Конкурсна наукова робота

на тему «Дослідження рівня унікальності текстового контенту та розробка програмного застосування для перевірки рівня унікальності текстового контенту»

Галузь 30 Інформаційні технології

Шифр – Text Uniqueness

ЗМІСТ

ВСТУП 3
Розділ 1. Дослідження та аналіз специфіки оцінки унікальності текстового контенту 4
Розділ 2. Розробка програмного застосування перевірки рівня унікальності текстового контенту 13
2.1 Проектування програмного застосування 13
2.2 Розробка схеми складу класів та алгоритму оцінки рівня унікальності текстового контенту 15
2.3 Реалізація функціональних можливостей програмного продукту 17
2.3.1 Розробка інтерфейсу користувача 18
ВИСНОВКИ 27
ПЕРЕЛІК ПОСИЛАНЬ 28

ВСТУП

Об’єкт дослідження наданої роботи – унікальність контенту, предмет дослідження – дослідження специфіки перевірки унікальності текстового контенту. Головна проблема при створенні сайту або статті - унікальність контенту [1]. Не має значення, чи пишете ви власноруч статті, чи замовляєте наповнення сайту у копірайтера (від анг. copywriter – автор і спеціаліст по створенню рекламних та презентаційних текстів), все одно необхідно перевірити тексти цих робіт на унікальність. Унікальність складу статті дуже важлива. На неунікальні сайти та статті накладаються санкції пошукових систем, що зменшує кількість відвідувачів сайту. Розміщуючи чужий текст (без вказівки авторства) людина ризикує отримати скаргу на порушення авторських прав. Актуальність моєї дослідницької роботи полягає в тому, що в роботі проведений аналіз методів перевірки рівня унікальності текстового контенту та розроблено програмне застосування перевірки рівня унікальності текстового контенту.

Мета роботи полягає у розробці програмного забезпечення оцінки рівня унікальності текстового контенту для забезпечення функцій швидкої та якісної перевірки унікальності заданого тексту статті чи тексту файлу.

Завданнями роботи є:

1. Проведення аналізу особливостей та призначення перевірки рівня унікальності текстового контенту.

2. Аналіз існуючих методів перевірки рівня унікальності тексту та існуючих програмних продуктів з перевірки контенту на плагіат.

3. Обґрунтування використаних програмних засобів розробки.

4. Розробка UML діаграм проекту програмного застосування.

5. Розробка алгоритму роботи програми.

6. Програмна реалізація програмного забезпечення перевірки рівня унікальності текстового контенту.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ СПЕЦИФІКИ ОЦІНКИ УНІКАЛЬНОСТІ ТЕКСТОВОГО КОНТЕНТУ

Унікальність тексту - показник відсутності дублів тексту в Інтернеті [2]. Унікальність є одним з базових критеріїв, за якими пошукові системи оцінюють якість текстового контенту. За публікацію неунікального контенту, на сайт, скоріше за все, будуть накладені санкції пошукових систем. До того ж, неунікальна інформація навряд чи представляє цінність і користь для відвідувачів сайту.

Плагіат — привласнення авторства на чужий твір або на чуже відкриття, винахід чи раціоналізаторську пропозицію, а також використання у своїх працях чужого твору без посилання на автора [3]. Плагіат з появою Інтернету перетворився в серйозну проблему.

Потрапивши в Інтернет, знання стає надбанням всіх, дотримуватися авторського права стає все важче і навіть неможливо. Поступово стає складніше визначити первісного автора.

Стрімкий розвиток мережі Інтернет поряд зі зростаючою комп'ютерною грамотністю сприяє проникненню плагіату в різні сфери людської діяльності: плагіат є гострою проблемою в освіті, промисловості та науковому співтоваристві.

Плагіат є злочином. Це вводить в оману читачів, приносить шкоду автору, і надає незаслужені блага плагіатору.

Широкий доступ до вітчизняної та зарубіжної літератури, багаторазове збільшення числа професійних видань, публікацій в Інтернеті - все це практично зводить нанівець будь-які редакторські прагнення «перевірити» або «встановити» справжність і оригінальність аргументів і фактів, які використовуються в рукописах, пропонованих до публікації [4].

На даний момент існує ряд комп’ютерних методів виявлення плагіату в залежності від його виду та заплутаності. Основні методи [5] по рівню їх ефективності наведені на рис. 1.1.

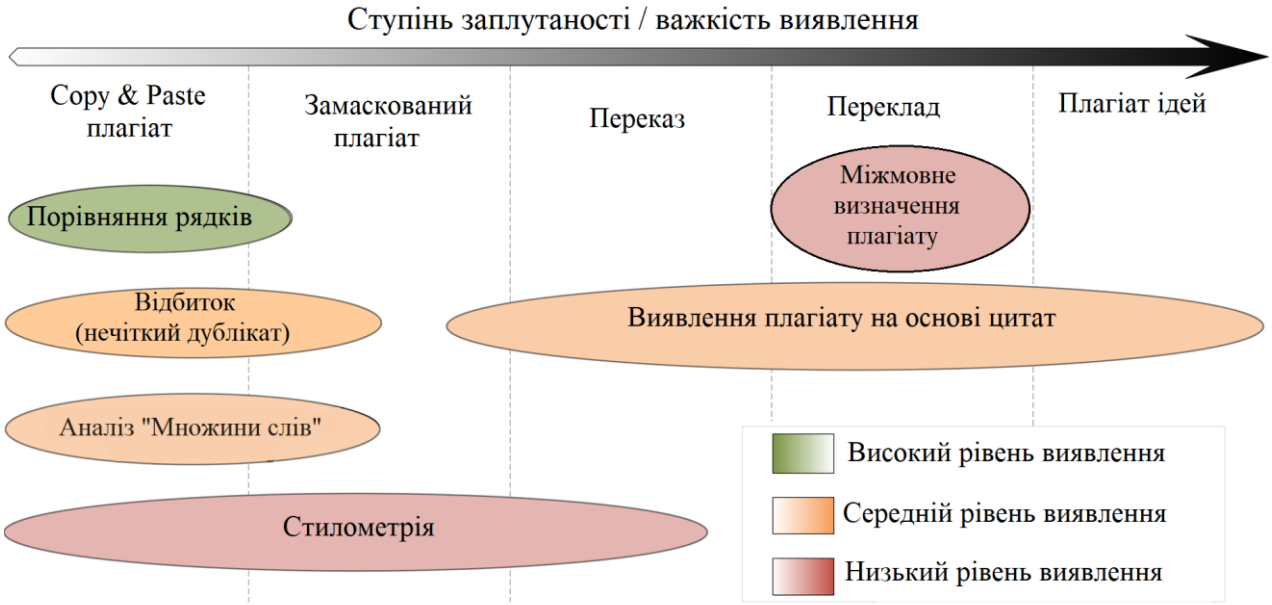

Рисунок 1.1 – Основні комп’ютерні методи виявлення плагіату в залежності від типу заплутаності

Типи унікальності текстового контенту:

1. Формальна. Оцінка проводиться за технічними показниками (послідовність слів, структура тексту і т.д.). Формальна унікальність показує, що стаття ніде більше не опублікована в тому вигляді, в якому вона є на аналізованому сайті. При цьому зміст може і не бути унікальним (тобто може дублюватися на інших сайтах, тільки в іншому викладі).

2. Смислова. Критерієм оцінки виступає зміст. Смислова унікальність тексту означає, що в статті викладена інформація, яка не освітлювалася на інших джерелах. Зазвичай такі тексти пишуться експертами в певному питанні і містять корисні для читачів тематичні відомості, поради, рекомендації і т.д.. Нерідко подібні статті супроводжуються фотографіями, рисунками, схемами, таблицями та ін.

Для відвідувачів сайту найбільше важить саме смислова оригінальність. Для успішного просування важливо забезпечувати, щоб контент був оригінальним і з формальної, і зі смислової точки зору [6].

Можна також класифікувати унікальність за джерелом:

– Всередині сайту. Оригінальність тексту визначається по відношенню до інших сторінок аналізованого ресурсу. Цей критерій особливо важливий для інтернет-магазинів, адже описи багатьох товарів є ідентичними або дуже схожими [7];

– В інтернеті. Перевірку проходить контент, опублікований на всіх сайтах в мережі.

Методи виявлення плагіату. Методи характеризуються по типу оцінки подібності. На рис.1.2 наведена класифікація методів комп'ютерного виявлення плагіату з технічної точки зору.

