙ ЦЕНТР', 'НАУКОВА І ВИДАВНИЧА ДІЯЛЬНІСТЬ', 'ФОНДИ І КОЛЕКЦІЇ', 'ФОРУМИ, КОНФЕРЕНЦІЇ', and 'ДАРИ ТА НОВІ НАДХОДЖЕННЯ'. A 'Режим нашої роботи' section is highlighted, containing a 'КАРАНТИН' banner and text explaining the library's transition to remote services due to the COVID-19 pandemic. A sidebar on the right contains a list of links: 'Режим нашої роботи', 'Ді уваги користувачів', and 'Звертаємо увагу!'.

Рисунок 1.6 – Сайт національної бібліотеки України імені В. І. Вернадського [22]

Наведені зразки систем доводять те, що використання технологій для автоматизації класифікації є ефективним способом спрощення та пришвидшення виконання даної задачі.

1.3 Аналіз сучасних засобів створення програмного забезпечення

Для розробки програмного забезпечення можна використати сучасні платформи, такі як: Java та .NET.

Java як платформа – це програмне забезпечення, що представляє собою робоче середовище для роботи програм, написаних на Java (і не тільки). Вона складається з Java API і Java віртуальної машини (JVM) [23].

Java обмежена певним набором функцій API віртуальної машини. У випадку з .NET, вона надає доступ до специфічних можливостей Microsoft Windows, дозволяючи додатку тісно взаємодіяти з операційною системою і

користуватися всіма перевагами платформи Windows, крім того, користувач має справу зі зручним і звичним графічним інтерфейсом Windows.

Ще одна відмінність між Java і .NET полягає в тому, що Java спочатку створювалася для однієї мови програмування - Java. У той час як .NET була спроектована з урахуванням підтримки необмеженої кількості мов програмування [24]. Платформа .NET є, мабуть, найбільш розвиваючою платформою для розробки додатків на даний момент.

Переваги Java платформи [25]:

- потужні засоби розробки;
- кросплатформність;
- велика колекція бібліотек з відкритим кодом;
- багато активних форумів та серверів з допомогою;
- зручна документація.

Переваги платформи .NET [26]:

- повні можливості взаємодії з існуючими кодом;
- можливість повноцінного використання кількох мов програмування при розробці складних додатків;
- загальне середовище виконання для усіх додатків;
- спрощення установки додатків методом копіювання файлів.

Як і у кожній платформі, так і у даних є свої недоліки. До недоліків платформи Java можна віднести те, що вона потребує багато пам'яті, орієнтована на мову Java та обмеження функціоналу [27]. Недоліки платформи .NET [28]:

- достатньо високі вимоги до апаратного забезпечення;
- обмежена підтримка мов програмування;
- документація по ряду нових функцій програмного забезпечення представлена не в повному обсязі.

Підсумувавши усі переваги і недоліки перерахованих платформ, взявши до уваги доцільність їх використання для обраного напрямку, оптимальним є вибір платформи .NET.

1.4 Постановка задачі та вимоги до розробки інформаційної системи

Метою кваліфікаційної роботи бакалавра є розробка автоматизованої інформаційної системи тематичної класифікації наукових статей на платформі .NET. В програмі необхідно реалізувати наступні основні функції:

- вибір кількості класів для класифікації;
- вибір кількості ключових слів у класах;
- обробка тексту статей;
- формування послідовних множин слів статей;
- формування множин унікальних слів статей;
- автоматизований підбір альтернативних текстів;
- обчислення значень TF-IDF унікальних слів класів;
- сортування множин ключових слів класів;
- обмеження множин ключових слів класів;
- перехід між тематичними класифікаціями;
- пошук збігів тестової статті за ключовими словами класів;
- обчислення оцінки приналежності тестової статті до класів;
- візуалізація результатів роботи системи.

Розділ 2

Проектування інформаційної системи

2.1 Моделі, методи, інформаційна технологія системи

2.1.1 Схема інформаційної технології

Схему інформаційної технології тематичної класифікації наукових статей зображена на рисунку 2.1. Вхідними даними системи є множина статей для кожного з класів, тестова стаття для аналізу та параметри роботи.

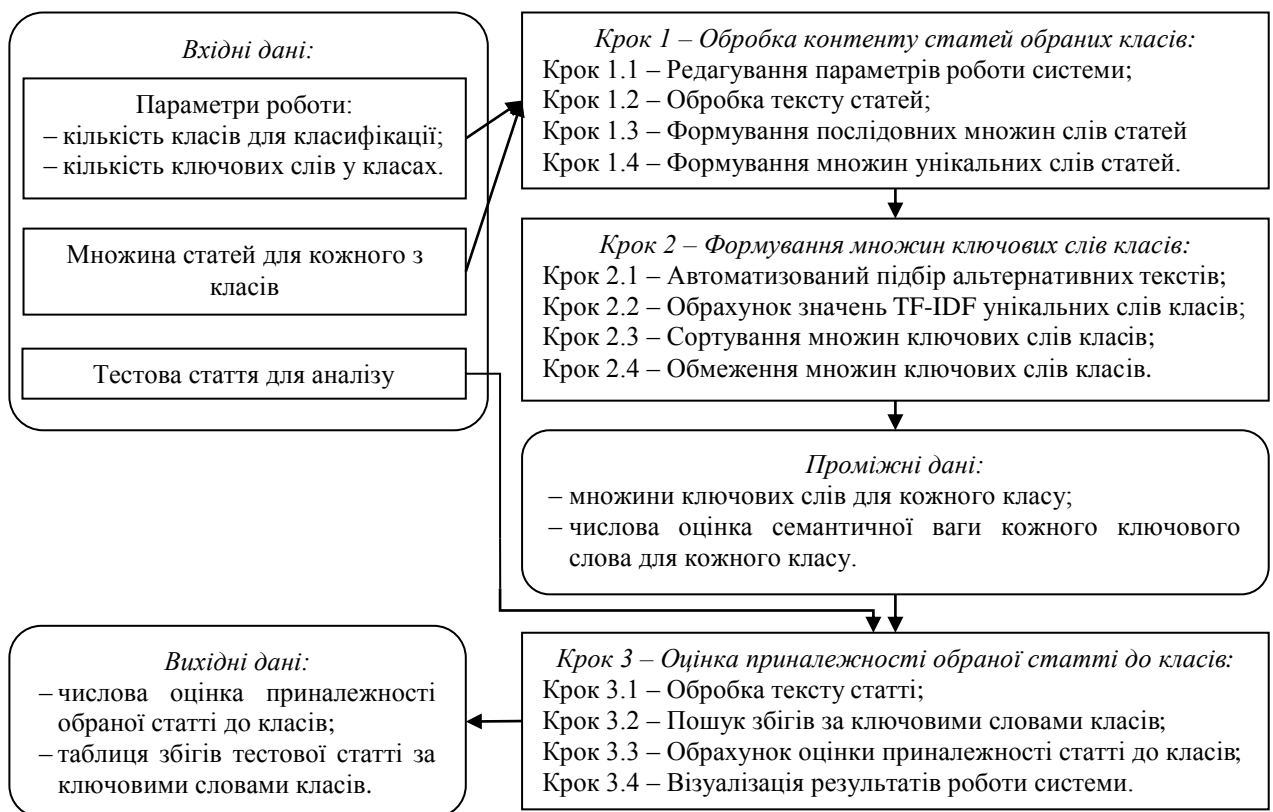


Рисунок 2.1 – Схема інформаційної технології тематичної класифікації наукових статей

З вхідних даних розпочинається робота системи, де першим етапом є Крок 1. Спершу на Кроці 1 виконується обробка контенту статей обраних класів. Зокрема, спочатку на Кроці 1.1 виконується редагування параметрів роботи системи (кількість класів для класифікації та кількість ключових слів у класах).

Тобто, користувач вводить значення параметрів, або починає роботу програми з автоматично введеними параметрами.

На Кроці 1.2 виконується обробка тексту статей, де використовується множина статей для кожного з класів. Під обробкою розуміється відкидання з тексту статей пунктуаційних знаків, чисел та інших символів, залишаючи лише слова в нижньому реєстрі. Після виконується Крок 1.3, під час якого формуються послідовні множини слів статей, слова відділяються один від одного в тексті статті та виводяться у відповідне поле.

Завершальним кроком даної частини є Крок 1.4 на якому відбувається формування множин унікальних слів статей. Це відбувається за допомогою відбирання кожного унікального слова з переліку множини послідовних слів, і до кожного слова додається його частота у множині та кількість раз які воно зустрічалось.

Наступним виконується Крок 2 – формування множин ключових слів класів. Спершу виконується Крок 2.1, на якому відбувається автоматизований підбір альтернативних текстів. Щоб подальший обрахунок був правильним, альтернативні тексти мають бути з відмінних від обраного класу класифікації, тобто, альтернативні тексти підбираються в залежності від обраного класу, тим самим виключаючи його статті з переліку альтернативних.

На Кроці 2.2 відбувається обрахунок значень TF-IDF для унікальних слів класів. Даний крок є досить важливим, оскільки, саме завдяки значенням TF-IDF і будуть визначатись ключові слова. Далі виконується Крок 2.3 – Сортування множин ключових слів класів. Сортування відбувається за оцінкою TF-IDF.

Останнім кроком виконується Крок 2.4, який обмежує множини ключових слів класів. Обмеження виконується згідно значення яке було введене на Кроці 1.1. З Кроку 2 отримуються проміжні дані: множини ключових слів для кожного класу та числова оцінка семантичної ваги кожного ключового слова для кожного класу.

Завершальним етапом є Крок 3 на якому визначається оцінка приналежності обраної статті до класів. На початку виконується Крок 3.1 на

якому відбувається обробка тексту статті. Стаття береться автоматичним чином, вона вже має свій певний клас, за допомогою чого перевіряється коректність роботи системи. Під обробкою розуміється формування множини унікальних, а потім і ключових слів статті з визначенням для кожного ключового слова значення TF-IDF.

На Кроці 3.2 відбувається пошук збігів за ключовими словами класів. Тобто, перевіряється кожне ключове слово тестової статті з наявністю такого ж ключового слова в статтях класів класифікації. Далі виконується Крок 3.3, де обраховуються оцінки приналежності статті до класів. Цей етап відбувається за допомогою сумування оцінок TF-IDF всіх слів що є ключовими для тестової статті. Після визначення оцінки приналежності до кожної статті відбувається визначення відсотка приналежності.

Останнім кроком, який приводить до вихідних даних є Крок 3.4, який виводить результати роботи системи. Вихідними даними системи є числова оцінка приналежності обраної статті до класів та таблиця збігів тестової статті за ключовими словами класів. Таким чином, було представлено та описано схему інформаційної технології тематичної класифікації наукових статей.

2.1.2 Метод пошуку ключових слів

Для кожного класу класифікації необхідно визначити множину ключових слів. Для цього на початку автоматично формується список альтернативних множин статей усіх класів окрім обраного, далі система переходить до обрахунку значень оцінки TF-IDF (рисунок 2.2). Обрахунок значень оцінки TF-IDF для унікальних слів класів відбувається за допомогою методів семантичного аналізу. Оцінка TF-IDF складається з частоти слова TF та оберненої частоти множини статей – IDF. Частотна оцінка TF (term frequency) є частотою зустрічей певного слова i у множині статей, що розглядається, й обчислюється за допомогою наступної формули [29]:

$$TF_i = \frac{n(i)}{N}, \quad (2.1)$$

де $n(i)$ – кількість появ слова i у множині статей, N – загальна кількість слів у множині статей.

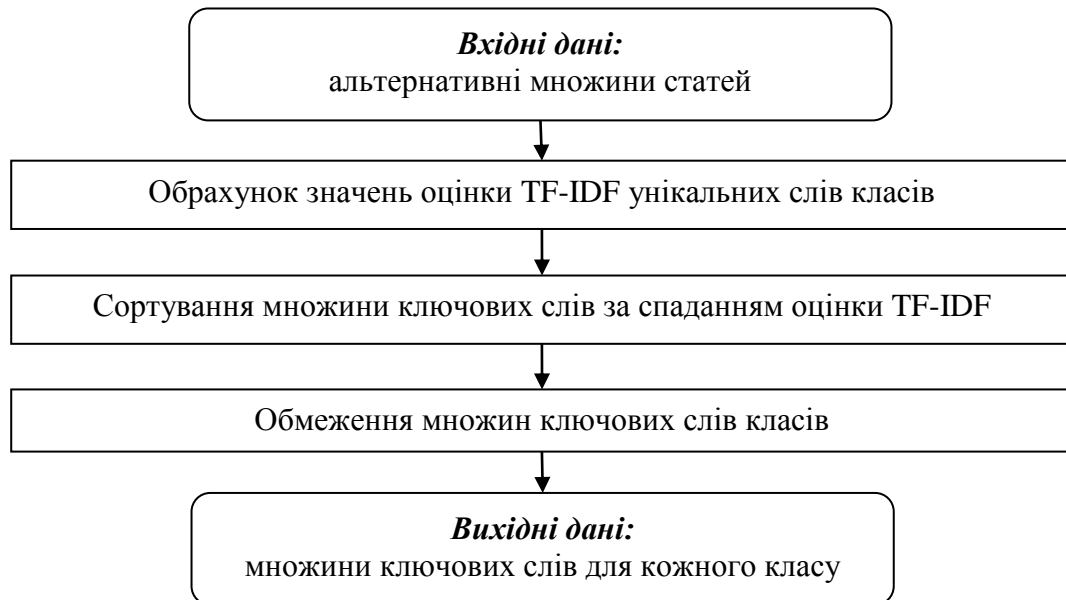


Рисунок 2.2 – Схема послідовності дій для визначення множини ключових слів

Обернена частота множини статей IDF (inverse document frequency) є значенням яке показує на скільки слово є цінним для окремого класу, й обраховується таким чином [29]:

$$IDF_i = \log \frac{|D|}{|(d_i \supset t_i)|}, \quad (2.2)$$

де $|D|$ – значення кількості множин статей, $|d_i \supset t_i|$ – кількість множин статей, де зустрічається слово t_i , за умови коли $n_i \neq 0$.

Значення TF-IDF обраховується добутком обрахованих вище значень, тобто за наступною формулою:

$$TF - IDF_i = TF_i \cdot IDF_i, \quad (2.3)$$

де TF_i – частота певного слова i у множині статей, IDF_i – обернена частота певного слова i у множині статей.

В результаті обраховано оцінку, яка високо оцінює ті слова, що часто зустрічаються в конкретній статті та, відповідно, низьку оцінку тим словам, які рідше зустрічаються.

2.2 Інформаційна структура системи

2.2.1 Структура компонентів інформаційної системи

Автоматизована інформаційної системи тематичної класифікації наукових статей складається з трьох підсистем та бази даних (рисунок 2.3).

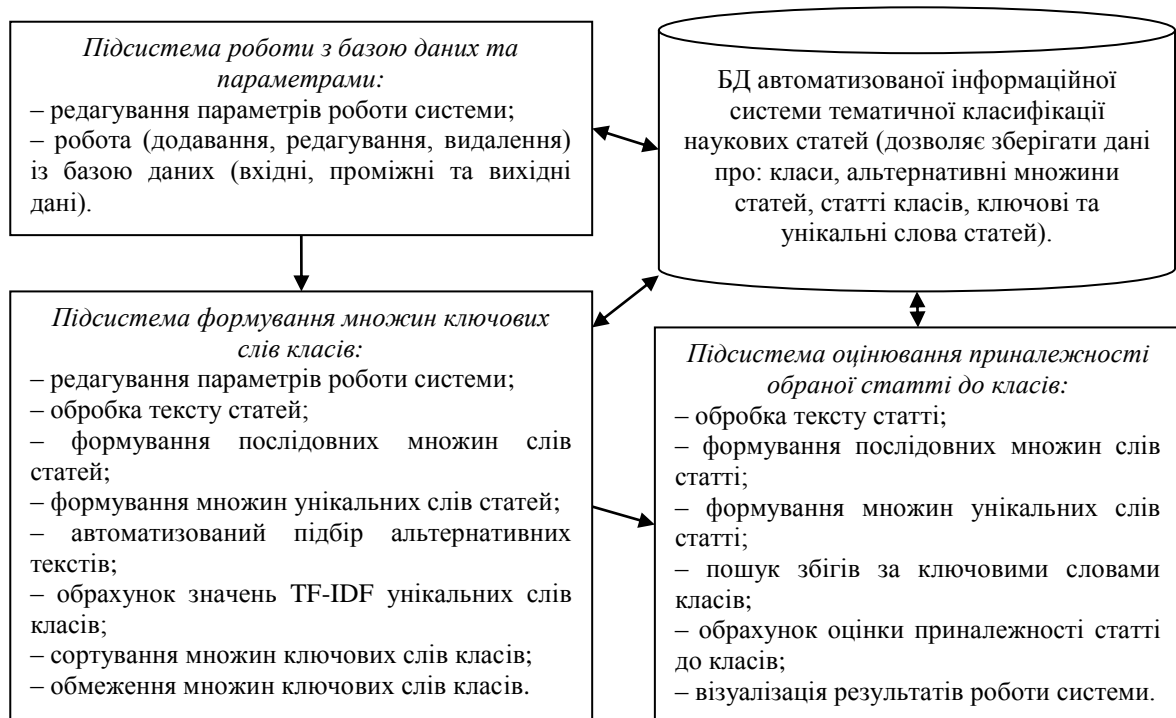


Рисунок 2.3 – Структура компонентів інформаційної системи тематичної класифікації наукових статей

Першою підсистемою є підсистема роботи з базою даних та параметрами. Дана підсистема виконує редагування параметрів роботи системи та роботу із базою даних таку, як: додавання, редагування та видалення усіх даних (вхідні, проміжні та вихідні дані).

Наступною підсистемою є підсистема формування множин ключових слів класів. Вона виконує ряд функцій:

- редагування параметрів роботи системи (за рахунок введення значень кількості класів класифікації та кількості ключових слів у класах);
- обробка тексту статей (за рахунок алгоритмів, які очищають текст статей від символів, які не є буквами);
- формування послідовних множин слів статей (за рахунок відокремлення слів один від одного в множині статей);
- формування множин унікальних слів статей (за рахунок визначення унікальних слів з послідовної множини статей);
- автоматизований підбір альтернативних текстів (за рахунок алгоритму порівняння класів класифікації з обраним класом, оскільки альтернативними є ті тексти, які не відносяться до обраного класу класифікації);
- обрахунок значень TF-IDF унікальних слів класів;
- сортування множин ключових слів класів та обмеження множин ключових слів класів (обмеження відбувається за значенням, яке вводиться на етапі роботи першої підсистеми).

Останньою є підсистема оцінювання приналежності обраної статті до класів. Завдяки цій підсистемі відбувається:

- обробка тексту обраної статті (як і у попередній підсистемі, за рахунок алгоритмів, які очищають текст статей від символів, які не є буквами);
- формування послідовних множин та унікальних множин слів статті;
- пошук збігів за ключовими словами класів (перевіряється кожне ключове слово тестової статті з наявністю такого ж ключового слова в статтях класів класифікації);
- обрахунок оцінки приналежності статті до класів (за рахунок сумування оцінок TF-IDF всіх слів, що є ключовими для тестової статті, та обрахунку відсотка приналежності);
- візуалізація результатів роботи системи (відображення вихідних даних системи).

База даних системи взаємодіє з усіма підсистемами зберігаючи в собі дані про: класи, альтернативні множини статей, ключові та унікальні слова статей.

Таким чином, представлена структура компонентів інформаційної системи тематичної класифікації наукових статей виконує поставлене завдання.

2.2.2 Структура бази даних інформаційної системи

Для забезпечення збереження даних було розроблено БД, даталогічна модель якої зображена на рисунку 2.4. База даних включає в себе такі таблиці: CLASSIFICATION, THEM_TEXTS, KEY_WORDS, TEST_TEXT, TEST_TEXT_WORDS, CLASS_TEST_WORDS та CLASS_TEST_TEXTS.

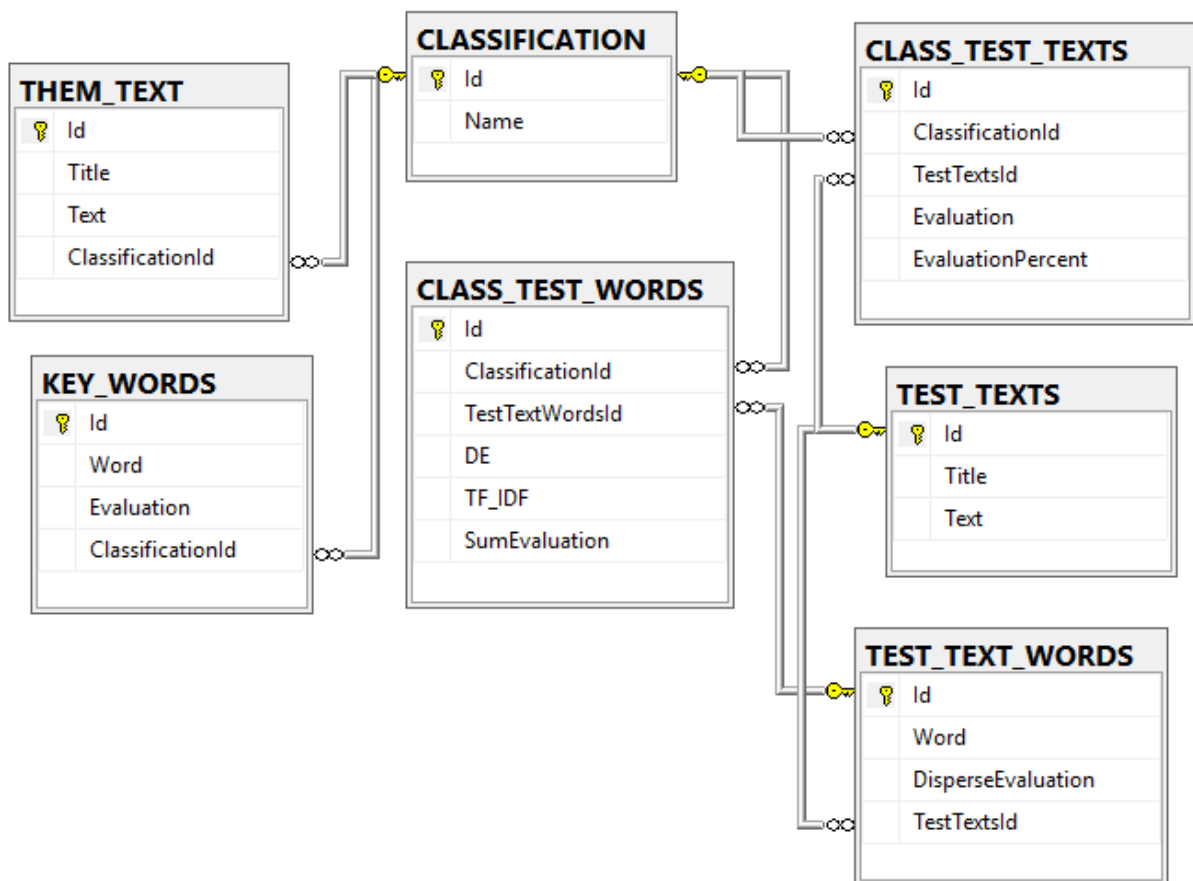


Рисунок 2.4 – Даталогічна модель БД автоматизованої інформаційної системи для класифікації тексту

Таблиця «CLASSIFICATION» є головною таблицею БД і використовується для збереження інформації про усі можливі класи текстів в програмі (таблиця 2.1). Поле «ID» є первинним ключем і має цілочисельний тип

даних. Поле «Name» має стрічковий тип даних nvarchar, а також обмеження в максимальну кількість символів – 100. Дане поле відповідає за назви класів, на які можуть бути класифіковані статті.

Таблиця 2.1 – Атрибути таблиці «CLASSIFICATION»

№ п/п	Назва	Тип поля	Опис
1	ID	int	Первинний ключ призначений для ідентифікації записів
2	Name	nvarchar(100)	Назва категорії

Для збереження основних даних про статті, які будуть класифікуватись за тематикою використовуємо таблицю 2.2. Поле «ID» є первинним ключем і має цілочисельний тип даних. Поля «Title» і «Text» мають строковий тип даних nvarchar, які відповідають за заголовок статті, а також за сам текст статті. Дані два поля мають обмеження в максимальну кількість символів – для заголовку це 1000 символів, а для тексту це максимальна можлива кількість символів для даного типу даних, яка складає більше ніж 1 мільярд символів. Поле «ClassificationId» - це вторинний ключ, що пов'язує цю таблицю з таблицею «Classification» і має цілочисельний тип даних.

Таблиця 2.2 – Атрибути таблиці БД «THEM_TEXTS»

№ п/п	Назва	Тип поля	Опис
1	ID	int	Первинний ключ призначений для ідентифікації записів
2	Title	nvarchar(1000)	Заголовок тексту
3	Text	nvarchar(max)	Текст
4	ClassificationId	int	Вторинний ключ, що пов'язується із таблицею «Classification»

Для збереження даних про ключові слова з статтів використовуємо таблицю «Key_Words» (табл. 2.3). Первинним ключем таблиці з цілочисельним типом даних є поле «ID». Поле «Word» з строковим типом даних відповідає за ключові слова, знайдені в статтях, а поле «Evaluation» за вагу коефіцієнта, який

відповідає на скільки сильно слово відповідає класу класифікації розміщеній в полі «ClassificationId». Поле «Evaluation» містить числовий тип даних з плаваючою крапкою. «ClassificationId» - це вторинний ключ, який поєднує таблицю «KeyWords» із таблицею «Classification».

Таблиця 2.3 – Атрибути таблиці БД «KEY_WORDS»

№ п/п	Назва	Тип поля	Опис
1	ID	int	Первинний ключ призначений для ідентифікації записів
2	Word	nvarchar(50)	Ключові слова
3	Evaluation	float	Вага коефіцієнта
4	ClassificationId	int	Вторинний ключ, що пов'язується із таблицею «Classification»

Для збереження даних про статті, які будуть слугувати тестовими зразками для знаходження ключових слів і присвоєння їм класів використовуємо таблицю 2.4. Основними атрибутами таблиці є заголовок статті і сам текст статті, які відповідають полям «Title» і «Text». Поле «ID» - це первинний ключ таблиці, який містить цілочисельний тип даних.

Таблиця 2.4 – Атрибути таблиці БД «TEST_TEXT»

№ п/п	Назва	Тип поля	Опис
1	ID	int	Первинний ключ призначений для ідентифікації записів
2	Title	nvarchar(1000)	Заголовок тексту
3	Text	nvarchar(max)	Текст

Таблиця «TEST_TEXT_WORDS» зберігає дані про слова знайдені в тестових зразках статті і дисперсійну оцінку, яка відповідає важливості кожного слова конкретної статті (таблиця 2.5). «ID» - це первинний ключ таблиці, який містить цілочисельний тип даних. Поле «Word» має максимальну кількість символів – 50, і відповідає строковому типу даних. Дисперсійна оцінка зберігається у полі «DisperseEvaluation» з числовим типом даних з плаваючою

крапкою. «TestTextsId» - це вторинний ключ, який поєднує таблицю 2.5 із таблицею «Test_Texts» і має цілочисельний тип даних.

Таблиця 2.5 – Атрибути таблиці БД «TEST_TEXT_WORDS»

№ п/п	Назва	Тип поля	Опис
1	ID	int	Первинний ключ призначений для ідентифікації записів
2	Word	nvarchar(50)	Ключове слово
3	DisperseEvaluation	float	Дисперсна оцінка ключового слова
4	TestTextsId	int	Вторинний ключ, що пов'язується із таблицею «Test_Texts»

Таблиця «CLASS_TEST_WORDS» (таблиця 2.6) є розвідною для таблиць «Classification» і «Test_Text_Words». Поле «ID» - це первинний ключ таблиці, який містить цілочисельний тип даних. Поля «ClassificationId» і «TestTextWordsId» є вторинними ключами до відповідних таблиць і мають цілочисельний тип даних. Усі інші поля мають числовий тип даних з плаваючою крапкою. Поле «DE» відповідає за дисперсійну оцінку, яка надається слову. «TF-IDF» відображає коефіцієнт TF-IDF, а «SumEvaluation» – сумарний коефіцієнт після обчислень TF-IDF і дисперсійної оцінки.

Таблиця 2.6 – Атрибути таблиці БД «CLASS_TEST_WORDS»

№ п/п	Назва	Тип поля	Опис
1	ID	int	Первинний ключ призначений для ідентифікації записів
2	ClassificationId	int	Вторинний ключ, що пов'язується із таблицею «Classification»
3	TestTextWordsId	int	Вторинний ключ, що пов'язується із таблицею «Test_Text_Words»
4	DE	float	Дисперсна оцінка
5	TF-IDF	float	Коефіцієнт TF-IDF
6	SumEvaluation	float	Сумарна оцінка коефіцієнта

Для співвідношення класу класифікації із тестовою статтею використовуємо розвідну таблицю 2.7. Поле «ID» - це первинний ключ, який має цілочисельний тип даних. Поля «ClassificationId» і «TestTexts_FK» є вторинними ключами, що пов'язують ці поля з таблицями «Classification» і «TestTexts» відповідно та мають цілочисельний тип даних. Поле «Evaluation» і «EvaluationPercent» мають числовий тип даних із плаваючою крапкою і відповідають відображення вагового коефіцієнта і його представлення у відсотках.

Таблиця 2.7 – Атрибути таблиці БД «CLASS_TEST_TEXTS»

№ п/п	Назва	Тип поля	Опис
1	ID	int	Первинний ключ призначений для ідентифікації записів
2	ClassificationId	int	Вторинний ключ, що пов'язується із таблицею «Classification»
3	Evaluation	float	Ваговий коефіцієнт
4	TestTextsId	int	Вторинний ключ, що пов'язується із таблицею «Test_Texts»
5	EvaluationPercent	float	Коефіцієнт у відсотках

Таким чином, в базі даних зберігається вся інформація, яка потрібна для роботи автоматизованої інформаційної системи тематичної класифікації наукових статей.

2.3 Вибір засобів розробки інформаційної системи

2.3.1 Вибір мови програмування

Перш за все, для розробки ІС потрібно обрати мову програмування. Для порівняння було обрано мови: С++ та С#. Дані мови користуються популярністю серед розробників різних видів додатків, тому вони залишаються актуальними.

Обидві мови об'єктно-орієнтовні, прості в розумінні та мають досить велику інформаційну базу по роботі з ними [31]. Мова програмування С++ була

розроблена Б. Страуструпом у 1979 році. Спочатку вона була названа «Сі з класами» а у 1983 році, розробник перейменував мову у С++. Мова підтримує три парадигми програмування, а саме: процедурну, об'єктно-орієнтовну та узагальнену [32]. Перевагами мови С++ можна назвати:

- швидкодія;
- висока захищеність;
- підходить для розробки майже будь-якого продукту (web, desktop, etc.).

Натомість до недоліків даної мови можна віднести [33]:

- складне написання коду;
- високий поріг входження;
- відсутня повноцінна інформація про типи;
- модулі можуть стати сильно пов'язаними, якщо використовувати `#define`.

Мова програмування С# об'єктно-орієнтована, проста у використанні, має потужне підґрунтя, чималу кількість документації і порад на форумах та серверах по її використанню [34]. Вона була розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під керівництвом Microsoft. Дана мова має переваги:

- вона розвивається;
- має велику документацію;
- легко опановується, потребується на ринку;
- має чималу сферу використання;
- не підтримує множинне спадкування класів.

Таким чином, для розробки автоматизованої інформаційної системи тематичної класифікації наукових статей було обрано мову програмування С#.

2.3.2 Вибір фреймворку

Фреймворк – програмний продукт, який спрощує створення і підтримку технічно складних або навантажених проектів [35]. Фреймворк, як правило,

містить тільки базові програмні модулі, а всі інші, специфічні для проекту компоненти реалізуються розробником на їх основі. Тим самим досягається не тільки висока швидкість розробки, а й велика продуктивність і надійність рішень.

Використання фреймворків стало досить популярним, завдяки тому, що він надає готову структуру додатка, що значно зменшує час при розробці великих програм. Найпопулярнішими на сьогодні є фреймворки .NET та .NET Core.

Перевагами .NET Framework є [36]:

- можливість розширення ієрархії класів;
- можливість реалізації будь-яких бізнес-процесів;
- швидкість роботи програми на базі фреймворку;
- безкоштовне оновлення версій.

Різниця між фреймворками у тому, що .NET Core модульний, кросплатформний та має можливість застосування хмарних технологій [37].

.NET Core – модульний фреймворк, тобто, кожен її компонент оновлюється через менеджер пакетів NuGet і тому можна оновлювати кожен модуль окремо, в той час як .NET Framework оновлюється разом. .NET Core, в основному, застосовується для розробки консольних додатків, веб-додатків, а також хмарних служб. А .NET Framework більш поширений при розробці додатків під Windows і веб-додатків ASP.NET.

Для розробки даної інформаційної системи було обрано .NET Framework, оскільки даний фреймворк є більше досліджуваний, а отже має більшу документацію та популярність.

2.3.3 Вибір СКБД

При розробці автоматизованої інформаційної системи тематичної класифікації наукових статей важливим етапом є вибір СКБД. Найпопулярнішими зараз є СКБД Microsoft SQL Server 2012 та MySQL.

Що одна, що інша СКБД базуються на SQL (структурована мова запитів), тому при їх використанні розробник може помітити декілька однакових моментів, зокрема, можливість зберігати дані у таблицях, можливість посылатись на первинні та зовнішні ключі та можливість додавати кілька баз у СКБД або у один сервер.

Microsoft SQL Server 2012 – це закінчена пропозиція в області баз даних та аналізу даних, що використовується для швидкого створення масштабних вирішень електронної комерції, бізнесу і сховищ даних [38]. Дана СКБД розроблена компанією Microsoft, доступна на різних мовах та має достатньо простий синтаксис та інтерфейс.

Використання даної СКБД дає змогу розробнику з легкістю підключити базу даних до додатків розроблених на основі .NET фреймворку. Якщо порівнювати цю СКБД з іншими системами керування, то SQL Server має такі переваги [39]:

- масштабованість, підтримка баз даних великих розмірів;
- безпека даних;
- зв'язаність;
- зручність використання;
- швидкість функціонування;
- відкритий код.

Такі переваги роблять систему керування базами даних більш затребуваною та поширенішою у використанні.

СКБД MySQL була розроблена компанією Oracle. Вона має складніший синтаксис в порівнянні з SQL Server. З точки зору функціональності, MySQL не настільки розвинений, що надає перевагу при виборі SQL Server. Зокрема, якщо звернути увагу на запити, то MySQL не дає можливості скасовувати запит в процесі його виконання.

Також, при порівнянні цих двох систем було проведено дослідження масштабованості [40]. В результаті MySQL показав при зростанні кількості рядків, збільшення часу в два рази, натомість, SQL Server продемонстрував

збільшення часу у меншій кількості. Це означає, що SQL Server масштабується краще, ніж MySQL. Тому, після усіх наведених переваг та функцій, для розробки було обрано СКБД – SQL Server.

Отже, для розробки програмного продукту було обрано такий комплекс програмних інструментів: мова програмування C#, .NET Framework, СКБД SQL Server 2012.

Розділ 3

Програмна реалізація інформаційної системи

3.1 Структура та функціональне призначення складових системи

Для структуризації системи було сформовано діаграму класів, яка зображена на рисунку 3.1.

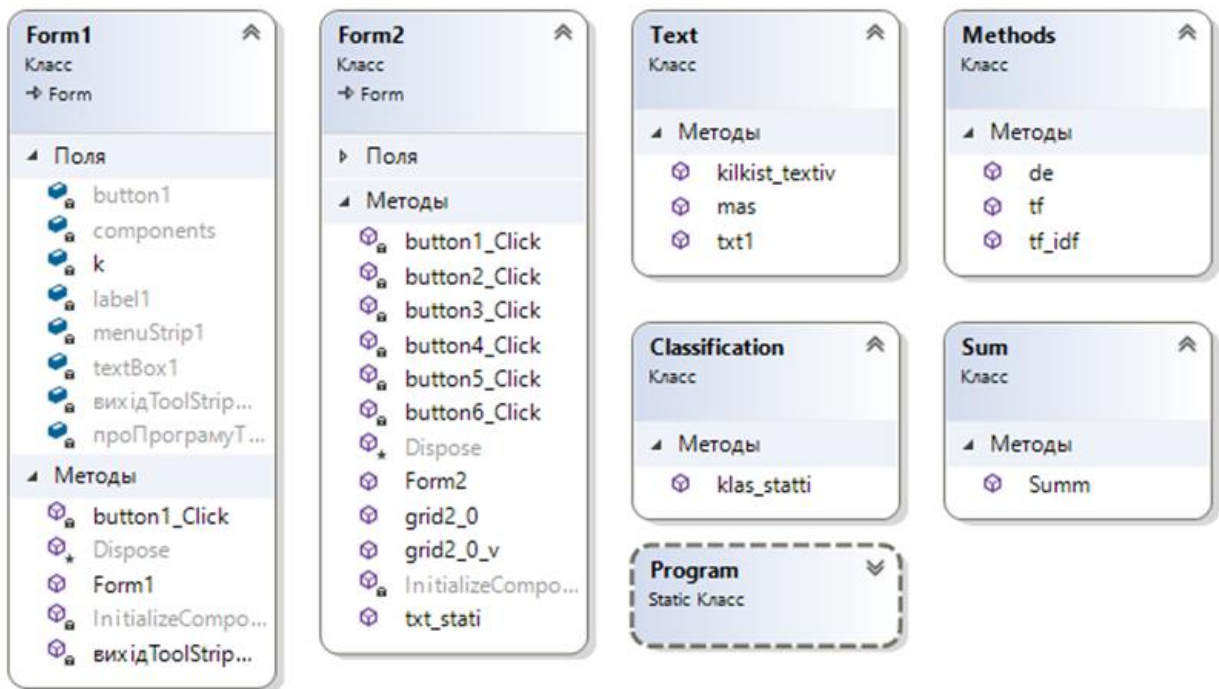


Рисунок 3.1 – Діаграма класів системи

Програма розпочинає свою роботу з класу Program. Клас Form1 реалізує вибір кількості класифікацій системи. Клас Form2 є основним графічним інтерфейсом, з яким буде працювати користувач. В цьому класі реалізовується відображення даних та проводиться їх частковий обрахунок. Клас Text включає методи які відповідають за обробку текстів в програмі. Клас Methods включає в себе методи визначення TF та IDF статей. Клас Classification за допомогою метода записує дані до класифікацій. Клас Sum обраховує в методі Summ суми коефіцієнтів по ключових словах.