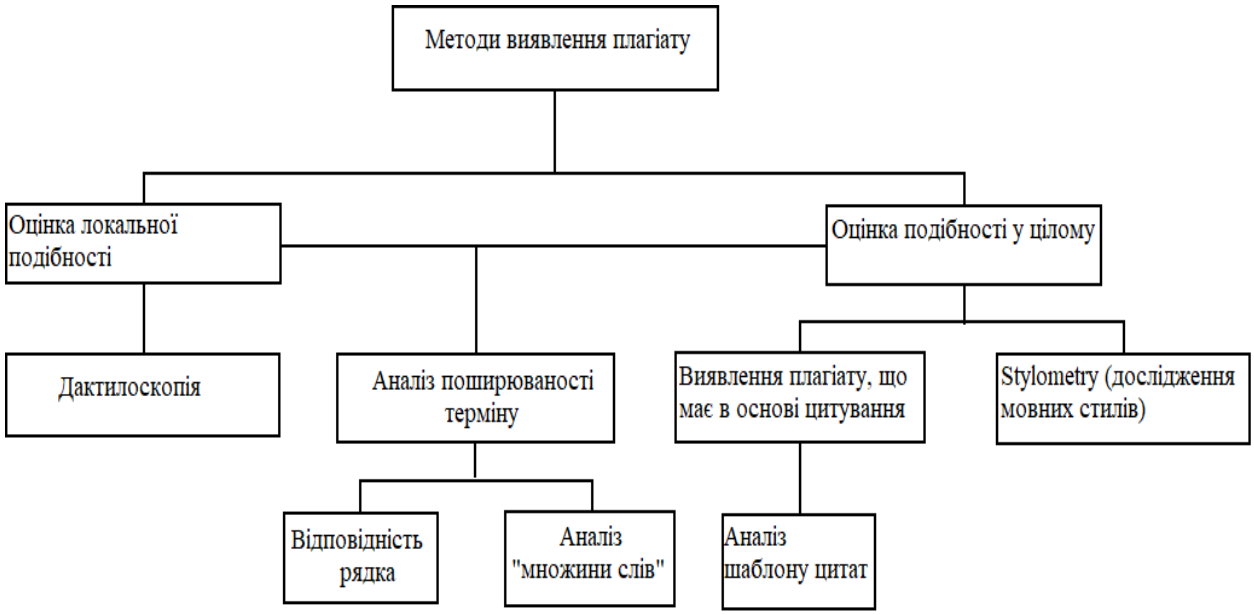

Рисунок 1.2 – Схема класифікації методів комп’ютерного виявлення плагіату

Глобальна оцінка використовує великі частини тексту або документа для знаходження подібності в цілому, а локальні методи на вході перевіряють обмежений сегмент тексту.

У даний час найбільш поширеним підходом є дактилоскопія. З ряду документів вибирається набір з декількох рядків, які і є «відбитками». Розглянутий документ буде порівнюватися з «відбитками» для всіх документів колекції. Знайдені відповідності з іншими документами вказують на загальні сегменти тексту [8].

Перевірка документа дослівним перебором тексту представляє собою класичне порівняння рядків. Як правило, у цьому методі використовують такі моделі, як суфіксне дерево або суфіксний масив, які були адаптовані для виконання цього завдання в контексті комп'ютерного виявлення плагіату. Однак зіставлення рядків є нежиттєздатним рішенням для перевірки великих колекцій документів (алгоритм відпрацьовує в середньому 2h порівнянь, де h - довжина рядка, в якій ведеться пошук).

Аналіз "множини слів" є спрощенням уявлення, що використовується в обробці природної мови і пошуку інформації. У цій моделі текст представлений як невпорядкований набір слів. Документи представлені у вигляді одного або декількох векторів, які використовуються для попарного виявлення подібності [9].

Цитування - комп'ютерний метод виявлення плагіату, призначений для використання у наукових документах, що дозволяє використовувати цитати і довідковий матеріал. Визначає спільні цитати двох наукових робіт. Шаблон цитат являє собою підпослідовності, що містять не тільки спільні цитати для двох документів, а й подібний порядок і близькість цитат в тексті, що є основними критеріями для визначення шаблону цитат [10].

Стильометрія або вивчення мовних стилів - це статистичний метод для виявлення авторства анонімних документів і для їх комп'ютерної перевірки на плагіат [11]. Будуються стильометричні моделі для різних фрагментів тексту та уривків, які стилістично відрізняються від інших. Шляхом порівняння моделей можна виявити плагіат. Наприклад, аналіз на основі послідовностей частин мови. Розглядається спосіб розбиття тексту на фрагменти однорідності. Як параметри розбиття беруться різні послідовності частин мови. Далі проводиться аналіз фрагментів. І в результаті для тексту знаходяться послідовності, які виділяли з текстів фрагменти, тобто алгоритм виділяє з тексту фрагменти неоднорідності, що мають різні частоти знаходження обраної послідовності частин мови, що показує на можливий плагіат в даному місці.

Існує два способи перевірки тексту на унікальність:

1. Онлайн сервіси. Більшість з них працює безкоштовно або умовно безкоштовно: без оплати є можливість перевірити на унікальність обмежене число текстів на добу та/або статті обмеженого об’єму, наприклад від п’яти до десяти тисяч символів.

2. Програми-антіплагіатори. Такі програми встановлюються на персональний комп’ютер та працюють як і інше програмне забезпечення. Кожна програма використовує свій алгоритм перевірки тексту. Для перевірки обов’язково потребується доступ до Інтернету.

Розглянемо декілька комп'ютерних програмних продуктів для статистичної обробки даних: Antiplagiarism.NET, Advego Plagiatus, Антиплагиат.ру.

AntiPlagiarism.NET (друга назва ETXT антіплагіат) – один з найпоширеніших програмних продуктів для перевірки тексту та рисунків на наявність запозичень з різних джерел інтернету. Програма може визначати унікальність тексту по збереженим копіям шести пошукових систем (Google, Bing, Rambler, QIP, Nigma, Yahoo), виділяти неунікальні фрагменти кольором і підраховувати відсоток збігів, виконувати перевірку тексту на поверхневий рерайт (від англ. rewriting – переписати, переробити), вести пакетну перевірку текстів, збережених в одній папці, порівнювати два завантажених тексти, перевіряти на унікальність зображень та видавати основні параметри при SEO-перевірці (для оптимізації тексту сайту для пошукових систем [12]). До переваг можна віднести автоматичну систему виділення окремих фраз і слів, які тлумачяться, як неунікальний текст. Але є і ряд недоліків – повільна швидкість роботи, необхідність вводити капчу (від англ. CAPTCHA - Completely Automated Public Turing test to tell Computers and Humans Apart – повністю автоматизований публічний тест Тьюринга для розрізнення комп’ютерів і людей) та можливі помилки під час аналізу введеного тексту, перевантажений інтерфейс [13].

Мета сервісу antiplagiat.ru - перевірка унікальності тексту з метою недопущення незаконного використання авторських матеріалів. Проект має

окрему, спеціальну версію для вищих учбових закладів. Послуги надаються усім зареєстрованим користувачам сайту з обмеженою функціональністю або на платній основі з розширеною функціональністю (по принципу freemium). У безкоштовній версії системи доступна лише коротка форма звіту. Аналіз робіт у цій системі виконується на основі спеціалізованої системи пошуку та обробки інформації, що була розроблена при участі російських вчених-математиків. Використання antiplagiat.ru дозволяє встановити факт не тільки прямого запозичення тексту цілком або його частини, але навіть якщо плагіатор в документі замінив окремі слова на синоніми, замість букв українського чи російського алфавіту використовував латинські (і навпаки), якщо проводилася перестановка слів у реченні, речень в абзаці, перегрупування абзаців, поділ або з'єднання речень [14]. При необхідності система має можливість відстежити неминуче цитування нормативно-правових документів в текстах юридичної тематики. Цитування з «білих» джерел буде відображено в звіті, але не знизить унікальність авторського тексту. Проте сервіс має ряд недоліків, таких як неточність результатів, нестабільність роботи, дуже обмежені можливості безкоштовної версії сервісу, високі ціни на послуги.

Advego Plagiatus – безкоштовна програма для перевірки унікальності текстів. Програма поширюється біржою статей та копірайтингу “Advego”. Програма видає результат у вигляді двох чисел. Перше число показує кількість унікального тексту у загальному об’ємі по знайденим збігам входжень шинглів (від англ. Shingles – лусочки). Друге враховує також непрямі входження шинглів, лексичні збіги за словами.

Завдяки інтелектуальним алгоритмам програма виявляє всі недобросовісні методи обходу перевірки на плагіат, такі як:

1. Заміна букв в словах на іншу розкладку: програма виділяє такі заміни кольором і автоматично підставляє правильні символи при перевірці.

2. Перестановка слів місцями, синонімайзінг, заміна слів на застарілі: джерела будуть знайдені завдяки порівнянню лексичного складу тексту, другий показник унікальності за словами буде низьким [15].

3. Додавання вставних слів, зміна відмінків і часів: алгоритм перевірить текст без «води» і джерела зі схожими формами слів, щоб знайти оригінальний текст.

4. Обробка тексту сервісами «обходу антиплагіату», вставка спец. символів і невидимих знаків: текст автоматично очищається від сторонніх символів, що гарантує якісну перевірку.

Програма має не досить зручний інтерфейс, потребує досить частого вводу капчі та видає неоднозначні результати при повторній перевірці текстового контенту [16].

Проведемо аналіз програм для статистичної обробки даних:

Критерій	Antiplagiarism.NET	Advego Plagiatus	Антиплагиат.ру
Тип поширення	Freemium	Shareware	Freemium
Тип програми	Програмне застосування	Веб сервіс та програмне застосування	Веб сервіс
Необхідність вводити капчу	+	При використанні програмного застосування	-
Додаткові можливості	Перевірка на рерайт, перевірка унікальності малюнків, порівняння текстів, SEO-перевірка тексту	Перевірка на рерайт, SEO-перевірка тексту	Глибока перевірка на рерайт
Доступні платформи	Windows	Windows, Linux, MacOS та Веб версія	Веб версія
Ціна	20$	Безкоштовна	Від 2$ в день до 2740$ в місяць

Швидкість перевірки	Повільна	Середня	Середня
Точність результатів	Середня	Середня	Середня

Побудуємо графік якості програм у залежності від їх характеристик.

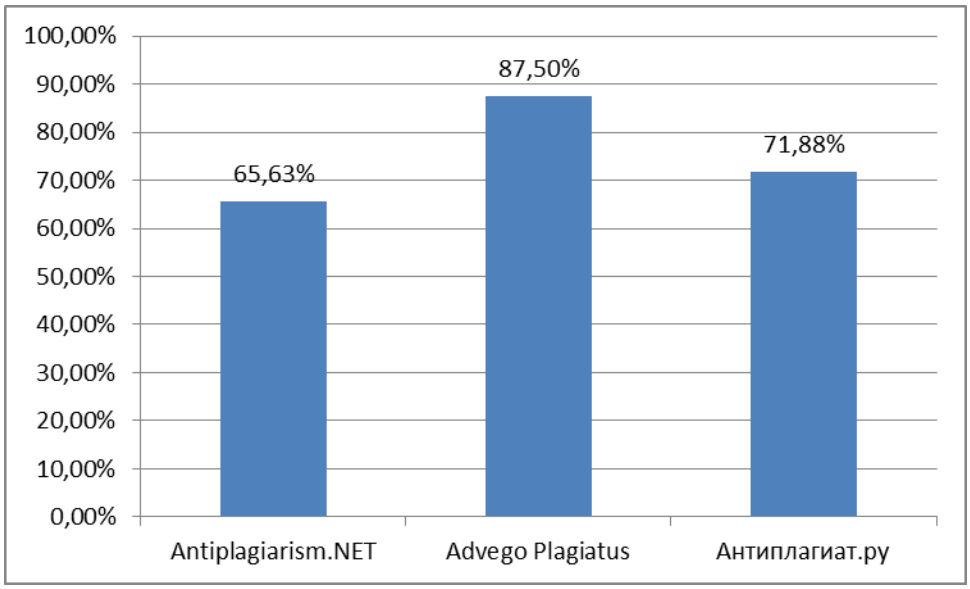

Рисунок 1.3 – Аналіз якості програм перевірки рівня унікальності тексту

За результатами проведеного аналізу якості програм перевірки рівня унікальності тексту, найбільш якісною є програма Advego Plagiatus за показниками тип поширення, тип програми, доступні платформи. Проте і це програмне застосування має ряд недоліків, таких як середня швидкість перевірки тексту та недостатня точність результатів.

Для створення програмного додатку були використані мова програмування Java та середовище програмування NetBeans.

Java — об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код,

який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

NetBeans IDE — вільне інтегроване середовище розробки (IDE) для мов програмування Java, JavaFX, C/C++, PHP, JavaScript, HTML5, Python, Groovy [17]. Середовище може бути встановлене і для підтримки окремих мов, і у повній конфігурації. Середовище розробки NetBeans за замовчуванням підтримує розробку для платформ J2SE і J2EE.

РОЗДІЛ 2. РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУВАННЯ ПЕРЕВІРКИ РІВНЯ УНІКАЛЬНОСТІ ТЕКСТОВОГО КОНТЕНТУ

2.1 Проектування програмного застосування

Проектування даного програмного застосування має базуватися на використанні сучасних технологій та засобів, найбільш поширеними з яких на практиці є мова UML та програмний засіб Rational Rose. Починати проектування будемо з діаграми варіантів використання програмного забезпечення.

У створюваному програмному застосуванні користувач повинен мати можливість вводити текст для аналізу як власноруч, так і мати можливість вибрати один чи декілька файлів з текстом, що необхідно перевірити, а також мати можливість вибрати каталог, всі файли якого повинні бути або перевірені на унікальність, або текст яких має бути семантично проаналізований.

Користувач повинен мати можливість очистки робочого простору від використаної інформації, а також мати можливість задавати максимальну кількість слів для пошукової вибірки.

У разі необхідності, користувач повинен мати можливість збереження результатів виконання програми до файлу та вибирати каталог для його збереження.

Також, необхідно реалізувати можливість перегляду діаграми унікальності та діаграми частоти значущих слів введеного тексту чи тексту вибраних файлів.

Також, необхідно реалізувати можливості перегляду загальної інформації по створеному програмному продукту, зокрема, загальний опис призначення програмного забезпечення та інформацію про розробника роботи.

Спроектована діаграма варіантів використання програмного застосування наведена на рис.2.1.

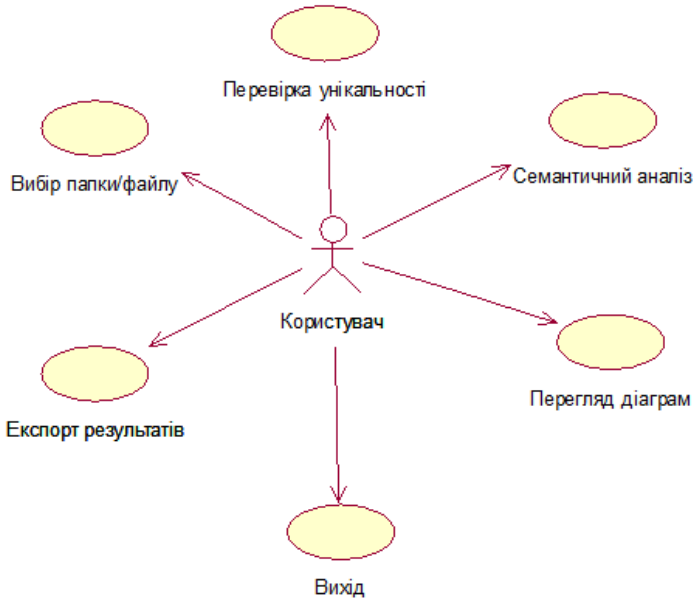

Рисунок 2.1 – Спроектована діаграма варіантів використання програмного застосування

Розроблена діаграма класів програмного застосування наведена на рис.2.2.

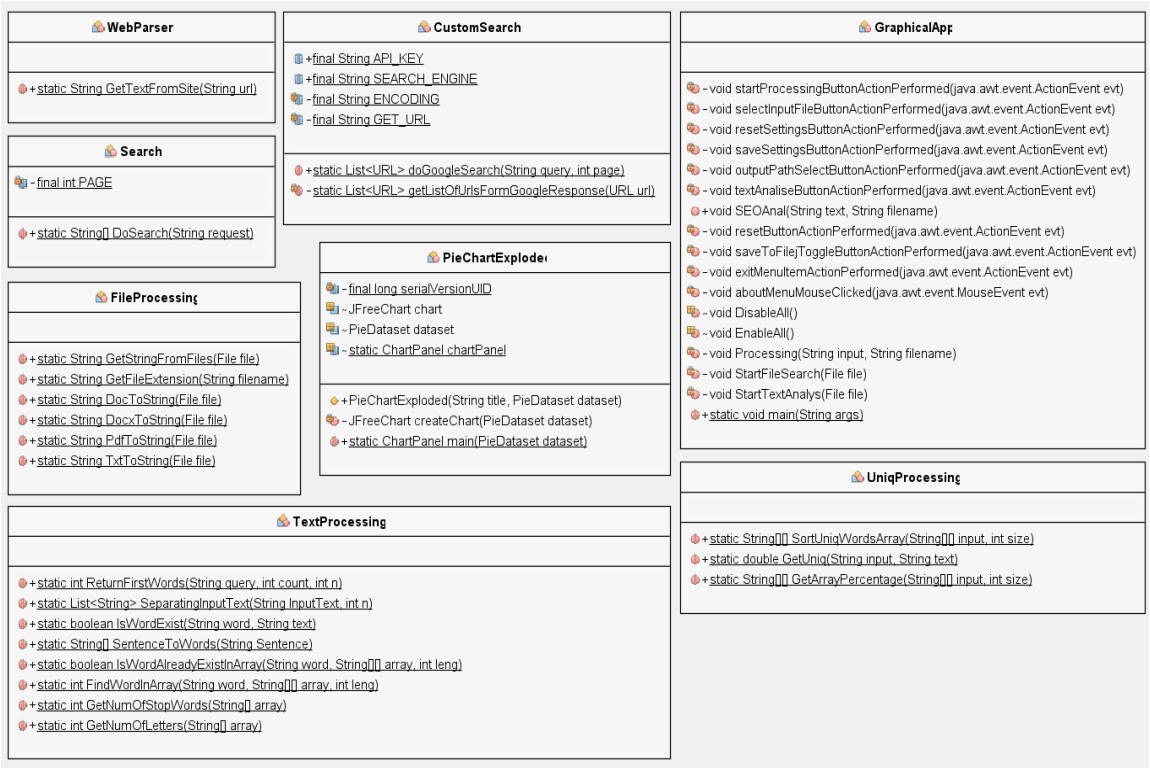

Рисунок 2.2 – Діаграма класів програмного застосування

Розроблена діаграма послідовності програмного застосування наведена на рис.2.3.