Інформаційна система включає в себе 2 основні форми. Перша форма надає користувачу можливість обрати кількість класифікацій та кількість

ключових слів що будуть відображатись, а друга форма включає 3 вкладки з інформацією та обрахунками для тематичних класифікацій. Модель системи зображено на рисунку 3.2.

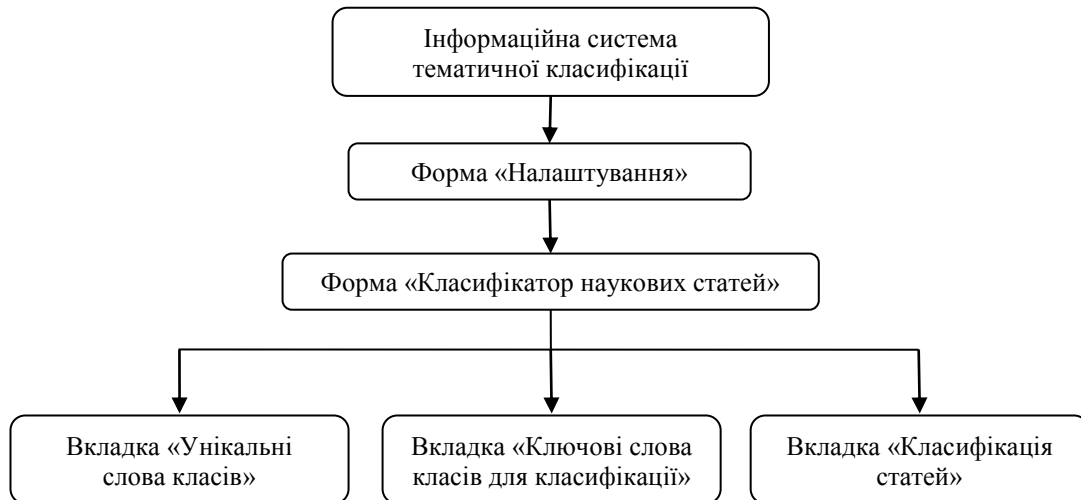


Рисунок 3.2 – Схема модулів інтерфейсу інформаційної системи

На початку роботи з системою користувачу виводиться головна форма налаштування, де можна обрати кількість класів для класифікації та кількість ключових слів у класах. Після натискання на кнопку «Почати роботу», відкривається перелік класів класифікації та множини статей кожного з них. На ній користувач, також, може визначити перелік унікальних слів множини статей.

Далі з головної вкладки, він може перейти на вкладку «Ключові слова класів для класифікації» та вкладку «Класифікація статей», де визначаються ключові слова класів для класифікації та класифікація тестової статті відповідно.

На вкладці «Ключові слова класів для класифікації» користувач має можливість обрахувати значення TF-IDF для кожного слова. Після обрахунку слова сортуються за спаданням та обмежуються у кількості.

Щоб провести аналіз власної наукової статті, користувач переходить на вкладку «Визначення класифікації». Далі завантажує тестову наукову статтю і натисканням на кнопку «Виконати всі кроки визначення приналежності обраної статті до класів автоматично» отримує необхідні обрахунки ключових слів та

класифікацію до якої найбільше підходить тестова стаття. В додатку В описані загальні етапи роботи проекту, у вигляді бізнес-процесів.

Таким чином, в результаті розробки інформаційної системи було отримано вище згадані модулі інтерфейсу, з якими буде працювати користувач та класи системи, які реалізують необхідні функції.

3.2 Особливості реалізації складових системи

Програмні модулі є важливим осередком програми. Найголовніші з них:

- визначення унікальних слів статті;
- визначення ключових слів класів для класифікації;
- визначення класифікації тестової статті.

Визначення унікальних слів статті відбувається за допомогою методів сортування та порівняння. Після виконання розрахунків з'являються дані в множині унікальних слів (рисунок 3.3).

Слово	К-ть	Частота
на	10	0,0166112...
репертуару	2	0,0033222...
статей	3	0,0049833...
журналу	4	0,0066445...
харківського	3	0,0049833...
наукового	7	0,0116279...
товариства	5	0,0083056...
вісник	3	0,0049833...
природознавства	4	0,0066445...
інститутизації	2	0,0033222...
та	9	0,0149501...
а	4	0,0066445...
її	5	0,0083056...
розвитку	2	0,0033222...
для	3	0,0049833...

Рисунок 3.3 – Множина унікальних слів класу

Відповідним чином, але з використанням методів TF, IDF та TF-IDF, що розташовуються в класі «Methods», визначається множина ключових слів класифікації (рисунок 3.4). Спочатку значення частоти слова TF, далі обернена частота IDF та TF-IDF добуток цих двох частот.

Слово	TF	IDF	TF-IDF
наукового	0,011627906976...	1,945910149055...	0,022626862198...
товариства	0,008305647840...	1,945910149055...	0,016162044427...
журналу	0,006644518272...	1,945910149055...	0,012929635541...
природознавства	0,006644518272...	1,945910149055...	0,012929635541...
україни	0,006644518272...	1,945910149055...	0,012929635541...
вісник	0,004983388704...	1,945910149055...	0,009697226656...
статей	0,004983388704...	1,945910149055...	0,009697226656...
журнал	0,004983388704...	1,945910149055...	0,009697226656...
житті	0,004983388704...	1,945910149055...	0,009697226656...
харківського	0,004983388704...	1,945910149055...	0,009697226656...
вивчення	0,004983388704...	1,945910149055...	0,009697226656...
видання	0,004983388704...	1,945910149055...	0,009697226656...
без	0,004983388704...	1,945910149055...	0,009697226656...
професора	0,004983388704...	1,945910149055...	0,009697226656...
або	0,006644518272...	1,098612288668...	0,007299749426...

Рисунок 3.4 – Вкладка «Ключові слова класів для класифікації»

Обрахунок значення частоти слова TF відбувається за допомогою наступного програмного коду:

```
public double[] tf(string[] nowy_mas, int[] mas_kilk, int zagal_kilk, double[] chast)
{
    for (int i = 0; i < nowy_mas.Length; i++)
    {
        chast[i] = Convert.ToDouble(mas_kilk[i]) / zagal_kilk;
    }
    return chast;
}
```

Обрахунок значення оберненої частоти IDF відбувається за допомогою наступного програмного коду:

```
public double[] tf_idf(string[] nowy_mas, int[] kilk_text, double[] idf)
{
    for (int i = 0; i < nowy_mas.Length; i++)
    {
        idf[i] = Math.Log((7 / (kilk_text[i] + 1)));
    }
    return idf;
}
```

Останнім модулем є визначення класифікації тестової статті. Даний модуль реалізовується натисканням кнопки «Виконати всі кроки визначення приналежності обраної статті до класів автоматично», що запускає обробку даних та викликає метод «klas_statti» з класу «Classification», та обраховує оцінку і відсоток приналежності тестової статті до кожної тематичної класифікації (рисунок 3.5).

Класифікатор наукових статей Шпичко

Класи статей

Природничі науки

Технічні науки

Медичні науки

Суспільні науки

Гуманітарні науки

Унікальні слова класів | Ключові слова класів для класифікації | Класифікація статей

Визначення приналежності обраної статті до класів

Тестова стаття для класифікації:

Існуючі методи керування та виміру високочастотних ліній зв'язку мають можливість дозволити забезпечити досить хороші характеристики для контролю вимірюваних значень. Низькочастотна лінія в основному представлена коаксіальною лінією телефонної провідної лінії. Як і високочастотні лінії, ці провідні лінії також повинні контролюватися. Для вирішення проблеми виміру відстані пошкодження існує велика кількість приладів, заснованих на використанні одного з вищезазначених методів (як правило, імпульсного). Далекоміри, засновані на вимірюванні фазового зсуву, ще не знайшли широкого практичного застосування. Пристрої, які засновані на виявленні імпульсних ліній.

Оцінка приналежності обраної статті до класів:

Класифікація	Оцінка	Відсоток
Природничі	0	0%
Технічні	0,0689659...	76,31%
Медичні	0,0084530...	9,35%
Суспільні	0	0%
Гуманітарні	0,0129629...	14,34%

Показники збігів тестової статті за ключовими словами класів:

	Природничі	Технічні	Медичні	Суспільні	Гуманітарні
існуючі	0	0	0	0	0
методи	0	0	0	0	0
керування	0	0	0	0	0
та	0	0	0	0	0
виміру	0	0	0	0	0
високочастотних	0	0	0	0	0
ліній	0	0	0	0	0
зв'язку	0	0	0	0	0
мають	0	0	0	0	0
можливість	0	0	0	0	0
дозволити	0	0	0	0	0
забезпечити	0	0	0	0	0
досить	0	0	0	0	0
хороші	0	0	0	0	0
характеристики	0	0,0163362...	0	0	0
для	0	0,0180372...	0	0	0

Обробка тексту статті | Пошук збігу тестової статті за ключовими словами класів | Обрахунок оцінки приналежності обраної статті до класів

Виконати всі кроки визначення приналежності обраної статті до класів автоматично

Рисунок 3.5 – Вкладка «Класифікація статей»

Програмний код який реалізовує всі вище згадані модулі описаний в додатку Г. Таким чином, реалізовані програмні модулі системи.

3.3 Тестування інформаційної системи

Для визначення коректності виконання функцій автоматизованої інформаційної системи тематичної класифікації наукових статей, було проведено функціональне тестування. Для цього у відкритій програмі перевіряється функціональність роботи кнопок «Про програму» та «Вихід». Далі вводилась кількість класів класифікації починаючи від 2 закінчуючи 10 та кількість ключових слів для класів (від 10 до 500), і натискалась кнопка «Відкрити класифікатор», таким чином перевіряється чи відображається введена кількість класифікацій (рисунок 3.6). Ввівши 3 класи було отримано три класи, результат зображено на рисунку 3.7, відповідно результат для п'ятих класів зображено на рисунку 3.8.

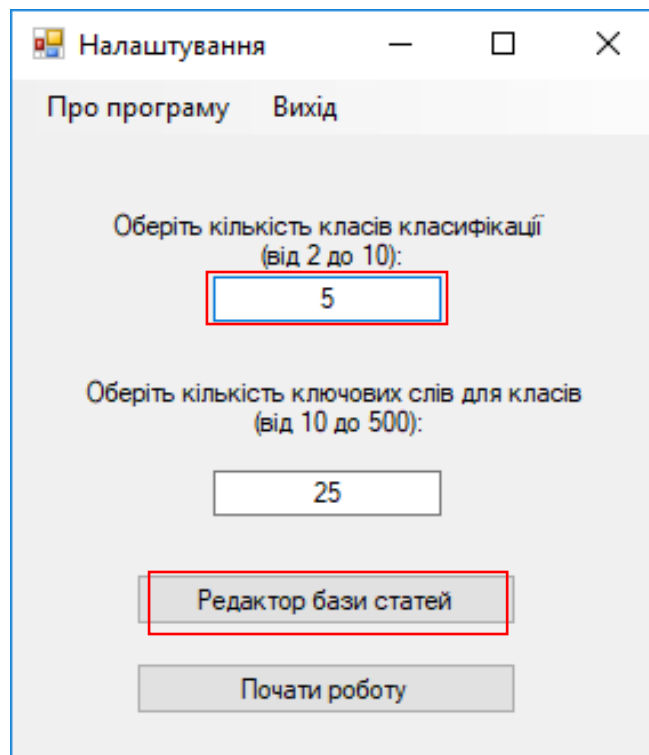


Рисунок 3.6 – Кнопки для тестування форми

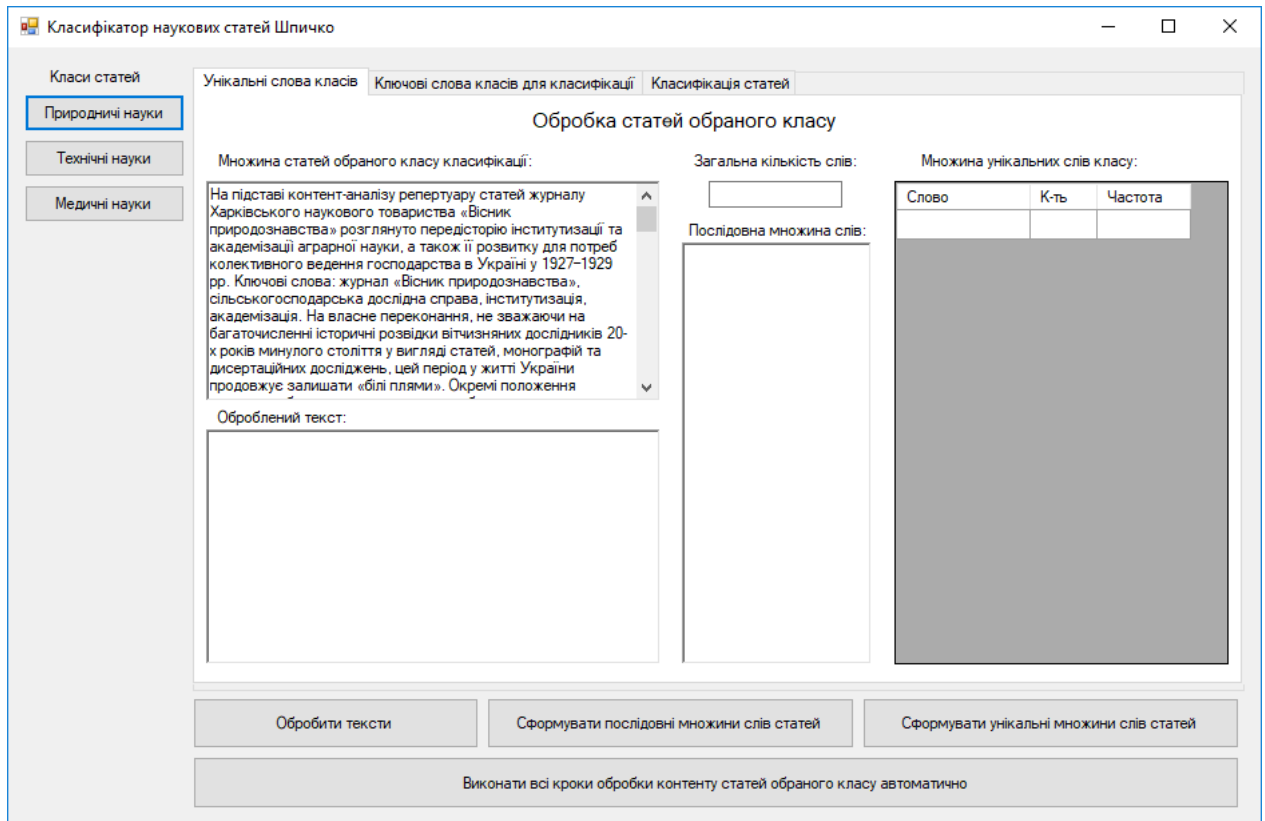


Рисунок 3.7 – Відображення 3-ох класифікацій

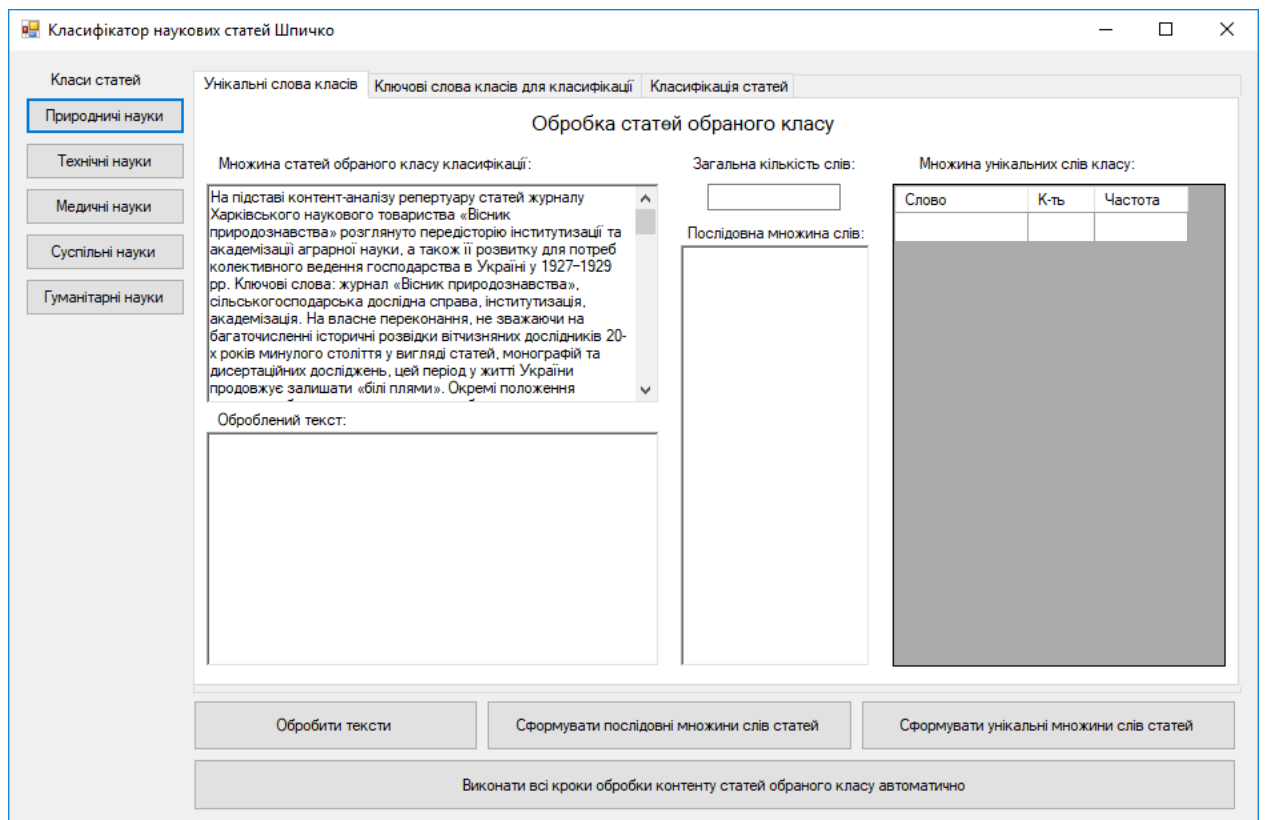


Рисунок 3.8 – Відображення 5-ох класифікацій

Для детальнішого дослідження коректності виконання функцій системи, було розроблено два тестових випадки (тест-кейси). У першому тестовому випадку (таблиця 3.1) перевіряється функціонал перегляду даних по натисканню на відповідну класифікацію системи. При натисканні на кнопки класифікацій (рисунки 3.9-3.10) у відповідному полі має відобразитись текст статті відповідної тематики, що береться з бази даних.

Таблиця 3.1 – Тест-кейс АВ0001

Тест-кейс ID: АВ0001	Пріоритет: 1	Створено: 03.06.2021, А.Шпичко
Назва: Перевірка відображення даних по натисканню на різні класифікації Вхідні дані: Класифікація = «Природничі науки», Класифікація = «Технічні науки»		
Кроки	Очікуваний результат	
<ol style="list-style-type: none"> 1. Запустити додаток 2. Ввести значення в поле кількості класифікацій та кількість ключових слів 3. Натиснути кнопку «Почати роботу» 4. Натиснути першу класифікацію «Природничі науки» 5. Порівняти фактичний результат з очікуваним 6. Натиснути другу класифікацію «Технічні науки» 7. Порівняти фактичний результат з очікуваним 	<p>Відображення даних класифікації «Природничі науки»</p> <p>Відображення даних класифікації «Технічні науки»</p>	
Результат виконання тест-кейсу: пройдено успішно		

У другому тестовому випадку (таблиця 3.2) перевіряється функціонал визначення класифікації тестової наукової статті. При переході на вкладку «Класифікація статей» та натисканні на кнопку «Виконати всі кроки визначення приналежності обраної статті до класів автоматично» у відповідних таблицях записуються обраховані дані по приналежності статті до кожного класу класифікації (рисунок 3.11). У кінцевому результаті в таблицю виводяться відсотки приналежності (рисунок 3.12).

Таблиця 3.2 – Тест-кейс АВ0002

Тест-кейс ID: АВ0002	Пріоритет: 1	Створено: А.Шпичко	03.06.2021,
Назва: Перевірка визначення класифікації тестової наукової статті			
Вхідні дані: Тестова стаття копіюється з бази даних			
Кроки		Очікуваний результат	
<ol style="list-style-type: none"> 1. Запустити додаток 2. Ввести значення в поле кількості класифікацій та кількість ключових слів 3. Натиснути кнопку «Почати роботу» 4. Перейти на вкладку «Класифікація статей» 5. Натиснути на кнопку «Виконати всі кроки визначення приналежності обраної статті до класів автоматично» 6. Порівняти фактичний результат з очікуваним 		Відображення обрахованих даних по приналежності статті до кожного класу класифікації та відсотків приналежності	
Результат виконання тест-кейсу: пройдено успішно			

Класифікатор наукових статей Шпичко

Класи статей: Природні науки, Технічні науки, Медичні науки, Суспільні науки, Гуманітарні науки

Визначення приналежності обраної статті до класів

Тестова стаття для класифікації:

Існуючі методи керування та виміру високочастотних ліній зв'язку мають можливість дозволити забезпечити досить хороші характеристики для контролю вимірюваних значень. Низькочастотна лінія в основному представлена коаксіальною лінією телефонної провідної лінії. Як і високочастотні лінії, ці провідні лінії також повинні контролюватися. Для вирішення проблеми виміру відстані пошкодження існує велика кількість приладів, заснованих на використанні одного з вищезазначених методів (як правило, імпульсного). Далекоміри, засновані на вимірюванні фазового зсуву, ще не знайшли широкого практичного застосування. Пристрої, які засновані на виявленні імпульсних ліній.

Оцінка приналежності обраної статті до класів:

Класифікації	Оцінка	Відсоток
Природні	0	0%
Технічні	0,0689659...	76,31%
Медичні	0,0084530...	9,35%
Суспільні	0	0%
Гуманітарні	0,0129629...	14,34%

Показники збігів тестової статті за ключовими словами класів:

	Природні	Технічні	Медичні	Суспільні	Гуманітарні
існуючі	0	0	0	0	0
методи	0	0	0	0	0
керування	0	0	0	0	0
та	0	0	0	0	0
виміру	0	0	0	0	0
високочастотних	0	0	0	0	0
ліній	0	0	0	0	0
зв'язку	0	0	0	0	0
мають	0	0	0	0	0
можливість	0	0	0	0	0
дозволити	0	0	0	0	0
забезпечити	0	0	0	0	0
досить	0	0	0	0	0
хороші	0	0	0	0	0
характеристики	0	0,0163362...	0	0	0
для	0	0,0180372...	0	0	0

Обробка тексту статті | Пошук збігу тестової статті за ключовими словами класів | Обрахунок оцінки приналежності обраної статті до класів

Виконати всі кроки визначення приналежності обраної статті до класів автоматично

Рисунок 3.11 – Відображення даних по приналежності статті до кожного класу

Класифікації	Оцінка	Відсоток
Природничі	0	0%
Технічні	0,0689659...	76,31%
Медичні	0,0084530...	9,35%
Суспільні	0	0%
Гуманітарні	0,0129629...	14,34%

Рисунок 3.12 – Відсотки приналежності тестової статті

Отже, в результаті тестування дійшли висновку що функції системи працюють коректно, а саме функції: вибір кількості класів для класифікації, вибір кількість ключових слів у класах, обробка тексту статей, формування послідовних множин слів статей, формування множин унікальних слів статей, автоматизований підбір альтернативних текстів, обрахунок значень TF-IDF унікальних слів класів, сортування множин ключових слів класів, обмеження множин ключових слів класів, перехід між тематичними класифікаціями, пошук збігів тестової статті за ключовими словами класів, обрахунок оцінки приналежності тестової статті до класів, візуалізація результатів роботи системи.

3.4 Інструкція користувача

Робота користувача з додатком розпочинається з головної форми, з аналогічною назвою – Головна (рисунок 3.13). На формі розташовується меню з пунктом «Про програму», з коротким описом програми та іменем автора, та пунктом «Вихід», який дозволяє вийти з програми. Головна функція даної форми це можливість обрати кількість тематичних класифікацій які будуть розглядатись в додатку далі. Обрати можна від 2 до 10 класифікацій, ввівши відповідне число та натиснувши на кнопку «Відкрити класифікатор». Далі відкривається нова форма – «Класифікатор наукових статей», на якій відображаються тематичні класи класифікації в тій кількості яку було обрано, та

множина статей першого класу (рисунок 3.14). Натискаючи на інші класи класифікації можна переглянути множину статей відповідного класу класифікації.

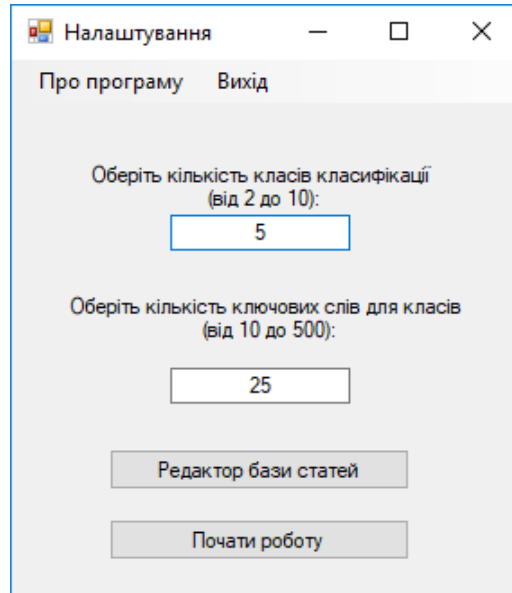


Рисунок 3.13 – Головна форма програми

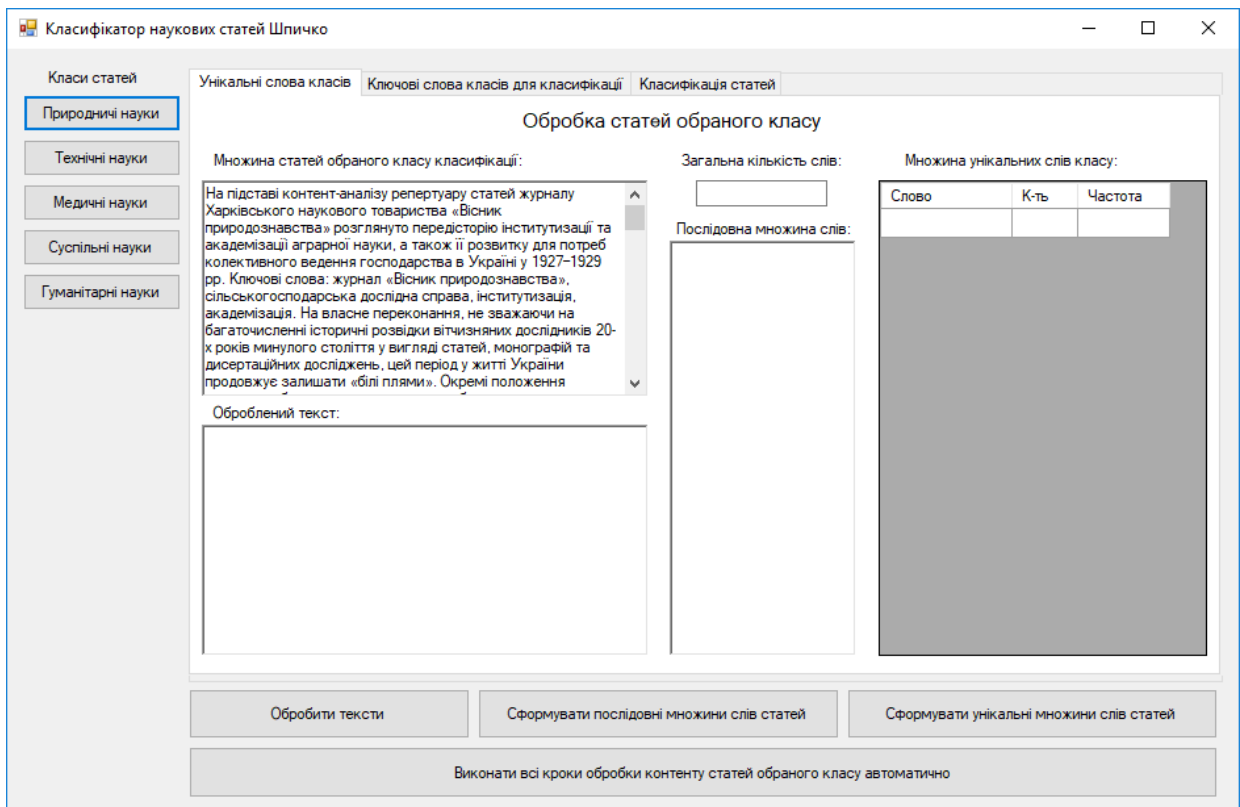


Рисунок 3.14 – Форма «Класифікатор наукових статей»

Щоб відобразити проміжні дані, користувачу потрібно натиснути на відповідний клас, а потім на кнопку «Виконати всі кроки обробки контенту статей обраного класу автоматично» (рисунки 3.15), або можна робити етапи відображення послідовно натискаючи відповідні кнопки.

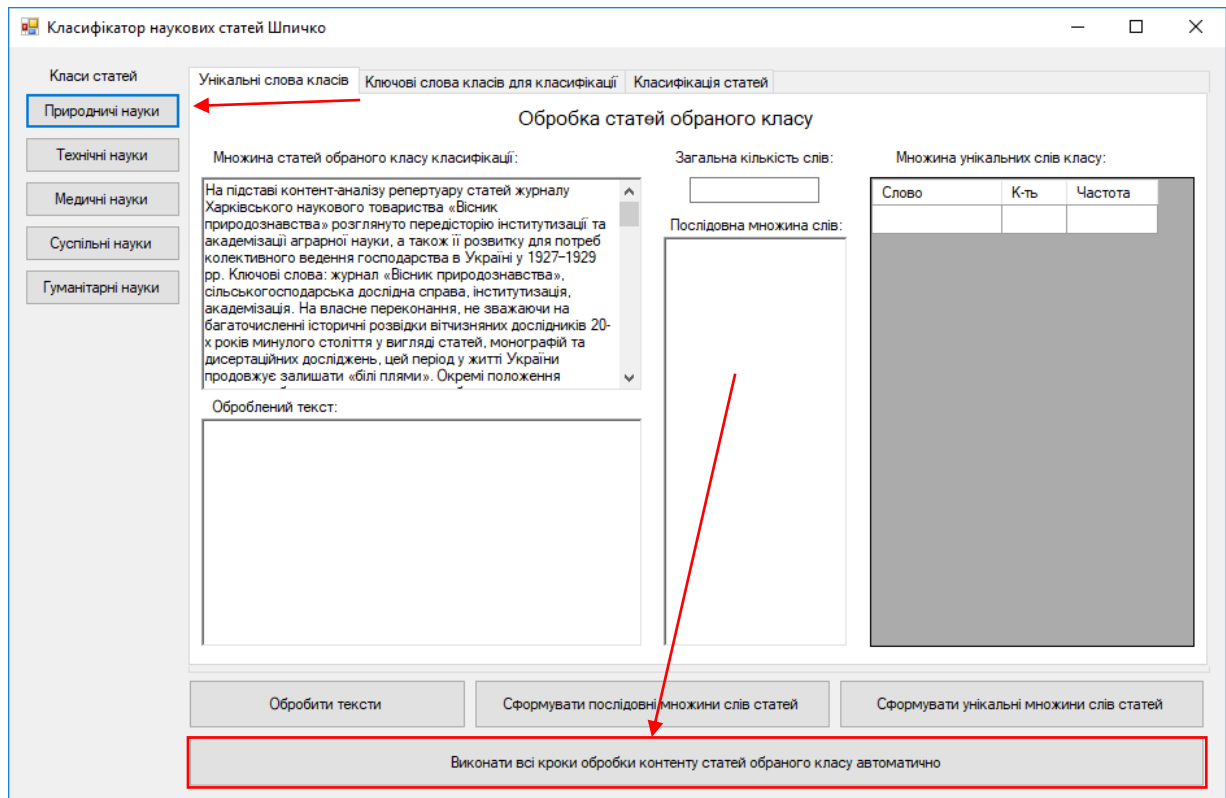


Рисунок 3.15 – Кнопка «Визначити»

Після виконання всіх розрахунків відображаються проміжні дані по класах класифікації (рисунки 3.16).

Користувач може виконувати і інші обрахунки перейшовши на вкладки: «Ключові слова класів для класифікації» та «Класифікація статей» (рисунки 3.17).

На вкладці «Ключові слова класів для класифікації», користувач може побачити альтернативні тексти, які обираються з інших класів, та TF, IDF, TF-IDF унікальних слів, які були визначені раніше (рисунки 3.18).