Користувач вводить у програму текст, який необхідно перевірити на унікальність, далі програма виконує запит до пошукової системи Google та використовує отримані результати пошуку для оцінки унікальності. Потім створюється результуючий набір даних унікальності та відправляється у клас PieChartExploded. Клас будує та повертає форму з діаграмою, далі програма відображає результати перевірки на унікальність та діаграму унікальності користувачеві.

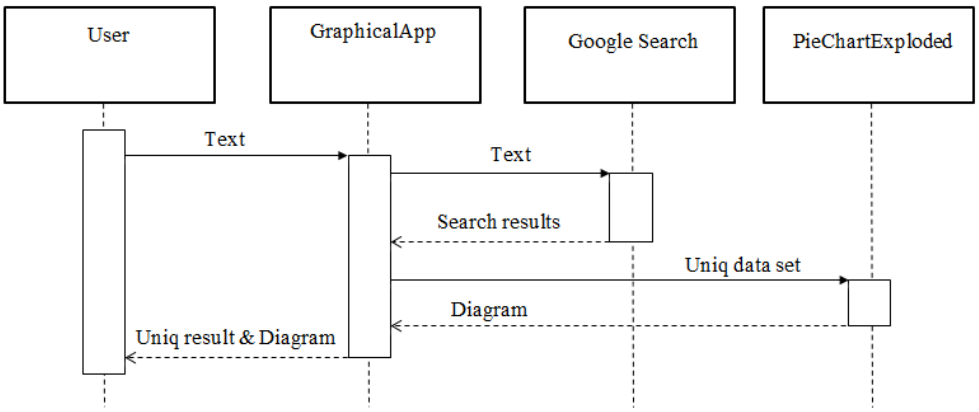

Рисунок 2.3 – Діаграма послідовності програмного застосування

2.2 Розробка схеми складу класів та алгоритму оцінки рівня унікальності текстового контенту

Склад структури каталогів проекту розроблюваного проекту наведено на рис. 2.4.

Для зручності зберігання використаних файлів сформовано наступні каталоги:

– diploma.Execute, містить основні програмні класи реалізації графічного інтерфейсу користувача.

– diploma.InternetSearch, містить програмні класи реалізації інтернет пошуку та витягу текстової інформації з веб-сторінок формату HTML.

– diploma.Processing, містить основні програмні класи реалізації функціональних можливостей програмного застосування.

– diploma.Images, зберігає графічні файли, що використовуються для створення більш інформативного інтерфейсу користувача програмного застосування.

– diploma, містить клас, що відповідає за створення графіків та діаграм.

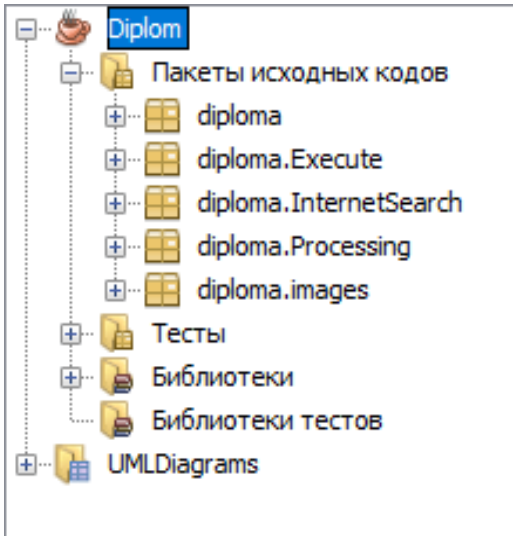

Рисунок 2.4 – Склад структури каталогів проекту

Загальна схема алгоритму оцінки рівня унікальності текстового контенту наведена на рис.2.5.

Алгоритм оцінки рівня унікальності текстового контенту містить наступні етапи:

1. Користувач вводить у програмне застосування початковий текст або вибирає файл, групу файлів чи директорію, де міститься початковий текст, що необхідно перевірити на унікальність.

2. Програмне застосування розбиває початковий текст чи текст з файлів на так названі «текстові відбитки», і на кожен відбиток, за допомогою інтерфейсу Google Custom Search Application Programming Interface, робить пошуковий Інтернет-запит, та отримує у результаті файли з результатами

пошуку збігу у середовищі Інтернет у форматі JSON (JavaScript Object Notation).

3. З JSON-файлу витягається список посилань, код сайтів цих посилань завантажується у пам'ять, щоб після цього відокремити з них текстову інформацію.

4. Отримана текстова інформація порівнюється з початковим текстом, що ввів користувач, та підраховується процент унікальності кожної окремої частини тексту.

5. Програмне застосування підраховує остаточний рівень унікальності як середнє арифметичне унікальності кожної окремої частини початкового тексту.

6. Програмне застосування будує графік унікальності, де враховує унікальність кожної частини та сайт, з якого, імовірно, був скопійований початковий текст.

7. Програмне застосування відображає результати оцінки окремих частин тексту, загальний результат та графік у зрозумілій користувачу формі.

2.3 Реалізація функціональних можливостей програмного продукту

Виконання розробки усіх передбачених створеною діаграмою варіантів використання буде наведено далі, спочатку розглянемо розробку основних програмних методів, після цього буде наведено процес створення графічного інтерфейсу користувача та здійснено опис покрокової взаємодії користувача з розробленим програмним застосуванням.

Рисунок 2.5 – Загальна схема алгоритму перевірки рівня унікальності текстового контенту

2.3.1 Розробка інтерфейсу користувача

Приклад процесу створення головної форми програмного застосування у середовищі розробки наведено на рис.2.6.

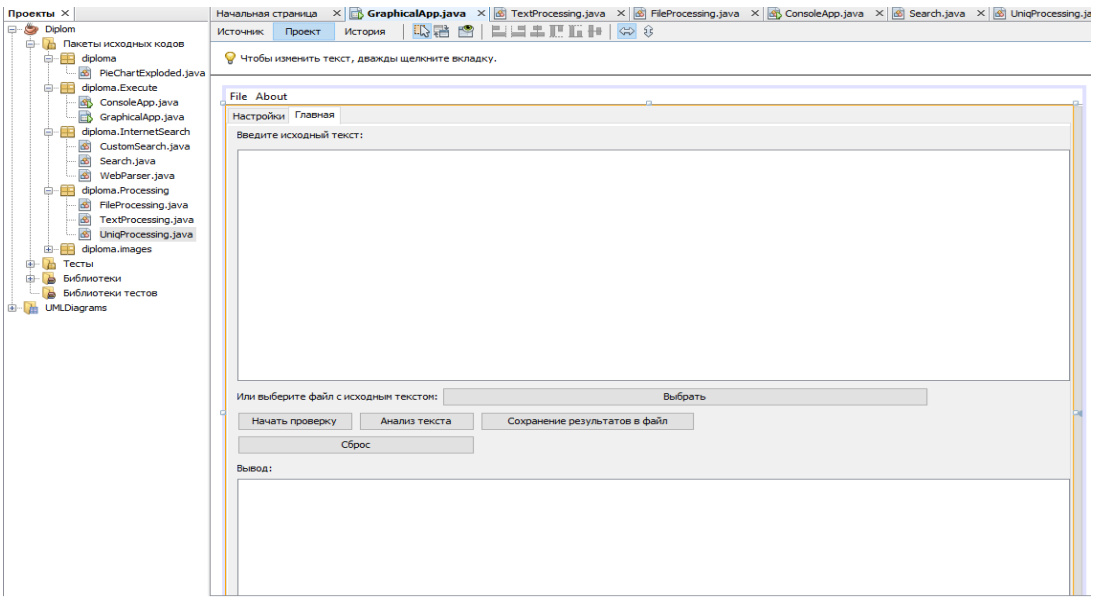

Рисунок 2.6 – Приклад процесу створення головної форми програмного застосування у середовищі розробки

За допомоги палітри компонентів та графічної бібліотеки swing було створено головну форму програмного застосування, яка зберігається у файлі формату *.java.

Панелі оглядача рішень дозволяють обрати необхідний файл інтерфейсу для його обробки та зміни, панель налаштувань дозволяє коригувати властивості активного компоненту форми інтерфейсу, класовий навігатор дозволяє швидко переходити до коду того чи іншого методу або полю.

Форма налаштувань створюється аналогічним чином.

Приклад створення форми налаштувань програмного застосування у середовищі розробки наведено на рис.2.7.

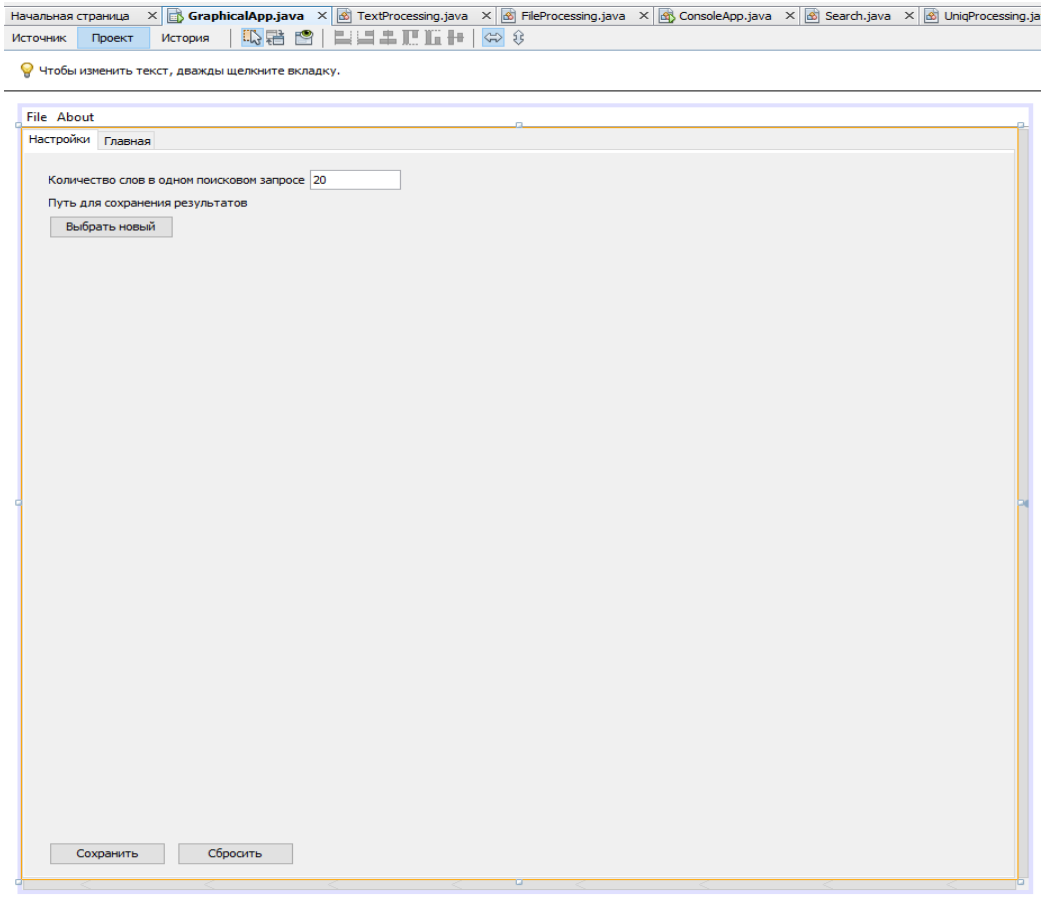

Рисунок 2.7 – Форма налаштувань програмного застосування

Форма перегляду інформації про розробника програмного застосування була створена за допомогою методу JOptionPane.showMessageDialog.

2.3.3 Опис процесу взаємодії користувача з програмним застосуванням

Головна форма розробленого програмного застосування перевірки рівня унікальності текстового контенту наведена на рис.2.8.

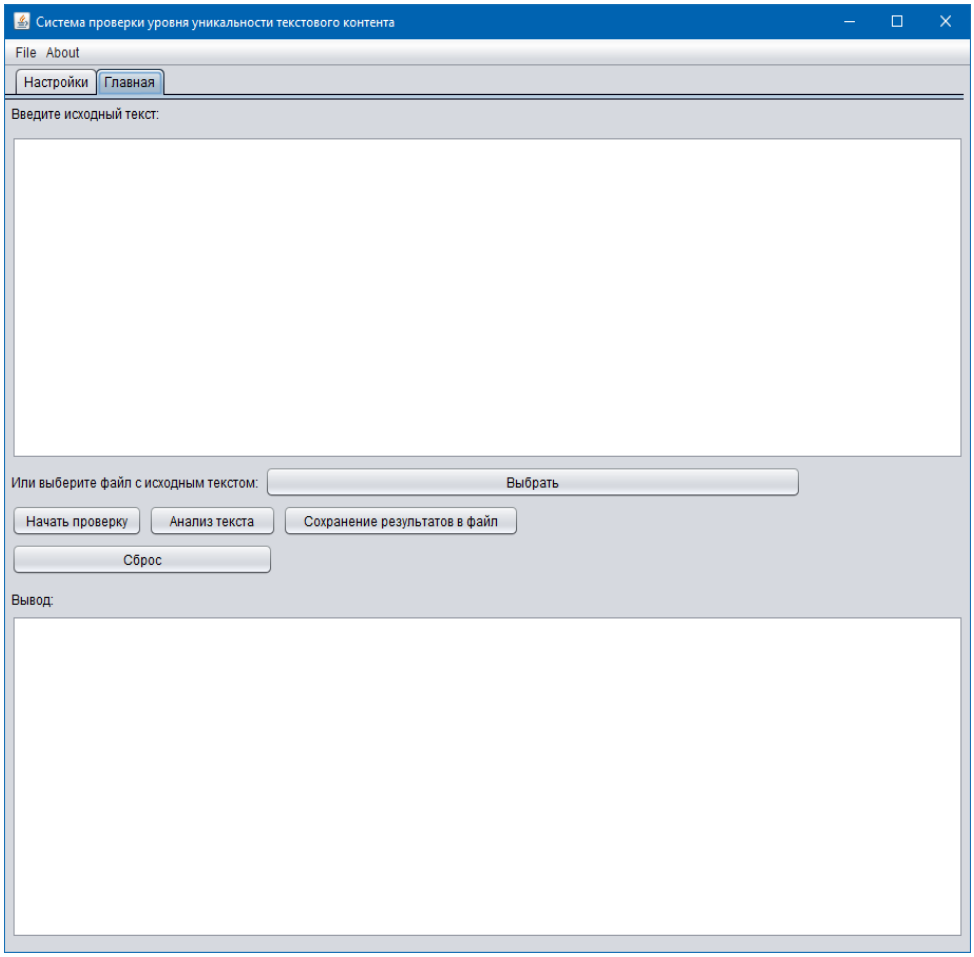

Рисунок 2.8 – Головна форма розробленого програмного застосування перевірки рівня унікальності текстового контенту

На базі створеної діаграми класів програмного застосування на головній формі було виконано розміщення відповідних кнопок, де, натискання на кожну з яких, здійснює виклик відповідного програмного класу чи методу.

У разі обирання пункту встановлення налаштувань здійснюється перехід на форму конфігурації. Користувач, шляхом використання функціоналу описаних вище класів, має змогу вибрати шлях для збереження файлу-звіту на максимальну кількість слів при на пошуку тексту у пошуковій системі Google

(від 1 до 30 слів). Також, користувач має можливість скинути усі налаштування на стандартні, натиснувши відповідну кнопку.

Приклад заповнення інформації у формі налаштування розробленого програмного забезпечення наведемо на рис.2.9.

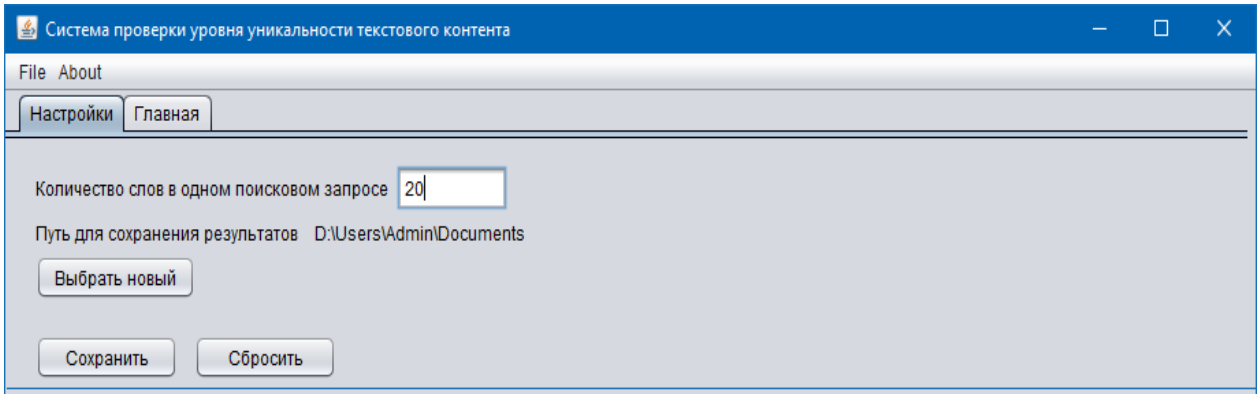

Рисунок 2.9 – Приклад заповнення інформації у формі налаштувань розробленого програмного застосування

Користувач програмного застосування має можливість як вводити необхідний для аналізу текст у відповідне поле на головній формі, так і натиснути кнопку вибору файлу, групи файлів чи директорії, що зберігаються на диску.