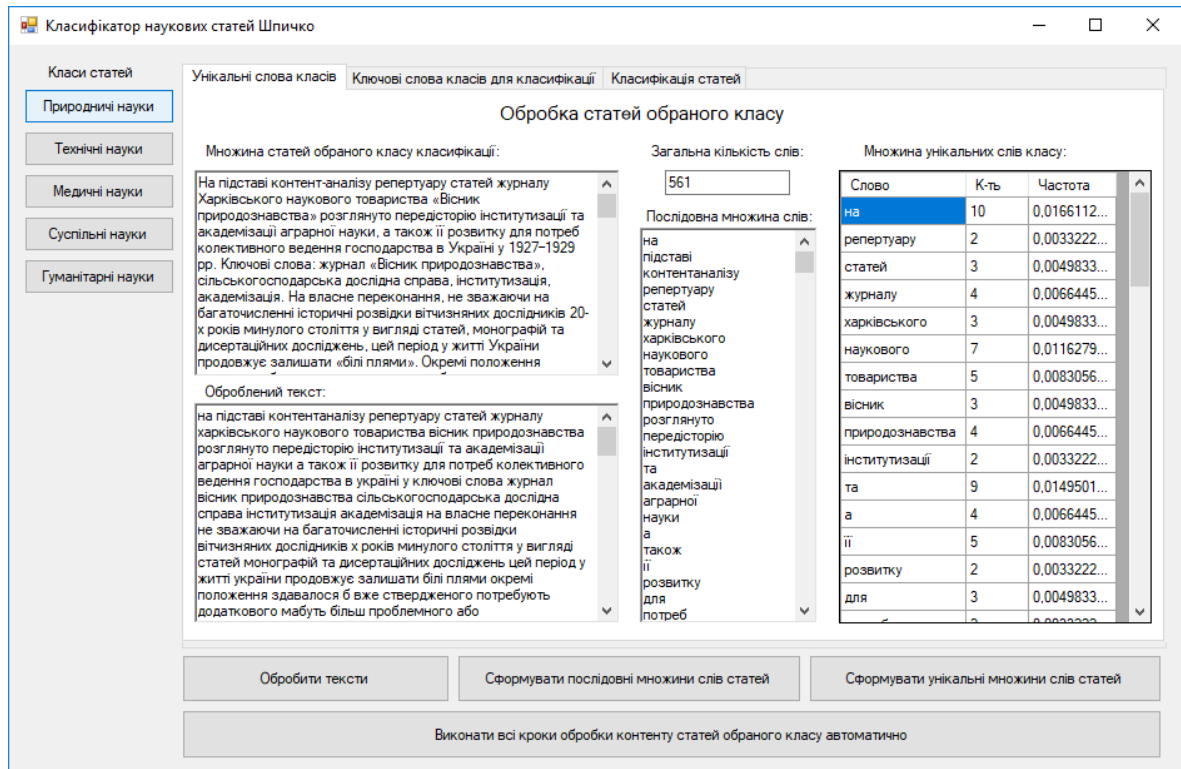


Рисунок 3.16 – Проміжні дані класифікації

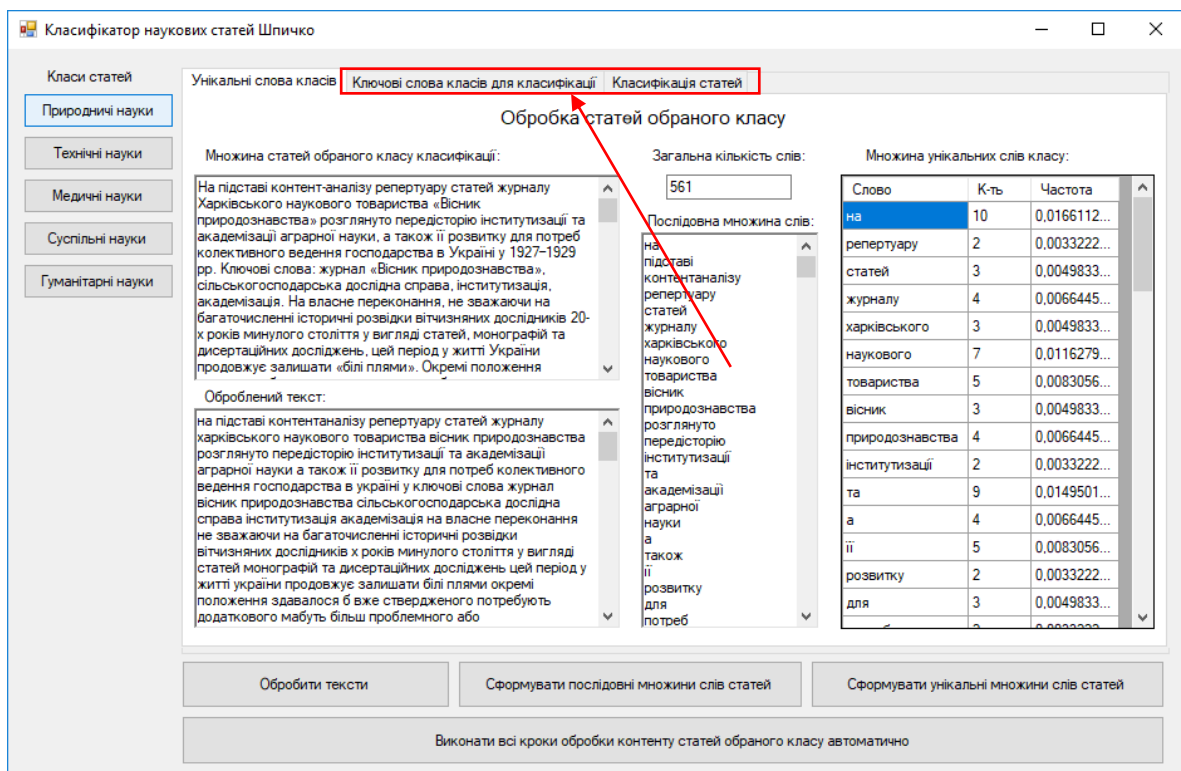


Рисунок 3.17 – Інші вкладки програми

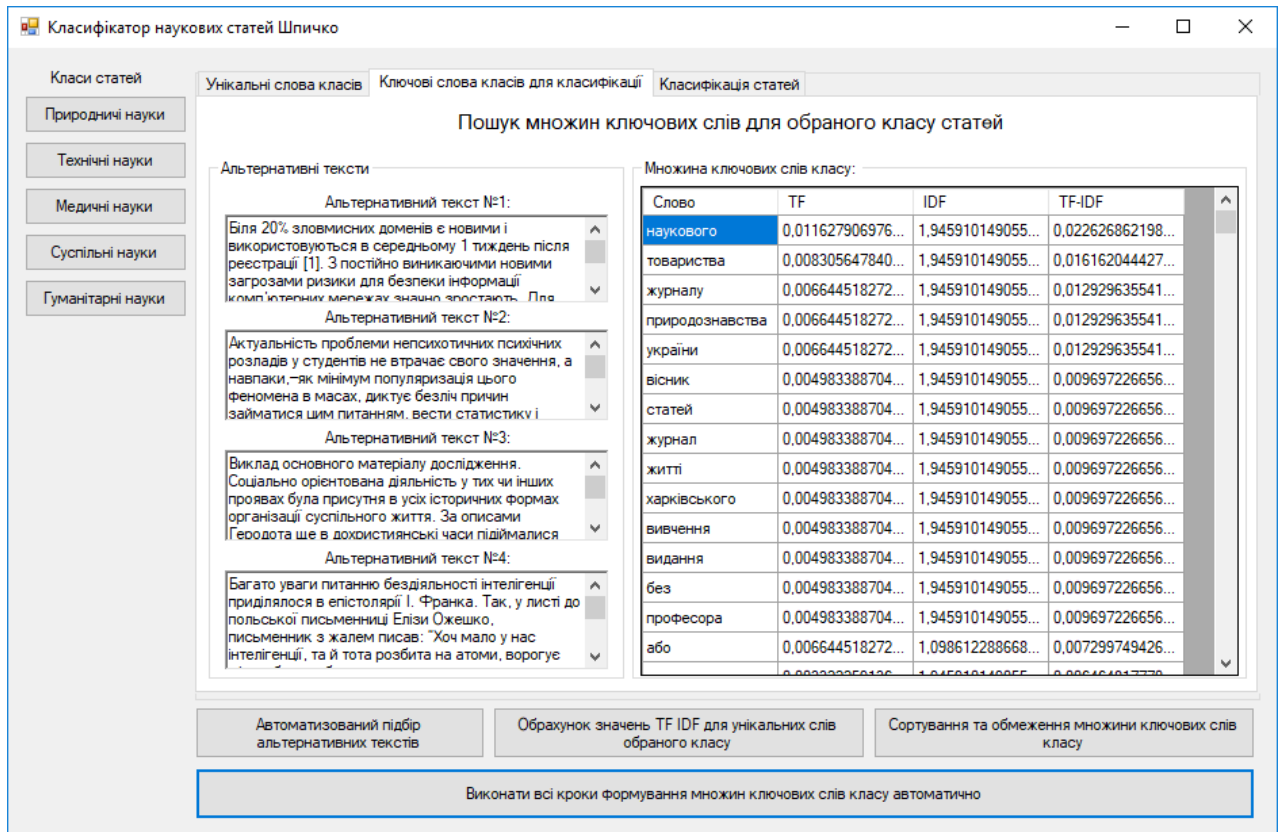


Рисунок 3.18 – Вкладка «Ключові слова класів для класифікації»

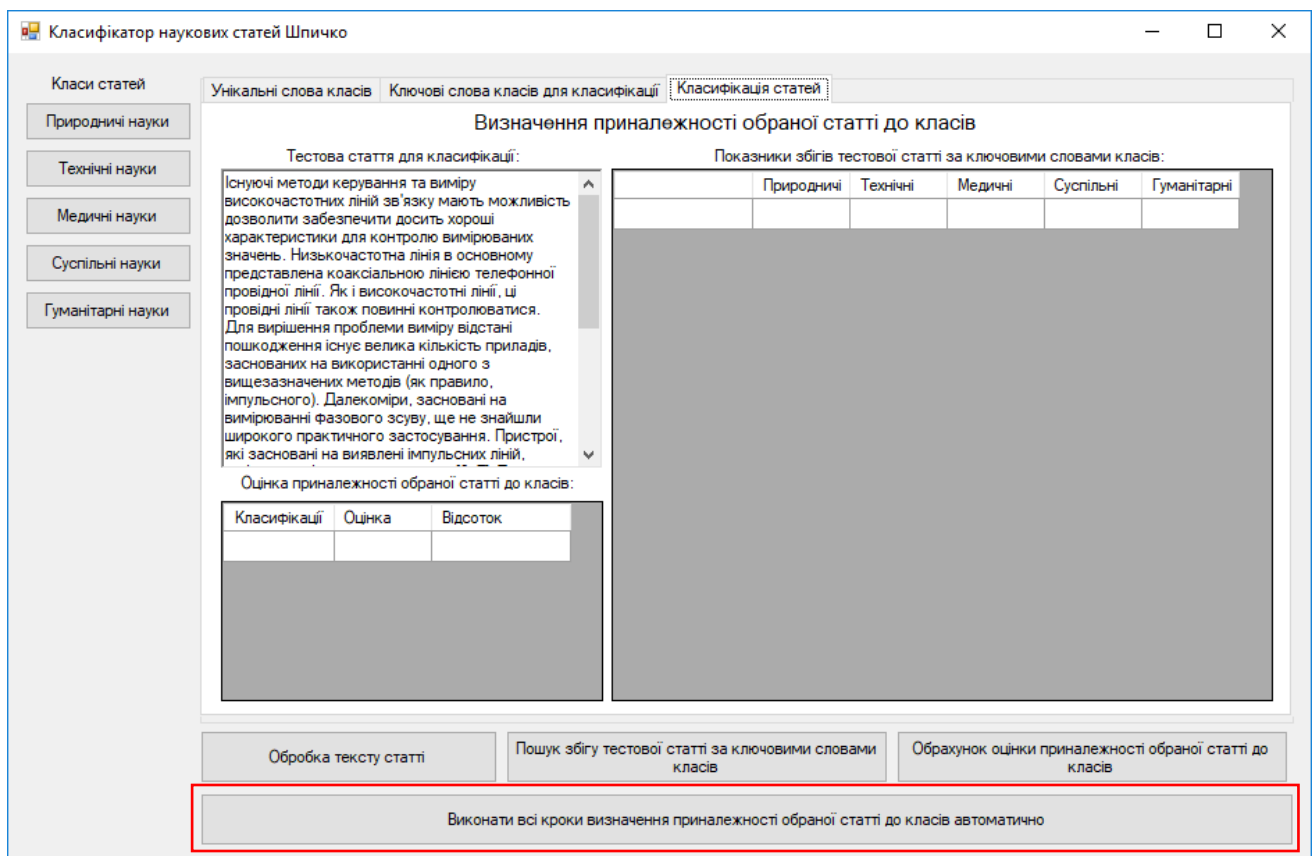


Рисунок 3.19 – Вкладка «Класифікація статей»

На вкладці «Класифікація статей», користувач має можливість визначити класифікацію тестової статті, натиснувши на кнопку «Виконати всі кроки визначення приналежності обраної статті до класів автоматично» (рисунок 3.19).

Після натискання кнопки у відповідних таблицях записуються обраховані дані по приналежності тестової статті до кожної класифікації та відсотки її приналежності (рисунок 3.20).

Класифікатор наукових статей Шпичко

Класи статей

Природні науки

Технічні науки

Медичні науки

Суспільні науки

Гуманітарні науки

Унікальні слова класів

Ключові слова класів для класифікації

Класифікація статей

Визначення приналежності обраної статті до класів

Тестова стаття для класифікації:

Існуючі методи керування та виміру високочастотних ліній зв'язку мають можливість дозволити забезпечити досить хороші характеристики для контролю вимірюваних значень. Низькочастотна лінія в основному представлена коаксіальною лінією телефонної провідної лінії. Як і високочастотні лінії, ці провідні лінії також повинні контролюватися. Для вирішення проблеми виміру відстані пошкодження існує велика кількість приладів, заснованих на використанні одного з вищезазначених методів (як правило, імпульсного). Далекоміри, засновані на вимірюванні фазового зсуву, ще не знайшли широкого практичного застосування. Пристрої, які засновані на виявленні імпульсних ліній.

Оцінка приналежності обраної статті до класів:

Класифікація	Оцінка	Відсоток
Природні	0	0%
Технічні	0,0689659...	76,31%
Медичні	0,0084530...	9,35%
Суспільні	0	0%
Гуманітарні	0,0129629...	14,34%

Показники збігів тестової статті за ключовими словами класів:

	Природні	Технічні	Медичні	Суспільні	Гуманітарні
існуючі	0	0	0	0	0
методи	0	0	0	0	0
керування	0	0	0	0	0
та	0	0	0	0	0
виміру	0	0	0	0	0
високочастотних	0	0	0	0	0
ліній	0	0	0	0	0
зв'язку	0	0	0	0	0
мають	0	0	0	0	0
можливість	0	0	0	0	0
дозволити	0	0	0	0	0
забезпечити	0	0	0	0	0
досить	0	0	0	0	0
хороші	0	0	0	0	0
характеристики	0	0,0163362...	0	0	0
для	0	0,0180372...	0	0	0

Обробка тексту статті

Пошук збігу тестової статті за ключовими словами класів

Обрахунок оцінки приналежності обраної статті до класів

Виконати всі кроки визначення приналежності обраної статті до класів автоматично

Рисунок 3.20 – Приналежність тестової статті до класифікацій

Таким чином, розроблений додаток реалізує функції: вибір кількості класів для класифікації, вибір кількості ключових слів у класах, обробка тексту статей, формування послідовних множин слів статей, формування множин унікальних слів статей, автоматизований підбір альтернативних текстів, обрахунок значень TF-IDF унікальних слів класів, сортування множин ключових слів класів, обмеження множин ключових слів класів, перехід між тематичними класифікаціями, пошук збігів тестової статті за ключовими словами класів,

обрахунок оцінки приналежності тестової статті до класів, візуалізація результатів роботи системи.

3.5 Вимоги до розгортання інформаційної системи

Мінімальні вимоги до апаратних засобів:

- Процесор: Intel Core I5.
- ОЗУ: 4 GB.
- Жорсткий диск: HDD (128GB).

Мінімальні вимоги до програмних засобів:

- Операційна система: Windows 7.
- СКБД MS SQL Server 2008.
- .NET Framework 4.5.

Висновки

В результаті виконання кваліфікаційної роботи бакалавра було розроблено автоматизовану інформаційну систему тематичної класифікації наукових статей. Розроблена система виконує такі основні функції:

- вибір кількості класів для класифікації;
- вибір кількість ключових слів у класах;
- обробка тексту статей;
- формування послідовних множин слів статей;
- формування множин унікальних слів статей;
- автоматизований підбір альтернативних текстів;
- обрахунок значень TF-IDF унікальних слів класів;
- сортування множин ключових слів класів;
- обмеження множин ключових слів класів;
- перехід між тематичними класифікаціями;
- пошук збігів тестової статті за ключовими словами класів;
- обрахунок оцінки приналежності тестової статті до класів;
- візуалізація результатів роботи системи.

Згідно проведеного тестування, створена автоматизована інформаційна система тематичної класифікації наукових статей відповідає поставленому завданню і виконує усі функції, які необхідно було реалізувати. Вхідними даними автоматизованої інформаційної системи тематичної класифікації наукових статей є: параметри роботи (кількість класів для класифікації та кількість ключових слів у класах), множина статей для кожного з класів та тестова стаття для аналізу; вихідними даними автоматизована інформаційна система тематичної класифікації наукових статей має: числова оцінка приналежності обраної статті до класів та таблиця збігів тестової статті за ключовими словами класів.

ІС може бути впроваджена в бібліотеки, навчальні заклади та інші електронні системи, що допоможе полегшити роботу з класифікацією наукових

статей. Програмний продукт також можна вдосконалювати, зокрема, забезпечити можливість врахування в оцінці приналежності тестової статті до класів оцінки семантичної важливості відповідних ключових слів у тестовій статті для підвищення кості класифікації.

Основні наукові та практичні результати доповідалися на VIII Міжнародній науково-практичній конференції «Інформаційні управляючі системи та технології ICST-ODESSA-2019», за темою «Методи автоматизованого визначення семантичних термінів у цифрових текстах» було виконано наукову публікацію [41]. В доповіді описувались методи, що використовуються і в створеному додатку. Також наукові та практичні результати доповідалися на XI Всеукраїнській науково-практичній конференції «Актуальні проблеми комп'ютерних наук АПКН-2019» було представлено публікацію на тему «Дослідження ефективності інформаційної технології тематичного сортування текстових повідомлень» [42]. При роботі над кваліфікаційною роботою бакалавра автором виконано 3 наукових публікації [41, 42, 43].

Перелік посилань

1. Порядок написання наукової статті. URL: https://pidru4niki.com/70368/buhgalterskiy_oblik_ta_audit/poryadok_napisannya_naukovoyi_statti.
2. Які бувають ключові слова і як їх гармонійно вписати в статтю? URL: <https://textum.com.ua/uk/blog/kakie-byvayut-klyuchevye-slova-i-kak-ih-garmonichno-vpisat-v-statyu/>.
3. Публікаційна активність як спосіб наукової комунікації та гонитви за рейтингами. С.94. URL: http://www.nas.gov.ua/publications/news/Documents/Visn_9-2017%2B11_Didenko.pdf.
4. Бібліографія. URL: http://esu.com.ua/search_articles.php?id=41911.
5. Бібліографія правила оформлення. С.5. URL: <https://physics.lnu.edu.ua/wp-content/uploads/bibliografia.pdf>.
6. Рубрикація тексту. URL: <https://studfile.net/preview/5585989/page:5/>.
7. Наукова публікація: поняття, функції, основні види. URL: <https://lektsii.org/12-26321.html>.
8. Робота над написанням наукових статей, монографій, наукових доповідей і повідомлень. URL: <https://studfile.net/preview/6872312/page:11/>.
9. Міжнародний науковий журнал «інтернаука» С.1. URL: <https://www.inter-nauka.com/uploads/public/15106525045803.pdf>.
10. Форми представлення результатів наукових досліджень. URL: <https://ppt-online.org/141875>.
11. Універсальна десяткова класифікація. URL: https://uk.wikipedia.org/wiki/Універсальна_десяткова_класифікація.
12. Наукова обробка документів. URL: http://www.biglib.com/book/78_Naykova_obrobka_dokumentiv/7624_363_Dokumentni_sistematchni_klasifikacii_yaki_vikoristovyutsya_v_Ukraini.

13. Термін «Галузь знань». URL: <https://zakon.rada.gov.ua/laws/term/33959>.
14. Класифікація наук. URL: <https://studfile.net/preview/6759588/page:3/>.
15. Поняття науки, її види і функції. URL: <https://studfile.net/preview/5470514/page:11/>.
16. Тематическая классификация статей новостного ресурса методами латентно-семантического анализа. URL: <https://www.top-technologies.ru/ru/article/view?id=36616>.
17. Наукометричні бази даних. URL: <https://library.pdpu.edu.ua/index.php/pro-nas/naukovtsiam/275-naukometrichni-bazi-danikh>.
18. Український індекс наукового цитування. URL: <http://uincit.uran.ua/>.
19. Наукометричні бази даних. URL: <http://www.nbuv.gov.ua/node/1367>.
20. Web of Science. URL: <https://www.webofknowledge.com/>.
21. Національна бібліотека України імені В. І. Вернадського. URL: https://uk.wikipedia.org/wiki/аціональна_бібліотека_України_імені_В._І._Вернадського.
22. Національна бібліотека України імені В. І. Вернадського. URL: <http://www.nbuv.gov.ua/>.
23. Java как язык и платформа. URL: <https://comaqa.gitbook.io/java-automation/java.-vvedenie/java-kak-yazyk-i-platforma>.
24. Java vs .net: почему .net. URL: <https://scientifically.info/news/2010-05-12-678>.
25. Можливості та технології Java. URL: <https://seo24.kiev.ua/rozrobka/rozrobka-android-kotlin-i-java/>.
26. Платформа .NET. URL: https://life-prog.ru/ukr/view_infosystem.php?id=27.
27. Розробка Android: Kotlin I Java. URL: <https://seo24.kiev.ua/rozrobka/rozrobka-android-kotlin-i-java/>.

28. Платформа .NET та її застосування для ООП. URL: <http://www.znannya.org/?view=csharp-dotNET>.
29. TF-IDF. URL: <https://ru.wikipedia.org/wiki/TF-IDF>.
30. Вступ в C#. URL: <https://programm.top/uk/c-sharp/tutorial/introduction/>.
31. Порівняння C++ з мовами Java і C# та майбутній розвиток мови. URL: <http://cpppm12.blogspot.com/2018/05/java-i-c.html>.
32. Обґрунтування вибору мови програмування. URL: https://studopedia.ru/15_59823_obruntuvannya-viboru-movi-programuvannya.html.
33. О. М. Спажев. Порівняльний аналіз найпопулярніших мов програмування в області штучного інтелекту, машинного навчання та глибинного навчання. URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2020/paper/download/10490/8923>.
34. Чому варто вивчати мову програмування c # (c sharp)? URL: <https://www.quality-assurance-group.com/chomu-varto-vyvchaty-movu-programuvannya-c-c-sharp/>.
35. Фреймворки в веб-розробці. URL: https://web-creator.ru/articles/about_frameworks.
36. Переваги Microsoft .NET Framework. URL: <https://support.microsoft.com/uk-ua/help/929300/benefits-of-the-microsoft-net-framework>.
37. .NET Core. URL: https://ru.wikipedia.org/wiki/.NET_Core.
38. Основне призначення системи керування базами даних. URL: <https://studepedia.org/index.php?vol=1&post=104628>.
39. Microsoft SQL Server. URL: https://uk.wikipedia.org/wiki/Microsoft_SQL_Server#SQL_Server_2012.
40. Різниця між SQL vs MySQL vs SQL Server. URL: <https://uk.myservername.com/difference-between-sql-vs-mysql-vs-sql-server>.
41. Шпичко А. В., Мазурець О. В. Методи автоматизованого визначення семантичних термінів у цифрових текстах. Інформаційні управляючі системи та технології ICST-ODESSA-2019. 2019. С.166-168.

42. О. Ю. Тимуш, А. В. Шпичко, О. В. Мазурець Дослідження ефективності інформаційної технології тематичного сортування текстових повідомлень. Збірник наукових праць за матеріалами XI всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2019». 2019. № 1. С.207-212.

43. О. М. Чугай, А. В. Шпичко, О. В. Мазурець Інформаційна модель кіберспортивної команди для автоматизованого формування складу команд. Збірник наукових праць за матеріалами XII Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2020». 2020. С.339-344.