При натисканні на кнопку вибору файлу з вхідними даними, програмне застосування здійснює створення та відображення на головній формі вікна вибору файлу, групи файлів чи директорії, що містить файли з текстовим контентом, що потребує перевірки на унікальність або потребує семантичного аналізу.

Вікно обирання текстових файлів для імпорту до програмного застосування наведено на рис.2.10.

Програма має головне меню з двома пунктами – File та About. У підменю File є одна кнопка – Exit (виконує метод виходу з програми).

При натисканні на кнопку About, відобразиться вікно у інформацією про програмне застосування та її розробника.

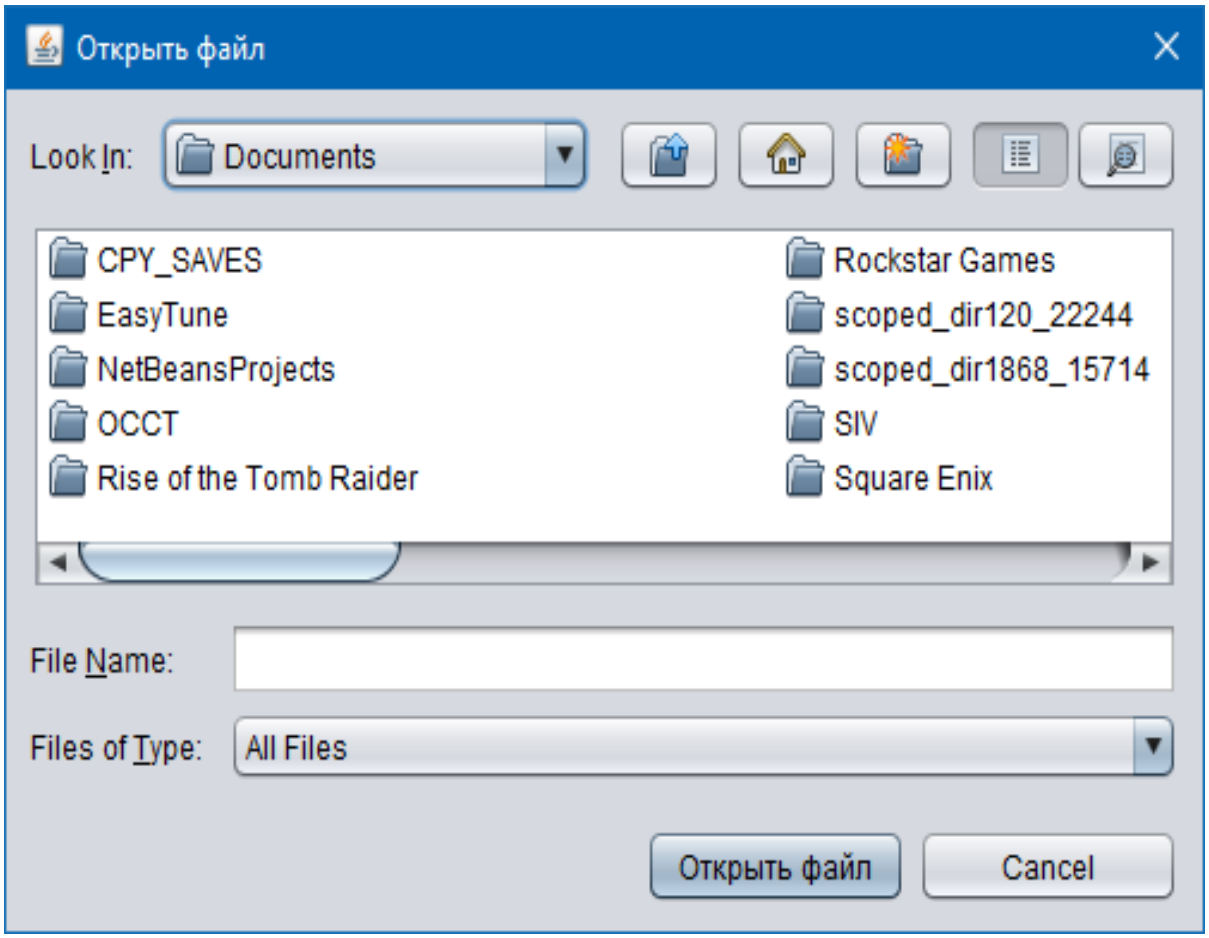

Рисунок 2.10 – Вікно обирання текстових файлів для імпорту до програмного застосування

На головній формі є кнопка-перемикач для збереження результатів виконання програмного застосування.

Якщо кнопка-перемикач знаходиться у натиснутій позиції, то усі результати виконання програми будуть записані у файл, який буде збережений у вибраний в налаштуваннях каталог для збереження вихідних файлів.

Також на головній формі є кнопка очищення, що очищує усі поля програмного застосування.

Результат виконання перевірки рівня унікальності тексту з неунікальним текстом, що був взятий з одного Інтернет-джерела, наведено на рис 2.11.

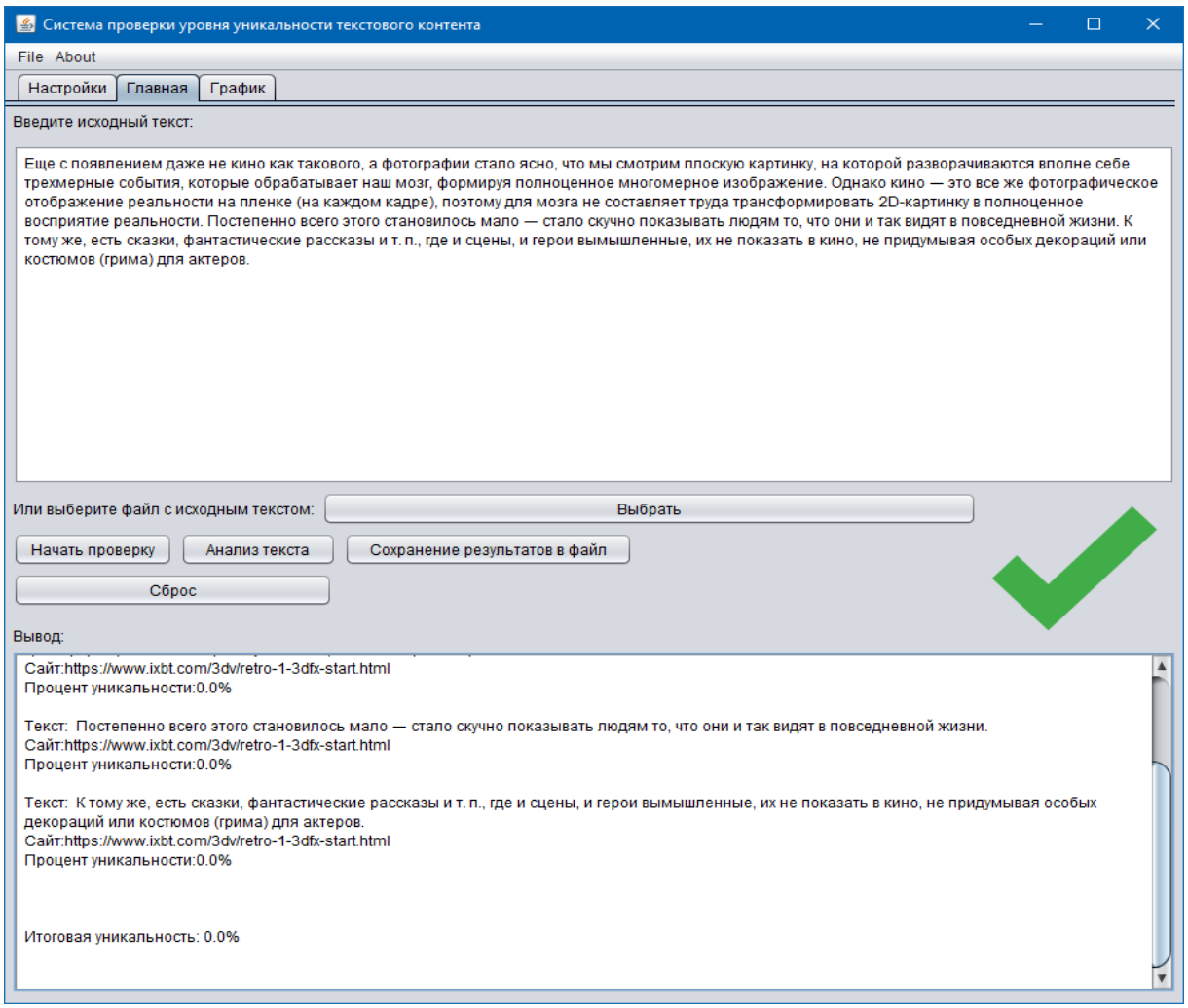

Рисунок 2.11 – Результат виконання перевірки рівня унікальності тексту з неунікальним текстом, що був взятий з одного Інтернет-джерела

Після введення вхідного тексту і натисканні кнопки початку перевірки програмне застосування почне процес перевірки введеного тексту на унікальність.

Під час виконання перевірки унікальності, у форму з вихідним текстом поступово буде виводитись інформація про прогрес виконання процесу перевірки та коли перевірка завершиться, у вихідне поле (та файл, якщо перемикач збереження до файлу увімкнений) буде виведено результат, а також буде створено нове вікно з діаграмою результатів.

Результат виконання перевірки рівня унікальності тексту з частково унікальним текстом та запозиченням тексту з декількох джерел наведено на рис.2.12.

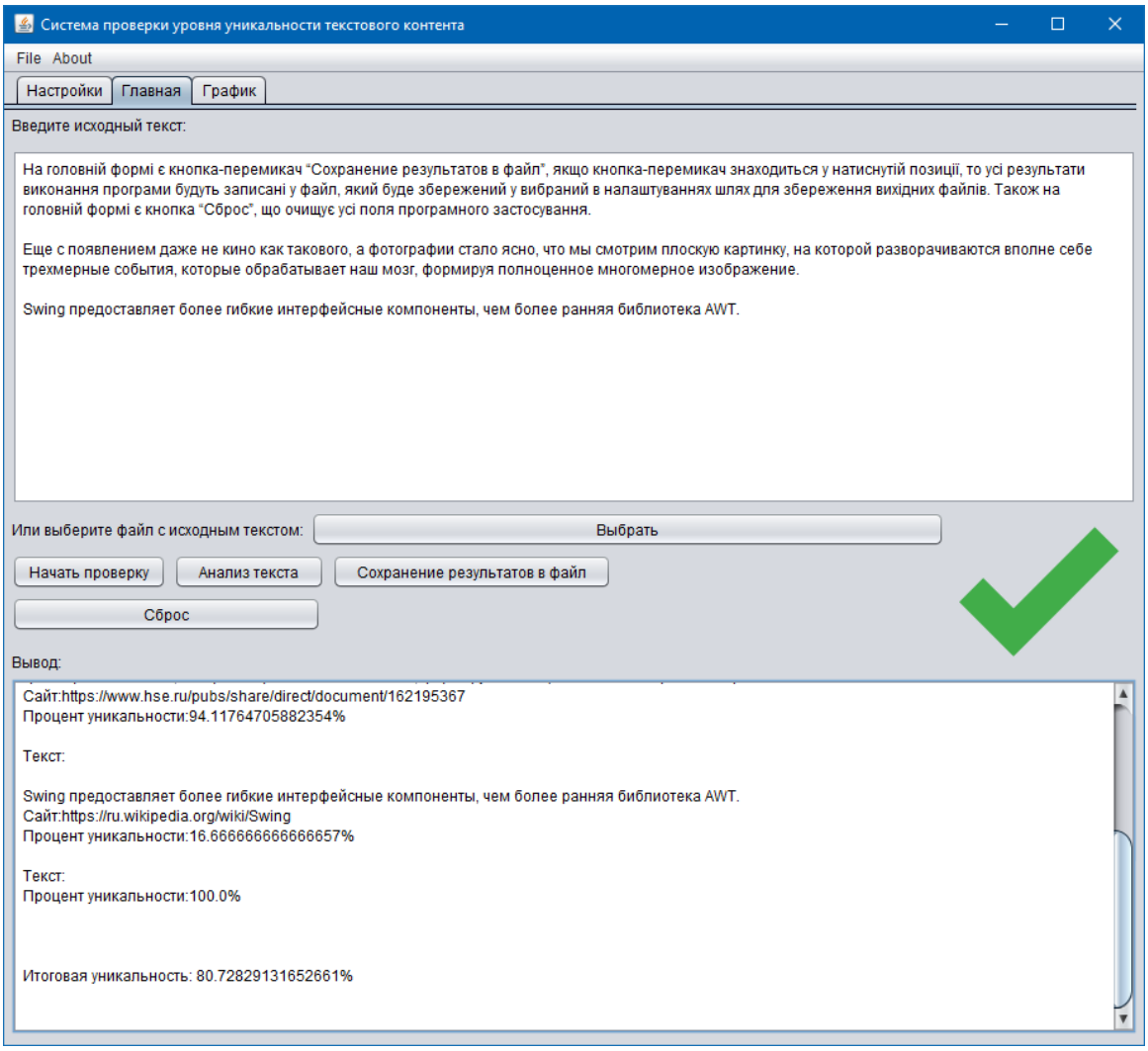

Рисунок 2.12 – Результат виконання перевірки рівня унікальності тексту з частково унікальним текстом та запозиченням з декількох джерел

Якщо користувач скопіював текст не з одного Інтернет-джерела, а, наприклад, з декількох різних, програмне застосування відобразить усі Інтернет-посилання, з яких користувач, імовірно, скопіював початкових текст.

Якщо користувач ввів унікальний, авторський текст, що не зустрічається в інтернеті, то програма відобразить рівень унікальності у 80 чи вище відсотків.

Діаграма унікальності, що створена за результатами перевірки рівня унікальності тексту з частково унікальним текстом та запозиченням тексту з декількох джерел наведено на рис.2.13.

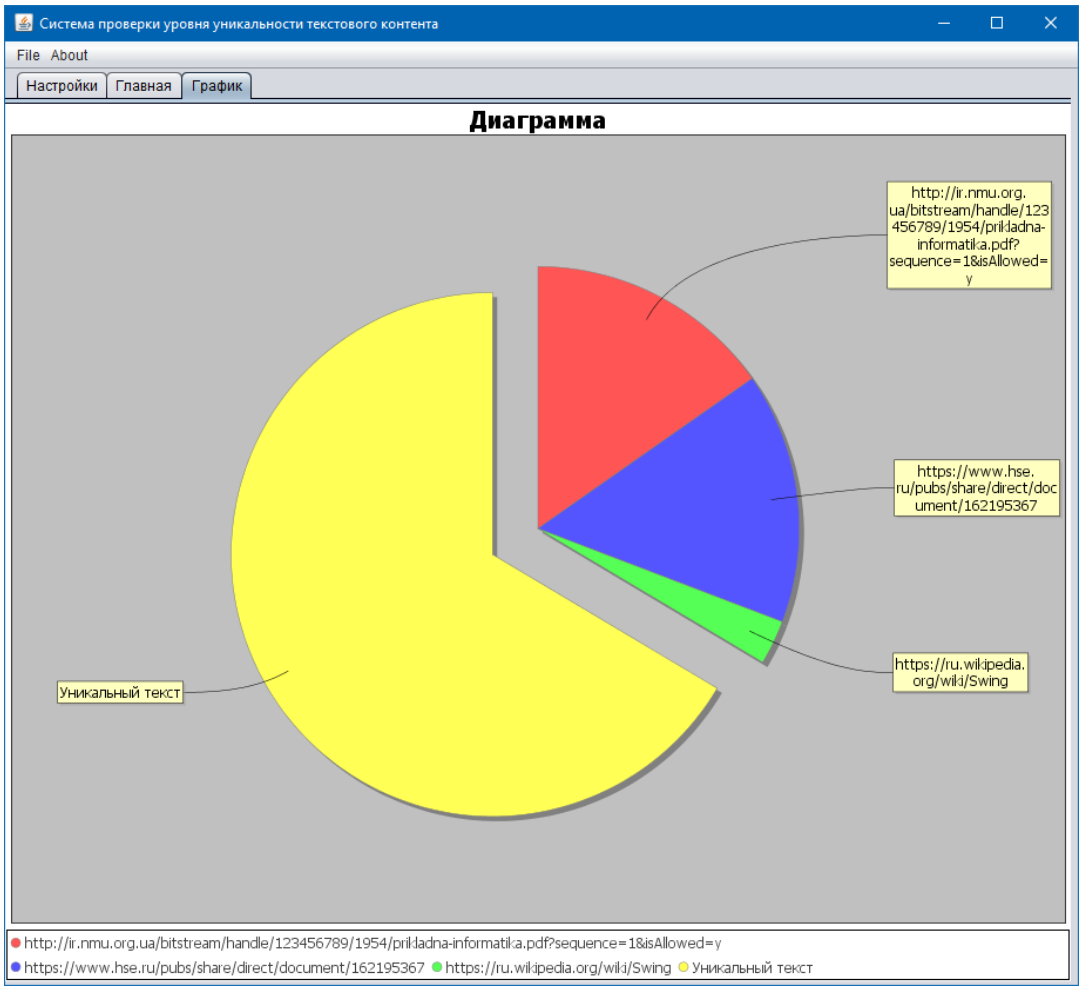

Рисунок 2.13 - Діаграма унікальності, що створена за результатами перевірки рівня унікальності тексту з частково унікальним текстом та запозиченням тексту з декількох джерел

Частина діаграми з унікальним текстом відокремлюється від частини з посиланнями. Легенда діаграми відображується унизу форми.