ДОДАТКИ

Додаток А
Класифікація статей за галуззю знань

Таблиця А.1 - Класифікація статей за галуззю знань МОН [13]

Шифр галузі	Галузь знань
01	Освіта
02	Культура і мистецтво
03	Гуманітарні науки
04	Богослов'я
05	Соціальні та поведінкові науки
06	Журналістика
07	Управління та адміністрування
08	Право
09	Біологія
10	Природничі науки
11	Математика та статистика
12	Інформаційні технології
13	Механічна інженерія
14	Електрична інженерія
15	Автоматизація та приладобудування
16	Хімічна та біоінженерія
17	Електроніка та телекомунікації
18	Виробництво та технології
19	Архітектура та будівництво
20	Аграрні науки та продовольство
21	Ветеринарна медицина
22	Охорона здоров'я
23	Соціальна робота
24	Сфера обслуговування
25	Воєнні науки, національна безпека, безпека державного кордону
26	Цивільна безпека
27	Транспорт

Додаток Б

Автоматизація бізнес-процесів інформаційної системи тематичної класифікації наукових статей

При розробці автоматизованої інформаційної системи тематичної класифікації наукових статей необхідно автоматизувати наступні бізнес-процеси:

- бізнес-процес «Робота з тематичними класифікаціями»;
- бізнес-процес «Робота з унікальними словами класифікації»;
- бізнес-процес «Робота з ключовими словами класифікації»;
- бізнес-процес «Робота з тестовою науковою статтею»;

Бізнес-процес «Робота з тематичними класифікаціями». Система повинна надавати доступ до тематичних класифікацій, таких як:

- природничі науки;
- технічні науки;
- медичні науки;
- суспільні науки;
- гуманітарні науки.

Доступ користувача до цієї інформації відбувається наступним чином:

1. Користувач обирає кількість класифікацій та відкриває сторінку «Класифікатор наукових статей» і бачить там наукові статті класифікації, яка обрана за замовчуванням («Природничі науки»).

2. Користувач натискає на назву іншої класифікації - «Технічні науки» та бачить наукові статті відповідної класифікації, що є в системі.

Діаграму дій для даного бізнес-процесу зображено на рисунку Б.1.



Рисунок Б.1 – Діаграма дій бізнес-процесу «Робота з тематичними класифікаціями».

Бізнес-процес «Робота з унікальними словами класифікації». Система повинна надавати доступ до множини наукових статей, що зберігаються в базі даних та можливість користувачу визначати її унікальні слова.

Робота користувача з множиною наукових статей та множиною унікальних слів відбувається наступним чином:

1. Користувач відкриває сторінку «Класифікатор наукових статей» та бачить там множину наукових статей класифікації, що береться з бази даних.

2. Користувач натискає на кнопку «Визначити унікальні слова» та бачить визначену послідовність множини слів, оброблений текст наукових статей і визначену множину унікальних слів з обрахованою частотою використання слова в множині статей.

Діаграму дій для даного бізнес-процесу зображено на рисунку Б.2.

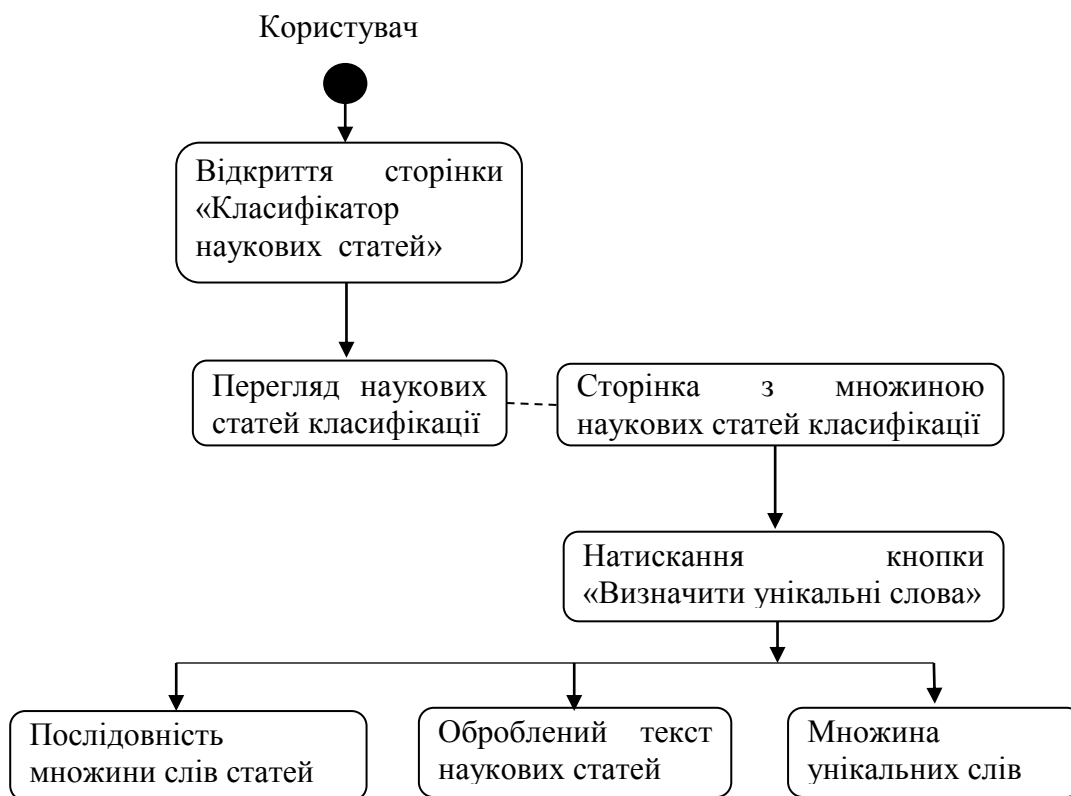


Рисунок Б.2 – Діаграма дій бізнес-процесу «Робота з унікальними словами класифікації».

Бізнес-процес «Робота з ключовими словами класифікації». Система повинна надавати доступ до альтернативних текстів наукових статей обраної класифікації, що зберігаються в базі даних та можливість користувачу визначити множину ключових слів.

Робота користувача з альтернативними текстами та множиною ключових слів відбувається наступним чином:

1. Користувач відкриває вкладку «Ключові слова наукових статей» та бачить там альтернативні тексти наукових статей, що береться з бази даних.

2. Користувач натискає на кнопку «Визначити та обмежити» та бачить визначену, відсортовану та обмежену послідовність ключових слів.

Діаграму дій для даного бізнес-процесу зображено на рисунку Б.3.



Рисунок Б.3 – Діаграма дій бізнес-процесу «Робота з ключовими словами класифікації»

Бізнес-процес «Робота з тестовою науковою статтею». Система повинна надавати користувачу можливість завантажувати файл з тестовою статтею та визначати класифікацію завантаженої статті.

Робота користувача з тестовою новиною відбувається наступним чином:

1. Користувач відкриває вкладку «Визначення класифікацій статті».
2. Використовуючи інтерфейс вкладки, завантажує файл з тестовою статтею.

3. Користувач натискає на кнопку «Визначити класифікацію» та бачить множину визначених ключових слів тестової статті, кожен класифікацію та приналежність слова до неї, результат обчислення визначення класифікації а також, класифікацію до якої найбільше підходить дана тестова стаття.

Діаграму дій для даного бізнес-процесу зображено на рисунку Б.4.

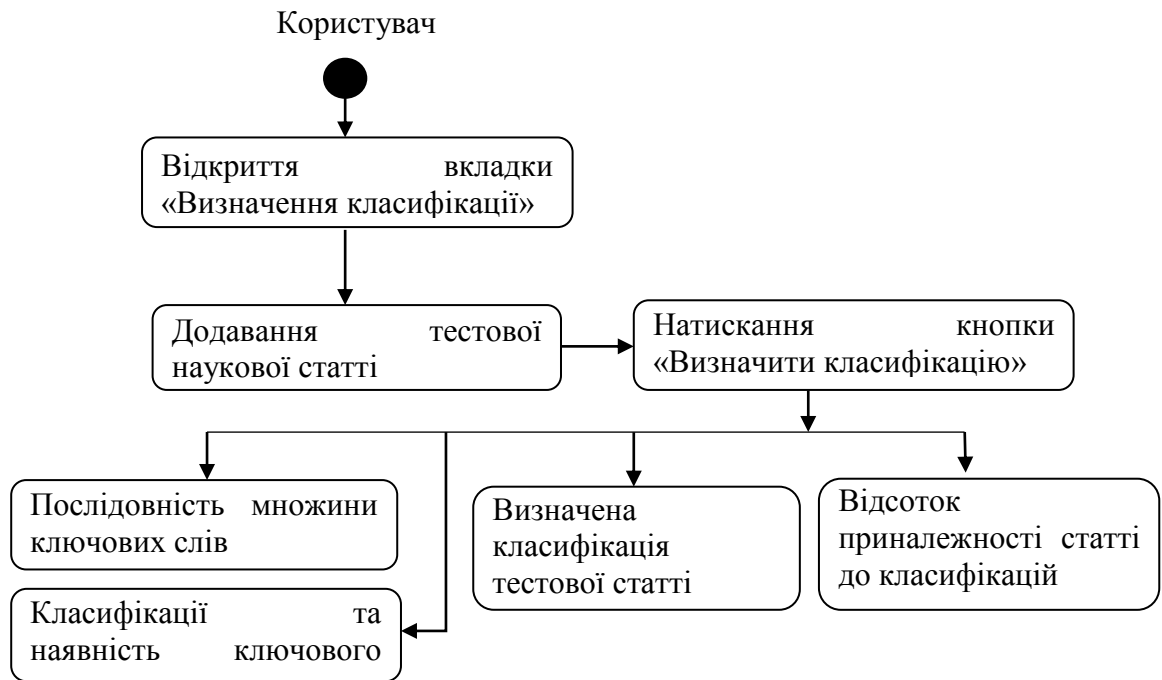


Рисунок Б.4 – Діаграма дій бізнес-процесу «Робота з тестовою науковою статтею».

Таким чином бізнес-процеси реалізують функції, що були описані в постановці завдання.

Додаток В

Програмні коди основних модулів автоматизованої інформаційної системи тематичної класифікації наукових статей

Лістинг Form1.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace klasyfikator
{
    public partial class Form1 : Form
    {
        int k,z;
        public Form1()
        {
            InitializeComponent();
        }

        private void вихідToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            z = Convert.ToInt32(textBox2.Text);
            k = Convert.ToInt32(textBox1.Text);
            if (k < 2 || k > 5)
            {
                MessageBox.Show("Введіть значення в межах від 2 до 5");
            }
            else
            {
                Form2 f2 = new Form2(k,z);
                f2.Show();
            }
        }
    }
}
```

Лістинг Form2.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using DocumentFormat.OpenXml.Packaging;
using DocumentFormat.OpenXml.Wordprocessing;
using System.Data.SqlClient;
using klasyfikator.Models;

namespace klasyfikator
{
    public partial class Form2 : Form
    {
        #region data
```

```

string[] mas = { "\", "-" };

List<string> text1 = new List<string>();
List<string> text2 = new List<string>();
List<string> text3 = new List<string>();
List<string> text4 = new List<string>();
List<string> text5 = new List<string>();

List<string> text_1 = new List<string>();
List<string> text_2 = new List<string>();
List<string> text_3 = new List<string>();
List<string> text_4 = new List<string>();
List<string> text_5 = new List<string>();

List<string> test_text = new List<string>();

string nazva;

Text t = new Text();
Methods Met = new Methods();
Sum Count = new Sum();
Classification Kategor = new Classification();

string[] mas_text;
string text_new = "";
int zagal_kilk = 0;
int number_vk1 = 1;
int len_gr;
int k = 0;
string text;
int[] index;
string[,] mas_ob_;
double[] rezerv;
double temp; int temp_ind;

string[] nowy_mas_test;
string[] nowy_mas;
string[] nowy_mas_2;
string[] nowy_mas_3;
string[] nowy_mas_4;
string[] nowy_mas_5;

int[] mas_kilk_;
int[] mas_kilk;
int[] mas_kilk_2;
int[] mas_kilk_3;
int[] mas_kilk_4;
int[] mas_kilk_5;

string[,] mas_ob_1;
string[,] mas_ob_2;
string[,] mas_ob_3;
string[,] mas_ob_4;
string[,] mas_ob_5;

int len = 0, i_len = 0;

string[] mas1;
double[] chast_;
string[] mas2;
double[] chast_2;
int[] mas_kilk2;
string[] mas3;
double[] chast_3;
int[] mas_kilk3;
string[] mas4;
double[] chast_4;
int[] mas_kilk4;
string[] mas5;
double[] chast_5;
int[] mas_kilk5;

int[] kilk_text;
string txt1, txt2, txt3, txt4, txt5, txt6;
string[] mas_text2;
string[] mas_text3;
string[] mas_text4;
string[] mas_text5;

```

```

double[] chast;
double[] chast2;
double[] chast3;
double[] chast4;
double[] chast5;

double[] idf1;
double[] tf_idf1;

double[] idf2;
double[] tf_idf2;

double[] idf3;
double[] tf_idf3;

double[] idf4;
double[] tf_idf4;

double[] idf5;
double[] tf_idf5;
int len_for_kat = 30;

#endregion
public void text_new_txt(string nazva)
{
nazva + ".docx", true);
    WordprocessingDocument wordDoc = WordprocessingDocument.Open(@Environment.CurrentDirectory + "/" +
    Body body = wordDoc.MainDocumentPart.Document.Body;
    var qwe = body.ChildElements;
    foreach (var child_element in qwe)
    {
        var asdf = child_element.InnerText;
        File.AppendAllText(@nazva + ".txt", '\n' + asdf);
    }
}
public List<string> txt_stati(string nazva, List<string> lines)
{
    string[] b = File.ReadAllText(@nazva + ".txt", Encoding.UTF8).
        Split(new Char[] { '\n', ' ' }, StringSplitOptions.RemoveEmptyEntries);

    for (int i = 0; i < b.Length; i++)
    {
        lines.Add(b[i]);
    }
    return lines;
}

public Form2(int k, int z)
{
    InitializeComponent();

    ClassifierEntities context = new ClassifierEntities();
    var text = context.THEM_TEXT;
    foreach (var i in text)
    {
        if (i.ClassificationId == 1)
            text_1.Add(i.Text);

        else if (i.ClassificationId == 2)
            text_2.Add(i.Text);

        else if (i.ClassificationId == 3)
            text_3.Add(i.Text);

        else if (i.ClassificationId == 4)
            text_4.Add(i.Text);

        else if (i.ClassificationId == 5)
            text_5.Add(i.Text);
    }

    #region button
    if (k == 5)
    {
        button3.Visible = true;
    }
}

```

```

        button4.Visible = true;
        button5.Visible = true;
    }
    else if (k == 4)
    {
        button3.Visible = true;
        button4.Visible = true;
    }
    else if (k == 3)
    {
        button3.Visible = true;
    }
    }
    #endregion

    button7.Visible = false;
    button8.Visible = false;

    for (int i = 0; i < text_1.Count; i++)
        richTextBox1.AppendText(text_1[i].ToString() + " ");
}

private void button1_Click(object sender, EventArgs e)
{
    number_vk1 = 1;
    richTextBox1.Clear();
    for (int i = 0; i < text_1.Count; i++)
        richTextBox1.AppendText(text_1[i].ToString() + " ");
    richTextBox4.Clear();
    richTextBox5.Clear();
    richTextBox6.Clear();
    richTextBox7.Clear();

    for (int i = 0; i < text_2.Count; i++)
        richTextBox4.AppendText(text_2[i].ToString() + " ");

    for (int i = 0; i < text_3.Count; i++)
        richTextBox5.AppendText(text_3[i].ToString() + " ");

    for (int i = 0; i < text_4.Count; i++)
        richTextBox6.AppendText(text_4[i].ToString() + " ");

    for (int i = 0; i < text_5.Count; i++)
        richTextBox7.AppendText(text_5[i].ToString() + " ");
}

private void button2_Click(object sender, EventArgs e)
{
    number_vk1 = 2;
    richTextBox1.Clear();
    richTextBox4.Clear();
    richTextBox5.Clear();
    richTextBox6.Clear();
    richTextBox7.Clear();

    for (int i = 0; i < text_2.Count; i++)
        richTextBox1.AppendText(text_2[i].ToString() + " ");

    for (int i = 0; i < text_1.Count; i++)
        richTextBox4.AppendText(text_1[i].ToString() + " ");

    for (int i = 0; i < text_3.Count; i++)
        richTextBox5.AppendText(text_3[i].ToString() + " ");

    for (int i = 0; i < text_4.Count; i++)
        richTextBox6.AppendText(text_4[i].ToString() + " ");

    for (int i = 0; i < text_5.Count; i++)
        richTextBox7.AppendText(text_5[i].ToString() + " ");
}

private void tabControl1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (tabControl1.SelectedTab.Name == "tabPage2") { button6.Visible = false; button7.Visible = true;
    button8.Visible = false;

```

```

        button9.Text = "Автоматизований підбір альтернативних текстів";
        button10.Text = "Обрахунок значень TF IDF для унікальних слів обраного класу";
        button11.Text = "Сортування та обмеження множини ключових слів класу";
    }
    if (tabControl1.SelectedTab.Name == "tabPage1") { button6.Visible = true; button7.Visible = false;
button8.Visible = false;
        button9.Text = "Обробити тексти";
        button10.Text = "Сформувати послідовні множини слів статей";
        button11.Text = "Сформувати унікальні множини слів статей";
    }
    if (tabControl1.SelectedTab.Name == "tabPage3") { button6.Visible = false; button7.Visible = false;
button8.Visible = true;
        button9.Text = "Обробка тексту статті";
        button10.Text = "Пошук збігу тестової статті за ключовими словами класів";
        button11.Text = "Обрахунок оцінки приналежності обраної статті до класів";
    }
}

public void grid_vyvod(string[,] mas_ob_mas, int len_gr)
{
    dataGridView3.RowCount = len_gr;
    for (int i = 0; i < len_gr; i++)
    {
        dataGridView3.Rows[i].Cells[0].Value = mas_ob_mas[i, 0];
        dataGridView3.Rows[i].Cells[1].Value = mas_ob_mas[i, 1];
        dataGridView3.Rows[i].Cells[2].Value = mas_ob_mas[i, 2];
        dataGridView3.Rows[i].Cells[3].Value = mas_ob_mas[i, 3];
    }
}

private void button7_Click(object sender, EventArgs e)
{
    #region TF-IDF kat1

    kilk_text = new int[nowy_mas.Length];
    txt1 = richTextBox4.Text.ToLower();
    txt2 = richTextBox5.Text.ToLower();
    txt3 = richTextBox6.Text.ToLower();
    txt4 = richTextBox7.Text.ToLower();

    text_new = "";
    mas_text2 = t.txt1(txt1, mas, text_new);
    text_new = "";
    mas_text3 = t.txt1(txt2, mas, text_new);
    text_new = "";
    mas_text4 = t.txt1(txt3, mas, text_new);
    text_new = "";
    mas_text5 = t.txt1(txt4, mas, text_new);
    text_new = "";

    idf1 = new double[nowy_mas.Length];
    t.kilkist_textiv(nowy_mas, mas_text2, mas_text3, mas_text4, mas_text5, kilk_text);

    tf_idf1 = new double[nowy_mas.Length];
    Met.tf_idf(nowy_mas, kilk_text, idf1);

    #endregion
    #region TF-IDF kat2

    kilk_text = new int[nowy_mas_2.Length];
    idf2 = new double[nowy_mas_2.Length];
    t.kilkist_textiv(nowy_mas_2, mas_text2, mas_text3, mas_text4, mas_text5, kilk_text);

    tf_idf2 = new double[nowy_mas_2.Length];
    Met.tf_idf(nowy_mas_2, kilk_text, idf2);

    #endregion
    #region TF-IDF kat3

    kilk_text = new int[nowy_mas_3.Length];
    idf3 = new double[nowy_mas_3.Length];
    t.kilkist_textiv(nowy_mas_3, mas_text2, mas_text3, mas_text4, mas_text5, kilk_text);

    tf_idf3 = new double[nowy_mas_3.Length];
    Met.tf_idf(nowy_mas_3, kilk_text, idf3);
}