Користувач має можливість визвати контекстне вікно правою кнопкою миші, де він має можливість налаштувати відображення діаграми, має можливість зберегти зміст діаграми у графічний PNG-файл, скопіювати діаграму, а також має можливість роздрукувати діаграму унікальності чи діаграму частоти слів на принтері та має змогу вибрати масштаб відображення даних на діаграмі.

У налаштуваннях діаграми користувач має можливість змінити текст заголовку на довільний (чи виключити його), вибрати шрифт і його розмір, та

атрибути (жирний, курсивний), вибрати стиль діаграмної рамки, її окрас та фоновий колір, а також включити/виключити згладжування контурів діаграми.

Вікно налаштувань діаграми наведено на рис 2.14.

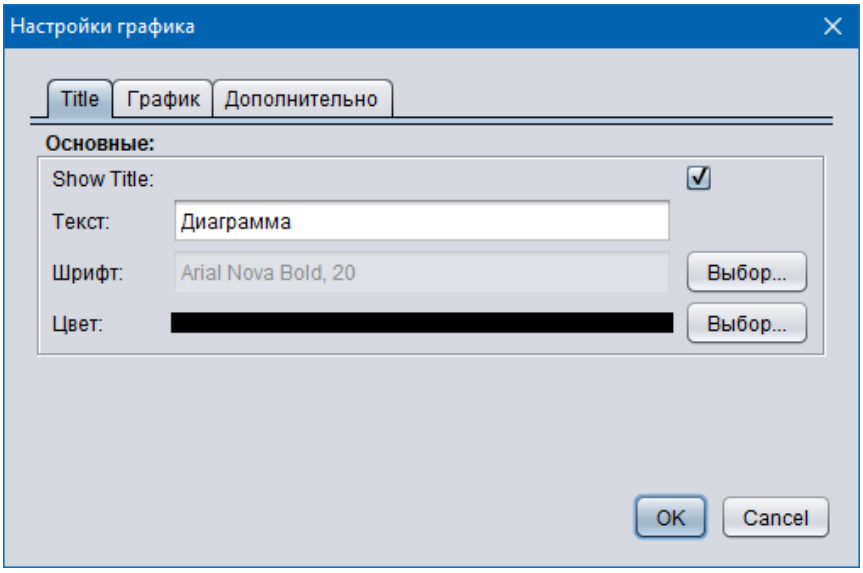

Рисунок 2.14 – Вікно налаштувань діаграми

ВИСНОВКИ

В рамках виконання даної роботи було виконано наступні поставлені завдання:

1. Був проведений аналіз та дослідження предметної області у сфері унікальності текстового контенту.

2. Був виконаний аналіз літературних джерел з питань складу існуючих технологій аналізу унікальності текстового контенту.

3. Був здійснений аналіз можливостей, переваг та недоліків існуючих програмних аналогів.

4. Було обґрунтовано використання програмних засобів розробки.

5. Виконана постановка мети та завдань дослідницької роботи.

6. Розроблений проект програмного забезпечення.

7. Розроблені схеми взаємозв’язків та викликів між класами.

8. Розроблені методи програмної реалізації основних функціональних можливостей.

9. Розроблений інтерфейс програмного забезпечення.

Таким чином, мета роботи була досягнута.

Результати роботи опубліковано у збірнику тез XV всеукраїнської конференції студентів і молодих науковців «Інформатика, інформаційні системи та технології», що проходила 27 квітня 2018р.

Програмна розробка застосовується в учбовому процесі для перевірки на плагіат курсових та дипломних робіт.

ПЕРЕЛІК ПОСИЛАНЬ

1. 10 способов проверить текст на уникальность [Електронний ресурс]. – Режим доступу: https://raskrutka.com.ua/blog/10-sposobov-proverit-tekst-na-unikalnost/. – Дата доступу: 16.04.2018.

2. Что такое уникальность текста? [Електронний ресурс]. – Режим доступу: https://wiki.rookee.ru/unikalnost/. – Дата доступу: 16.04.2018.

3. Великий тлумачний словник сучасної української мови [уклад. В.Т.Бусел]. — К.: Ірпінь: ВТФ «Перун», 2005. – 1728 с.

4. Ушакин С. Плагиат? Об этике в науке / С. Ушакин. – М.: Общественные науки и современность, 2001. – 191 с.

5. Выявление плагиата [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Выявление_плагиата/. – Дата доступу: 16.04.2018.

6. Техническая и смысловая уникальность – что это такое? [Електронний ресурс]. – Режим доступу: https://wordfactory.ua/texnicheskaya-i-smyslovaya-unikalnost-tekstov-chto-eto-takoe/. – Дата доступу: 16.04.2018.

7. Имеет ли значение дублирование контента на сайте? [Електронний ресурс]. – Режим доступу: https://seo-akademiya.com/baza-znanij/kontent/dublirovanie-kontenta/. – Дата доступу: 16.04.2018.

8. Brin S. Copy Detection Mechanisms for Digital Documents / S. Brin, J. Davis, H. Garcia-Molina – Stanford, 2001. – 5 с.

9. Leong A. Check: A Document Plagiarisment Detection System / A. Leong, H. Lau, W.H. Rynson – Hong Kong, 1997. – С. 70-77.

10. Gipp B. Citation Pattern Matching Algorithms for Citation-based Plagiarism Detection / B. Gipp, N. Meuschke // Greedy Citation Tiling, Citation Chunking and Longest Common Citation Sequence. – ACM, 2011. – С. 249-258.

11. Стилометрия [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Стилометрия/. – Дата доступу: 20.04.2018.

12. Ашманов И. Оптимизация и продвижение сайтов в поисковых системах / И. Ашманов, А. Иванов - СПб: Пітер, 2011. – 427 с.

13. "Etxt Антиплагиат" - программа проверки текста на уникальность для Windows – плюсы и минусы [Електронний ресурс]. – Режим доступу: http://otzovik.com/reviews/etxt_antiplagiat-programma_proverki_teksta_na _unikalnost_dlya_windows/. – Дата доступу: 22.04.2018.

14. Antiplagiat.ru – обзор сервиса [Електронний ресурс]. – Режим доступу: http://actualtraffic.ru/site/antiplagiat/. – Дата доступу: 22.04.2018.

15. Advego антиплагиат онлайн [Електронний ресурс]. – Режим доступу: https://advego.com/antiplagiat/. – Дата доступу: 01.05.2018.

16. Обзор программы Advego Plagiatus [Електронний ресурс]. – Режим доступу: http://vsetyrabota.ru/napisanie-statej/77-obzor-programmy-advego-plagiatus/. – Дата доступу: 01.05.2018.

17. Böck H. The Expert's Voice in Java / H. Böck // The Definitive Guide to NetBeans Platform 7. – Tübingen, 2012. – C. 29-30.

АНОТАЦІЯ

на конкурсну наукову роботу під шифром «Text Uniqueness»

В науково-дослідній роботі об'єктом дослідження є унікальність інформації, предметом дослідження – проведення розгляду та аналізу перевірки текстової інформації на унікальність з метою запобігання плагіату. Відкритий доступ до літератури чи текстового контенту в мережі Інтернет, а також застосовування принципу Copy&Paste призвели до появи робіт, що дублюють одна одну. Вважаю, що оскільки виконана робота зі створеним додатком допоможе виявити таке дублювання, то вона є актуальною.

Мета роботи - розробити програмне забезпечення для оцінки рівня унікальності текстового контенту для забезпечення функцій швидкої та якісної перевірки унікальності заданого тексту.

Завданнями роботи є проведення аналізу особливостей та призначення перевірки рівня унікальності текстового контенту, проведення аналізу існуючих методів перевірки рівня унікальності тексту та існуючих програмних продуктів з перевірки контенту на плагіат, обґрунтування програмних засобів розробки, розробка UML діаграм проекту програмного застосування, розробка алгоритму роботи програми, програмна реалізація програмного забезпечення перевірки рівня унікальності текстового контенту.

В роботі використані методи дослідження: аналіз, синтез, порівняння та моделювання.

Практична значущість: результати даної роботи можно використовувати в учбовому процесі при перевірці кваліфікаційних (курсових і дипломних) робіт на плагіат, а також розширити перевірку на плагіат рецензованих оригінальних наукових та оглядових статей в наукових журналах.

Конкурсна наукова робота представлена на 29 аркушах, складається з 2 розділів, 3 підрозділів, 1 таблиці, 1 діаграми, 3 uml-діаграм, 1 блок-схеми, 17 рисунків, з яких 9 відображають роботу програмного додатку. Список використаної літератури складається з 17 джерел.

Ключові слова: унікальність контенту, плагіат, комп’ютерні методи виявлення плагіату.