```

```

#endregion
#region TF-IDF kat4

kilk_text = new int[nowy_mas_4.Length];
idf4 = new double[nowy_mas_4.Length];
t.kilkist_textiv(nowy_mas_4, mas_text2, mas_text3, mas_text4, mas_text5, kilk_text);

tf_idf4 = new double[nowy_mas_4.Length];
Met.tf_idf(nowy_mas_4, kilk_text, idf4);

#endregion
#region TF-IDF kat5

kilk_text = new int[nowy_mas_5.Length];
idf5 = new double[nowy_mas_5.Length];
t.kilkist_textiv(nowy_mas_5, mas_text2, mas_text3, mas_text4, mas_text5, kilk_text);

tf_idf5 = new double[nowy_mas_5.Length];
Met.tf_idf(nowy_mas_5, kilk_text, idf5);

#endregion
mas_ob_1 = grid(nowy_mas, chast, idf1, tf_idf1);
mas_ob_2 = grid(nowy_mas_2, chast2, idf2, tf_idf2);
mas_ob_3 = grid(nowy_mas_3, chast3, idf3, tf_idf3);
mas_ob_4 = grid(nowy_mas_4, chast4, idf4, tf_idf4);
mas_ob_5 = grid(nowy_mas_5, chast5, idf5, tf_idf5);

if (number_vkl == 1)
{
    len_gr = nowy_mas.Length;
    grid_vyvod(mas_ob_1, len_gr);
}
else if(number_vkl == 2)
{
    len_gr = nowy_mas_2.Length;
    grid_vyvod(mas_ob_2, len_gr);
}
}

//запис tf_idf 2 вкладка
public string[,] grid(string[] nowy_mas, double[] chast_i, double[] idf, double[] tf_idf)
{
    mas_ob_ = new string[nowy_mas.Length, 4];
    index = new int[nowy_mas.Length];
    rezerv = new double[tf_idf.Length];
    for (int i = 0; i < tf_idf.Length; i++)
    {
        tf_idf[i] = chast_i[i] * idf[i];
        rezerv[i] = tf_idf[i];
        index[i] = i;
    }
    //сортування
    for (int j = 0; j < rezerv.Length - 1; j++)
    {
        for (int i = j + 1; i < rezerv.Length; i++)
        {
            if (rezerv[j] * (-1) > rezerv[i] * (-1))
            {
                temp = rezerv[j];
                temp_ind = index[j];

                rezerv[j] = rezerv[i];
                index[j] = index[i];

                rezerv[i] = temp;
                index[i] = temp_ind;
            }
        }
    }
    for (int i = 0; i < rezerv.Length; i++)
    {
        mas_ob_[i, 0] = nowy_mas[index[i]];
        mas_ob_[i, 1] = chast_i[index[i]].ToString();
        mas_ob_[i, 2] = idf[index[i]].ToString();
        mas_ob_[i, 3] = rezerv[i].ToString();
    }
    return mas_ob_;
}

```

```

}
public void classif(string text, string[] mas_text, string text_new)
{
    text_new = "";

    test_text.Clear();

    k = 0;
    for (int i = 0; i < mas_text.Length; i++)
    {
        test_text.Add(mas_text[i]);
        mas_text[i] = "";
    }
    for (int i = 0; i < test_text.Count; i++)
    {
        if (test_text[i] == "\n" || test_text[i] == " " || test_text[i] == "" ||
string.IsNullOrEmpty(test_text[i]))
        {
            test_text.Remove(test_text[i]);
        }
        else
        {
            mas_text[k] = test_text[i];
            k++;
        }
    }
}
private void button8_Click(object sender, EventArgs e)
{
    text = richTextBox15.Text.ToLower();

    mas_text = t.txt1(text, mas, text_new);
    classif(text, mas_text, text_new);

    int[] mas_kilk_test = new int[mas_text.Length];
    string[] mas_slova = new string[mas_text.Length];

    mas_slova[0] = mas_text[0];
    zagal_kilk = 0;

    nowy_mas_test = t.mas(mas_text, mas_slova, mas_kilk_test);

    dataGridView8.RowCount = nowy_mas_test.Length + 1;
    dataGridView9.RowCount = 5;
    double[] znach = new double[nowy_mas_test.Length];
    double[] znach_2 = new double[nowy_mas_test.Length];
    int z = 1;
    Random rand = new Random();

    for (int i = 0; i < nowy_mas_test.Length; i++)
    {
        dataGridView8.Rows[i].Cells[0].Value = nowy_mas_test[i];
        z = 1;
        Kategor.klas_statti(dataGridView8, len_for_kat, mas_ob_1, mas_kilk_test, i, tf_idf1, z,
nowy_mas_test);
        z = 2;
        Kategor.klas_statti(dataGridView8, len_for_kat, mas_ob_2, mas_kilk_test, i, tf_idf2, z,
nowy_mas_test);
        z = 3;
        Kategor.klas_statti(dataGridView8, len_for_kat, mas_ob_3, mas_kilk_test, i, tf_idf3, z,
nowy_mas_test);
        z = 4;
        Kategor.klas_statti(dataGridView8, len_for_kat, mas_ob_4, mas_kilk_test, i, tf_idf4, z,
nowy_mas_test);
        z = 5;
        Kategor.klas_statti(dataGridView8, len_for_kat, mas_ob_5, mas_kilk_test, i, tf_idf5, z,
nowy_mas_test);

        zagal_kilk += mas_kilk_test[i];
    }

    dataGridView8.Rows[nowy_mas_test.Length].Cells[0].Value = "Cuma";
    double s = 0;
    int q = 1;

```

```

for (int i = 0; i < 5; i++)
{
    Count.Summ(dataGridView8, s, nowy_mas_test, q);
    s = 0; q++;
}

dataGridView9.Rows[0].Cells[0].Value = Column6.HeaderText;
dataGridView9.Rows[1].Cells[0].Value = Column7.HeaderText;
dataGridView9.Rows[2].Cells[0].Value = Column8.HeaderText;
dataGridView9.Rows[3].Cells[0].Value = Column9.HeaderText;
dataGridView9.Rows[4].Cells[0].Value = Column10.HeaderText;

double sum=0;
for (int i = 0; i < 5; i++)
{
    sum += Convert.ToDouble(dataGridView8.Rows[nowy_mas_test.Length].Cells[i + 1].Value);
    dataGridView9.Rows[i].Cells[1].Value = dataGridView8.Rows[nowy_mas_test.Length].Cells[i +
1].Value;
}
for (int i = 0; i < 5; i++)
{
    dataGridView9.Rows[i].Cells[2].Value
Math.Round(Convert.ToDouble(dataGridView9.Rows[i].Cells[1].Value) / sum * 100,2) + "%";
}
}

private void button3_Click(object sender, EventArgs e)
{
    richTextBox1.Clear();
    for (int i = 0; i < text_3.Count; i++)
        richTextBox1.AppendText(text_3[i].ToString() + " ");
}

private void button4_Click(object sender, EventArgs e)
{
    richTextBox1.Clear();
    for (int i = 0; i < text_4.Count; i++)
        richTextBox1.AppendText(text_4[i].ToString() + " ");
}

private void button5_Click(object sender, EventArgs e)
{
    number_vk1 = 5;
    richTextBox1.Clear();
    for (int i = 0; i < text_5.Count; i++)
        richTextBox1.AppendText(text_5[i].ToString() + " ");
}

private void button6_Click(object sender, EventArgs e)
{
    string[] mas_slova;
    richTextBox2.Clear();
    richTextBox3.Clear();
    if (number_vk1 == 1)
    {
        #region tab1
        k = 0;
        string text = richTextBox1.Text.ToLower();

        mas_text = t.txt1(text, mas, text_new);
        text_new = "";

        text1.Clear();
        for (int i = 0; i < mas_text.Length; i++)
        {
            text1.Add(mas_text[i]);
            mas_text[i] = "";
        }
        for (int i = 0; i < text1.Count; i++)
        {
            if (text1[i] == "\n" || text1[i] == " " || text1[i] == "" ||
string.IsNullOrEmpty(text1[i]))
            {
                text1.Remove(text1[i]);
            }
            else

```

```

        {
            richTextBox2.AppendText(text1[i] + "\n");
            richTextBox3.AppendText(text1[i].ToString() + " ");
            mas_text[k] = text1[i];
            k++;
        }
    }
    mas_kilk = new int[mas_text.Length];
    mas_slova = new string[mas_text.Length];

    mas_slova[0] = mas_text[0];
    zagal_kilk = 0;

    nowy_mas = t.mas(mas_text, mas_slova, mas_kilk);

    for (int i = 0; i < nowy_mas.Length; i++)
    {
        zagal_kilk += mas_kilk[i];
    }
    #endregion

    //Chastota
    chast = new double[nowy_mas.Length];
    Met.tf(nowy_mas, mas_kilk, zagal_kilk, chast);

    textBox3.Text = (zagal_kilk - mas_kilk[nowy_mas.Length - 1]).ToString();

    for (int i = 0; i < mas_text.Length; i++)
    {
        mas_text[i] = "";
    }
    i_len = 0;
    for (int i = 0; i < nowy_mas.Length; i++)
    {
        if (mas_kilk[i] > 1)
        { i_len++; }
    }

    mas1 = new string[i_len];
    chast_ = new double[i_len];
    mas_kilk_ = new int[i_len];
    grid2_0(nowy_mas, mas_kilk, chast, mas1, chast_, mas_kilk_);
    grid2_0_v(dataGridView1, mas1, mas_kilk_, chast_);
}

if (number_vkl == 2)
{
    #region tab2
    k = 0;
    string text = richTextBox1.Text.ToLower();

    mas_text = t.txt1(text, mas, text_new);
    text_new = "";

    text2.Clear();

    for (int i = 0; i < mas_text.Length; i++)
    {
        text2.Add(mas_text[i]);
        mas_text[i] = "";
    }

    for (int i = 0; i < text2.Count; i++)
    {
        if (text2[i] == "\n" || text2[i] == " " || text2[i] == "" ||
string.IsNullOrEmpty(text2[i]))
        {
            text2.Remove(text2[i]);
        }
        else
        {
            richTextBox2.AppendText(text2[i] + "\n");
            richTextBox3.AppendText(text2[i].ToString() + " ");
            mas_text[k] = text2[i];
            k++;
        }
    }
}

```

```

}
mas_kilk_2 = new int[mas_text.Length];
mas_slova = new string[mas_text.Length];

mas_slova[0] = mas_text[0];
zagal_kilk = 0;
nowy_mas_2 = t.mas(mas_text, mas_slova, mas_kilk_2);

for (int i = 0; i < nowy_mas_2.Length; i++)
{
    zagal_kilk += mas_kilk_2[i];
}
//Chastota
chast2 = new double[nowy_mas_2.Length];
Met.tf(nowy_mas_2, mas_kilk_2, zagal_kilk, chast2);

textBox3.Text = (zagal_kilk - mas_kilk_2[nowy_mas_2.Length - 1]).ToString();

i_len = 0;
for (int i = 0; i < nowy_mas_2.Length; i++)
{
    if (mas_kilk_2[i] > 1)
    { i_len++; }
}

mas2 = new string[i_len];
chast_2 = new double[i_len];
mas_kilk2 = new int[i_len];
grid2_0(nowy_mas_2, mas_kilk_2, chast2, mas2, chast_2, mas_kilk2);
grid2_0_v(dataGridView1, mas2, mas_kilk2, chast_2);
#endregion

#region tab1_3
k = 0;
text = richTextBox5.Text.ToLower();

mas_text = t.txt1(text, mas, text_new);
text_new = "";

text3.Clear();

for (int i = 0; i < mas_text.Length; i++)
{
    text3.Add(mas_text[i]);
    mas_text[i] = "";
}

for (int i = 0; i < text3.Count; i++)
{
    if (text3[i] == "\n" || text3[i] == " " || text3[i] == "" ||
string.IsNullOrEmpty(text3[i]))
    {
        text3.Remove(text3[i]);
    }
    else
    {
        mas_text[k] = text3[i];
        k++;
    }
}
mas_kilk_3 = new int[mas_text.Length];
mas_slova = new string[mas_text.Length];

mas_slova[0] = mas_text[0];
zagal_kilk = 0;
nowy_mas_3 = t.mas(mas_text, mas_slova, mas_kilk_3);

for (int i = 0; i < nowy_mas_3.Length; i++)
{
    zagal_kilk += mas_kilk_3[i];
}

```

```

//Chastota
chast3 = new double[nowy_mas_3.Length];
Met.tf(nowy_mas_3, mas_kilk_3, zagal_kilk, chast3);

i_len = 0;
for (int i = 0; i < nowy_mas_3.Length; i++)
{
    if (mas_kilk_3[i] > 1)
    { i_len++; }
}

mas3 = new string[i_len];
chast_3 = new double[i_len];
mas_kilk3 = new int[i_len];

#endregion
#region tab1_4
k = 0;
text = richTextBox6.Text.ToLower();

mas_text = t.txt1(text, mas, text_new);
text_new = "";

text4.Clear();

for (int i = 0; i < mas_text.Length; i++)
{
    text4.Add(mas_text[i]);
    mas_text[i] = "";
}

for (int i = 0; i < text4.Count; i++)
{
    if (text4[i] == "\n" || text4[i] == " " || text4[i] == "" ||
string.IsNullOrEmpty(text4[i]))
    {
        text4.Remove(text4[i]);
    }
    else
    {
        mas_text[k] = text4[i];
        k++;
    }
}
mas_kilk_4 = new int[mas_text.Length];
mas_slova = new string[mas_text.Length];

mas_slova[0] = mas_text[0];
zagal_kilk = 0;
nowy_mas_4 = t.mas(mas_text, mas_slova, mas_kilk_4);

for (int i = 0; i < nowy_mas_4.Length; i++)
{
    zagal_kilk += mas_kilk_4[i];
}

//Chastota
chast4 = new double[nowy_mas_4.Length];
Met.tf(nowy_mas_4, mas_kilk_4, zagal_kilk, chast4);

i_len = 0;
for (int i = 0; i < nowy_mas_4.Length; i++)
{
    if (mas_kilk_4[i] > 1)
    { i_len++; }
}

mas4 = new string[i_len];
chast_4 = new double[i_len];
mas_kilk4 = new int[i_len];

```

```

#endregion
#region tab1_5
k = 0;
text = richTextBox7.Text.ToLower();

mas_text = t.txt1(text, mas, text_new);
text_new = "";

text5.Clear();

for (int i = 0; i < mas_text.Length; i++)
{
    text5.Add(mas_text[i]);
    mas_text[i] = "";
}

for (int i = 0; i < text5.Count; i++)
{
    if (text5[i] == "\n" || text5[i] == " " || text5[i] == "" ||
string.IsNullOrEmpty(text5[i]))
    {
        text5.Remove(text5[i]);
    }
    else
    {
        mas_text[k] = text5[i];
        k++;
    }
}
mas_kilk_5 = new int[mas_text.Length];
mas_slova = new string[mas_text.Length];

mas_slova[0] = mas_text[0];
zagal_kilk = 0;
nowy_mas_5 = t.mas(mas_text, mas_slova, mas_kilk_5);

for (int i = 0; i < nowy_mas_5.Length; i++)
{
    zagal_kilk += mas_kilk_5[i];
}
//Chastota
chast5 = new double[nowy_mas_5.Length];
Met.tf(nowy_mas_5, mas_kilk_5, zagal_kilk, chast5);

i_len = 0;
for (int i = 0; i < nowy_mas_5.Length; i++)
{
    if (mas_kilk_5[i] > 1)
    { i_len++; }
}

mas5 = new string[i_len];
chast_5 = new double[i_len];
mas_kilk5 = new int[i_len];

#endregion

}

if (number_vk1 == 5)
{
    #region tab1_5
k = 0;
text = richTextBox7.Text.ToLower();

mas_text = t.txt1(text, mas, text_new);
text_new = "";

```

```

text5.Clear();

for (int i = 0; i < mas_text.Length; i++)
{
    text5.Add(mas_text[i]);
    mas_text[i] = "";
}

for (int i = 0; i < text5.Count; i++)
{
    if (text5[i] == "\n" || text5[i] == " " || text5[i] == "" ||
string.IsNullOrEmpty(text5[i]))
    {
        text5.Remove(text5[i]);
    }
    else
    {
        richTextBox2.AppendText(text5[i] + "\n");
        richTextBox3.AppendText(text5[i].ToString() + " ");
        mas_text[k] = text5[i];
        k++;
    }
}
mas_kilk_5 = new int[mas_text.Length];
mas_slova = new string[mas_text.Length];

mas_slova[0] = mas_text[0];
zagal_kilk = 0;
nowy_mas_5 = t.mas(mas_text, mas_slova, mas_kilk_5);

for (int i = 0; i < nowy_mas_5.Length; i++)
{
    zagal_kilk += mas_kilk_5[i];
}
//Chastota
chast5 = new double[nowy_mas_5.Length];
Met.tf(nowy_mas_5, mas_kilk_5, zagal_kilk, chast5);

textBox3.Text = (zagal_kilk - mas_kilk_5[nowy_mas_5.Length - 1]).ToString();

i_len = 0;
for (int i = 0; i < nowy_mas_5.Length; i++)
{
    if (mas_kilk_5[i] > 1)
    { i_len++; }
}

mas5 = new string[i_len];
chast_5 = new double[i_len];
mas_kilk5 = new int[i_len];
grid2_0(nowy_mas_5, mas_kilk_5, chast5, mas5, chast_5, mas_kilk5);
grid2_0_v(dataGridView1, mas5, mas_kilk5, chast_5);

#endregion
}
}

public void grid2_0(string[] nowy_mas, int[] mas_kilk, double[] chast, string[] mas1, double[] chast_,
int[] mas_kilk_)
{
    for (int i = 0; i < nowy_mas.Length; i++)
    {
        if (mas_kilk[i] > 1)
        {
            mas1[i] = nowy_mas[i];
            chast_[i] = chast[i];
            mas_kilk_[i] = mas_kilk[i];
            len++;
        }
    }
    len = 0;
}

public void grid2_0_v(DataGridView dataGridView1, string[] mas1, int[] mas_kilk_, double[] chast_)

```

```

    {
        dataGridView1.RowCount = mas1.Length;

        for (int i = 0; i < mas1.Length; i++)
        {
            dataGridView1.Rows[i].Cells[0].Value = mas1[i];
            dataGridView1.Rows[i].Cells[1].Value = mas_kilk_[i];
            dataGridView1.Rows[i].Cells[2].Value = chast_[i];
        }
    }
}
}
Лістинг Text.cs:
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace klasyfikator
{
    class Text
    {
        public string[] txt1(string text, string[] mas, string text_new)
        {
            string[] mas_chys = { "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "\n", "\r", "+", "°" };
            int i, j, k = 0;

            for (i = 0; i < text.Length; i++)
            {
                if (Char.IsPunctuation(text[i]))
                {
                    k = 1;
                }
                else
                {
                    for (j = 0; j < mas_chys.Length; j++)
                    {
                        if (Convert.ToString(text[i]) == mas_chys[j])
                        {
                            k = 1;
                            break;
                        }
                    }

                    for (j = 0; j < mas.Length; j++)
                    {
                        if (text[i] == Convert.ToChar(mas[j])) { k = 0; }
                    }

                    if (k == 0)
                    {
                        text_new += Convert.ToString(text[i]);
                    }
                    k = 0;
                }
            }

            return text_new.Split(' ');
        }
        public string[] mas(string[] mas_text, string[] mas_slova, int[] mas_kilk)
        {
            int i, j;
            int kilk, zmin = 1, zagal_kilk = 0;
            for (i = 1; i < mas_text.Length; i++)
            {
                kilk = 0;

                for (j = 0; j < mas_text.Length; j++)
                {
                    if (mas_text[i] == mas_slova[j])
                    {
                        kilk++;
                    }
                }
            }
        }
    }
}

```



```

class Classification
{
    //запис
    public void klas_statti(DataGridView dataGridView8, int len_for_kat, string[,] mas_ob, int[]
mas_kilk_test, int i, double[] tf_idf1, int z, string[] nowy_mas_test)
    {
        for (int j = 0; j < len_for_kat; j++)
        {
            if (mas_ob[j, 0] == nowy_mas_test[i])
            {
                dataGridView8.Rows[i].Cells[z].Value = mas_kilk_test[i] * Convert.ToDouble(mas_ob[j, 3]);
                break;
            }
            else dataGridView8.Rows[i].Cells[z].Value = 0;
        }
    }
}
}
}

```

Лістинг Methods.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace klasyfikator
{
    class Methods
    {
        //TF
        public double[] tf(string[] nowy_mas, int[] mas_kilk, int zagal_kilk, double[] chast)
        {
            for (int i = 0; i < nowy_mas.Length; i++)
            {
                chast[i] = Convert.ToDouble(mas_kilk[i]) / zagal_kilk;
            }
            return chast;
        }
        //TF-IDF
        public double[] tf_idf(string[] nowy_mas, int[] kilk_text, double[] idf)
        {
            for (int i = 0; i < nowy_mas.Length; i++)
            {
                idf[i] = Math.Log((7 / (kilk_text[i] + 1)));
            }
            return idf;
        }
    }
}

```

Лістинг Sum.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;

namespace klasyfikator
{
    class Sum
    {
        public void Summ(DataGridView dataGridView8, double s, string[] nowy_mas_test, int k)
        {
            for (int i = 0; i < nowy_mas_test.Length; i++)
            {
                s += Convert.ToDouble(dataGridView8.Rows[i].Cells[k].Value);
            }
            dataGridView8.Rows[nowy_mas_test.Length].Cells[k].Value = s;
        }
    }
}
}

```

Додаток Г

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Автоматизована інформаційна система тематичної класифікації наукових статей

Виконала: студентка 4 курсу групи КН-17-2 Шпичко А.В.

Керівник: к.т.н., доцент кафедри КНІТ Мазурець О.В.

Актуальність

Кожного року наукові статті збільшуються в кількості, і кожна з них має відноситись до певної науки, тобто повинна мати свою класифікацію. Для більш зручної та чіткої класифікації використовуються ключові слова кожної статті. За допомогою ІТ, пошук цих слів відбувається автоматично, що значно економить час та відносить статтю до правильної класифікації.

Таким чином, у сфері класифікації, використовуючи ІТ можна швидко віднести будь-яку статтю до відповідного їй класу класифікації.



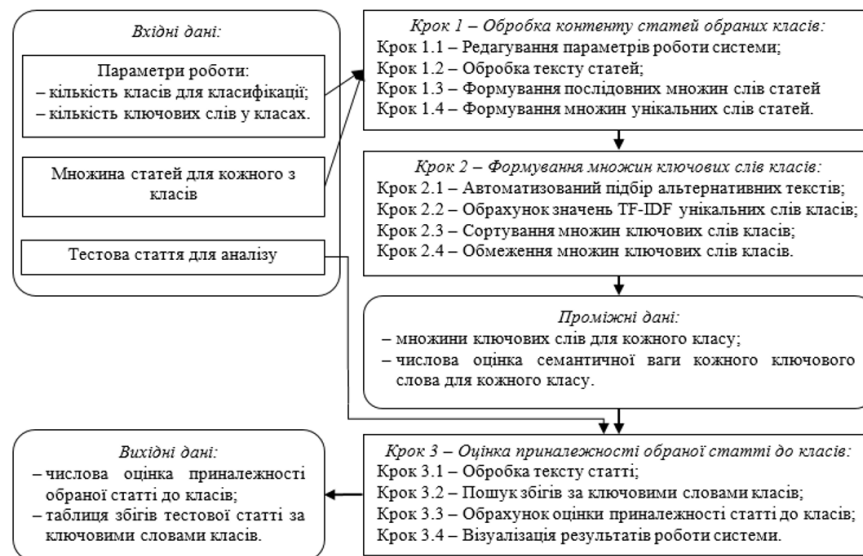
Завдання

Мета кваліфікаційної роботи бакалавра – створення автоматизованої інформаційної системи тематичної класифікації наукових статей.

Автоматизувати наступні функції:

- вибір кількості класів для класифікації
- вибір кількості ключових слів у класах
- обробка тексту статей
- формування послідовних множин та унікальних множин слів статей
- автоматизований підбір альтернативних текстів
- обрахунок значень TF-IDF унікальних слів класів
- сортування множин ключових слів класів
- обмеження множин ключових слів класів
- перехід між тематичними класифікаціями
- пошук збігів тестової статті за ключовими словами класів
- обрахунок оцінки приналежності тестової статті до класів
- візуалізація результатів роботи системи

Схема інформаційної технології тематичної класифікації наукових статей



Структура компонентів інформаційної системи

Підсистема роботи з базою даних та параметрами:

- редагування параметрів роботи системи;
- робота (додавання, редагування, видалення) із базою даних (вхідні, проміжні та вихідні дані).

Підсистема формування множин ключових слів класів:

- редагування параметрів роботи системи;
- обробка тексту статей;
- формування послідовних множин слів статей;
- формування множин унікальних слів статей;
- автоматизований підбір альтернативних текстів;
- обрахунок значень TF-IDF унікальних слів класів;
- сортування множин ключових слів класів;
- обмеження множин ключових слів класів.

БД автоматизованої інформаційної системи тематичної класифікації наукових статей (дозволяє зберігати дані про: класи, альтернативні множини статей, статті класів, ключові та унікальні слова статей).

Підсистема оцінювання приналежності обраної статті до класів:

- обробка тексту статті;
- формування послідовних множин слів статті;
- формування множин унікальних слів статті;
- пошук збігів за ключовими словами класів;
- обрахунок оцінки приналежності статті до класів;
- візуалізація результатів роботи системи.

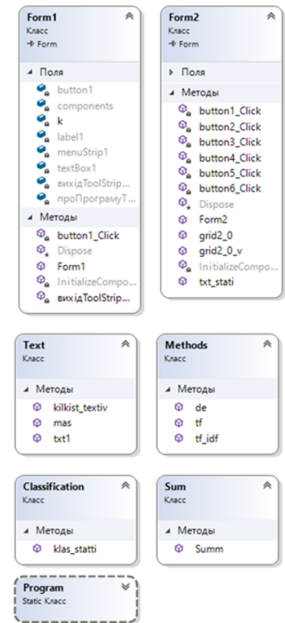
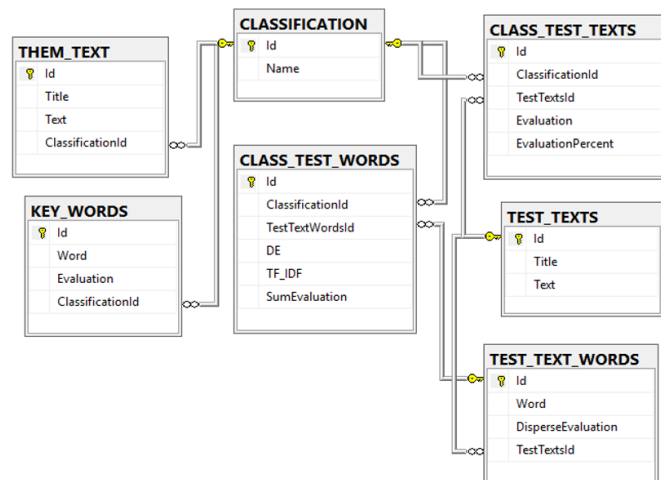


Схема бази даних

Для забезпечення збереження даних було розроблено базу даних автоматизованої інформаційної системи тематичної класифікації наукових статей



Початковий етап роботи АІС

The image shows the initial setup phase of the AI classifier. On the left is the 'Налаштування' (Settings) dialog box with the following options:

- Оберть кількість класів класифікації (від 2 до 10):
- Оберть кількість ключових слів для класів (від 10 до 500):
- Редактор бази статей
- Почати роботу

An arrow points to the main application window 'Класифікатор наукових статей Шпичко'. The 'Класифікація статей' (Article Classification) tab is active, showing the 'Обробка статей обраного класу' (Processing articles of the selected class) section. It includes a preview of the selected article text and a table for the 'Множина унікальних слів класу' (Unique word set of the class).

Визначення множини ключових слів класифікації

The image shows the 'Пошук множин ключових слів для обраного класу статей' (Search for sets of key words for the selected class of articles) section. It displays 'Альтернативні тексти' (Alternative texts) and a table for the 'Множина ключових слів класу' (Key word set of the class).

The 'Альтернативні тексти' section shows three alternative texts (N1, N2, N3) and their corresponding TF-IDF values.

The 'Множина ключових слів класу' table is as follows:

Слово	TF	IDF	TF-IDF
наукового	0.011627906976...	1.945910149055...	0.022626862198...
товариства	0.008305647840...	1.945910149055...	0.016162044427...
журналу	0.006644518272...	1.945910149055...	0.012929639541...
природознавства	0.006644518272...	1.945910149055...	0.012929639541...
україни	0.006644518272...	1.945910149055...	0.012929639541...
вісник	0.004983388704...	1.945910149055...	0.009697226656...
статей	0.004983388704...	1.945910149055...	0.009697226656...
журнал	0.004983388704...	1.945910149055...	0.009697226656...
житті	0.004983388704...	1.945910149055...	0.009697226656...
харківського	0.004983388704...	1.945910149055...	0.009697226656...
вивчення	0.004983388704...	1.945910149055...	0.009697226656...
видання	0.004983388704...	1.945910149055...	0.009697226656...
без	0.004983388704...	1.945910149055...	0.009697226656...
професора	0.004983388704...	1.945910149055...	0.009697226656...
або	0.006644518272...	1.098612289668...	0.007299749426...

Buttons at the bottom include: 'Автоматизований підбір альтернативних текстів', 'Обрахунок значень TF-IDF для унікальних слів обраного класу', and 'Сортування та обмеження множини ключових слів класу'. A note at the bottom states: 'Виконати всі кроки формування множин ключових слів класу автоматично'.

Визначення класифікації тестової статті

Класифікатор наукових статей Шпичко

Класи статей: Природні науки, Технічні науки, Медичні науки, Суспільні науки, Гумантарні науки

Визначення приналежності обраної статті до класів

Тестова стаття для класифікації:

Існуючі методи керування та вищур високочастотних ліній зв'язку мають можливість дозволити забезпечити досить хороші характеристики для контролю вимірюваних значень. Низькочастотна лінія в основному представлена коаксіальною лінією, телефонної провідної лінії. Як і високочастотні лінії, ці провідні лінії також повинні контролюватися. Для вирішення проблеми вищур висотні пошкодження існують велика кількість приладів, заснованих на використанні одного з вищезазначених методів (як правило, імпульсного). Далекомири, засновані на вимірюванні фазового зсуву, ще не знайшли широкого практичного застосування. Пристрої, які засновані на виявленні імпульсних ліній.

Оцінка приналежності обраної статті до класів:

Класифікація	Оцінка	Відсоток
Природні науки	0	0%
Технічні науки	0,0689659...	76,31%
Медичні науки	0,0084530...	9,35%
Суспільні науки	0	0%
Гумантарні науки	0,0129629...	14,34%

Показники збігу тестової статті за ключовими словами класів:

	Природні науки	Технічні науки	Медичні науки	Суспільні науки	Гумантарні науки
існуючі	0	0	0	0	0
методи	0	0	0	0	0
керування	0	0	0	0	0
та	0	0	0	0	0
вищур	0	0	0	0	0
високочастотних	0	0	0	0	0
ліній	0	0	0	0	0
зв'язку	0	0	0	0	0
мають	0	0	0	0	0
можливість	0	0	0	0	0
дозволити	0	0	0	0	0
забезпечити	0	0	0	0	0
досить	0	0	0	0	0
хороші	0	0	0	0	0
характеристики	0	0,0163362...	0	0	0
для	0	0,0180372...	0	0	0

Обробка тексту статті | Пошук збігу тестової статті за ключовими словами класів | Обрахунок оцінки приналежності обраної статті до класів

Виконати всі кроки визначення приналежності обраної статті до класів автоматично

Висновки

В результаті виконання кваліфікаційної роботи бакалавра було розроблено автоматизовану інформаційну систему тематичної класифікації наукових статей. Розроблена система виконує такі основні функції:

- вибір кількості класів для класифікації
- обробка тексту статей
- формування послідовних множин та унікальних множин слів статей
- обрахунок значень TF-IDF унікальних слів класів
- перехід між тематичними класифікаціями
- обрахунок оцінки приналежності тестової статті до класів
- візуалізація результатів роботи системи

ІС може бути впроваджена в бібліотеки, навчальні заклади та інші електронні системи, що допоможе полегшити роботу з класифікацією наукових статей.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ

Направляється студент Шпичко А. В. на захист дипломного проекту (роботи)
(прізвище, ініціали)

за спеціальністю 122 - Комп'ютерні науки

На тему: Автоматизована інформаційна система тематичної класифікації наукових статей

Дипломний проект (робота), рецензія і довідка про перевірку на плагіат додаються.

Декан факультету



САВЕНКО О.С.

(прізвище та ініціали)

ДОВІДКА УСПІШНОСТІ

Шпичко А. В. за період навчання на факультеті програмування та комп'ютерних і телекомунікаційних систем з 2017 по 2021 роки повністю виконав навчальний план спеціальності з такими розподілом оцінок за:

національною шкалою: відмінно 93,75 %, добре 6,25 %, задовільно 0,00 %.
шкалою ЄКТС: А 96,36 %, В 3,64 %, С 0,00 %, D 0,00%, Е 0,00 %.

Методист факультету

(підпис)

(прізвище та ініціали)

ВИСНОВОК КЕРІВНИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ) ТА ОБГРУНТУВАННЯ ОЦІНКИ

Студент Шпичко А. В. при виконанні класифікаційної роботи проявив себе кваліфікованим фахівцем, дисципліновано і наполегливо виконав поставлені задачі. Робота виконана на високому рівні, на актуальну тематику, є унікальними в комунікаційних та науковій публікації. Використано сучасні технології розробки програмного забезпечення.

Оцінка дипломного проекту (роботи) відмінно

Керівник дипломного проекту (роботи)

(підпис)

(прізвище та ініціали)

" 08 " 06 2021 р.

ВИСНОВОК КАФЕДРИ ПРО ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Дипломний проект (роботу) розглянуто. Студент Шпичко А. В. допускається до захисту цього

Завідувач кафедри

КМІТ

(назва)

" 08 " 06 2021 р.

(підпис, прізвище, ініціали)

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 8.0%

Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 11%

ID: 92922 Название: Автоматизована інформаційна система тематичної класифікації наукових статей Добавлено в БД: 2021-06-09 Авторы: А.В. Шпичко Руководители: О.В. Мазурець Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	50246	473	6218 (12%)	90 (19%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы
90098	Название: ЗВІТ з професійної практики Добавлено в БД: 2021-05-10 Авторы: Шпичко А.В. Руководители: Скрипник Т.К. Консультанты: Оponentы:	3814 (8.0%)	55 (12.0%)

Ім'я користувача:
Кафедра КН

Дата перевірки:
09.06.2021 16:41:06 EEST

Дата звіту:
09.06.2021 16:42:50 EEST

ID перевірки:
1008247361

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005671

Назва документа: КРБ Шпичко 20210607 4 фінал повний Lite

Кількість сторінок: 53 Кількість слів: 8029 Кількість символів: 60064 Розмір файлу: 2.62 MB ID файлу: 1008319516

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

10.3% Схожість

Найбільша схожість: 3.08% з джерелом з Бібліотеки (ID файлу: 1008318499)

6.81% Джерела з Інтернету

324

Сторінка 55

5.06% Джерела з Бібліотеки

65

Сторінка 56

0% Цитат

Вилучення цитат виключено

Вилучення списку бібліографічних посилань виключено

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

6

Підозріле форматування

18
сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Автоматизована інформаційна система тематичної класифікації наукових статей

Автор: студентка групи КН-17-2 Шпичко Анастасія Валентинівна

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: доцент кафедри КНІТ Мазурець О.В.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<i>відповідає</i>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні;
- 3) до запозичень входять фрагменти програмного коду, що на мають авторства і містять поширені конструкції;
- 4) серед запозичень знаходяться загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 10.3 % і адресується до першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

О. В. Мазурець

Гарант ОП

О. В. Мазурець

Завідувач кафедри КНІТ

О. В. Бармак

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-17-2 Шпичко Анастасії Валентинівни

за темою Автоматизована інформаційна система тематичної класифікації наукових статей

1. Актуальність і значення теми Наукові дослідження у вигляді статей відіграють важливу роль у формуванні майбутніх спеціалістів. Кожна наукова стаття має відноситись до певної науки, тобто повинна мати свою класифікацію. Для більш зручної та чіткої класифікації використовуються ключові слова кожної статті. Використання інформаційних технологій дозволяє швидко віднести наукову статтю до відповідного їй класу класифікації.

2. Оцінка запропонованих моделей, підходів, алгоритмів, інформаційної складової та засобів розробки Розроблені автором схема і компоненти інформаційної технології тематичної класифікації наукових статей дозволяють ефективно формувати семантичні вектори вказаних користувачем класів тематичної класифікації, що дозволяє в подальшому визначати числові оцінки приналежності обраної статті до класів. Розроблена інформаційна система підтвердила високу ефективність запропонованих рішень з тематичної класифікації наукових статей.

3. Оцінка розробленої інформаційної системи, її практична цінність та економічна доцільність Розроблена інформаційна система дозволяє за вхідними даними (параметри роботи, множина статей для кожного з класів та тестова стаття для аналізу) одержувати вихідні дані (оцінка приналежності обраної статті до класів та таблиця збігів тестової статті за ключовими словами класів), дозволяє ефективно вирішувати актуальну задачу інформаційних технологій, автоматизовану тематичну класифікацію наукових статей.

4. Загальний висновок та оцінка Всі поставлені завдання виконані повністю і на високому професійному рівні. За своєю структурою, практичними цінностями, поставленій меті та вирішеними завданнями робота відповідає вимогам вищої школи і вимогам, що пред'являються до освітньо-кваліфікаційного рівня «бакалавр», а її автор Шпичко А.В. заслуговує присвоєння кваліфікації бакалавра з комп'ютерних наук.

Робота заслуговує на оцінку « Відмінно ».

Рецензент А.Г.Н., доц. каф. ТМІТ ФІІТ Олександр О.Ю