

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ SEO-
АНАЛІЗУ ВЕБСАЙТУ
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS FOR THE
SEO-ANALYSIS OF A WEBSITE

Виконав: студент(ка) 2 курсу, групи КНТ-210м
Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)
Системи штучного інтелекту

Москаленко О.О.

(прізвище та ініціали)

Керівник Касьян М.М.

(прізвище та ініціали)

Рецензент Поляков М.О.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ПРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти магістр
 Спеціальність 122 Комп'ютерні науки
(код і найменування)
 Освітня програма (спеціалізація) Системи штучного інтелекту
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 20 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Москаленко Олега Олександровича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів SEO-аналізу вебсайту. Research and Software Implementation of Methods for the SEO-analysis of a Website.

керівник проєкту (роботи) Касьян Микола Миколайович, к.т.н., доцент.
прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “5” листопада 2021 року №435

2. Строк подання студентом проєкту (роботи) 1 грудня 2021 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Дослідження методів SEO-аналізу. 3. Розробка архітектури сервісу. 4. Основні рішення щодо реалізації компонентів системи. 5. Тестування SEO-аналізу. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Касьян М.М., доцент		
Нормоконтролер	Белова А.В., асистент		

7. Дата видачі завдання 9 вересня 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітки
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2 тиждень	Розділ 1
3	Розробка та удосконалення методів, моделей й алгоритмів вирішення задачі.	3-4 тижні	Розділ 2
4	Розробка архітектури програми.	5 тиждень	Розділ 3
5	Розробка програми.	6-7 тижні	Розділ 4
6	Тестування програмного забезпечення.	8 тиждень	Розділ 5
7	Оформлення пояснювальної записки та документів до неї.	9-10 тижні	Додатки
8	Нормоконтроль та рецензування.	11-12 тижні	
9	Захист роботи.	13 тиждень	

Студент (ка) _____ Москаленко О.О.
(підпис) (прізвище та ініціали)

Керівник проекту (роботи) _____ Касьян М.М.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи магістра: 91 с., 6 табл., 44 рис., 2 дод., 37 джерел.

АНАЛІЗ, ВЕБСАЙТ, ВОДНІСТЬ, ЗАСПАМЛЕНІСТЬ, МАШИНЕ НАВЧАННЯ, ФРЕЙМВОРК, УНІКАЛЬНІСТЬ, ЯКІСТЬ ТЕКСТУ, DJANGO, PYTHON, SEO.

Об'єкт дослідження – SEO-аналіз вебсайту.

Предмет дослідження – SEO-аналіз тексту вебсторінки.

Мета роботи – провести дослідження методів SEO-аналізу тексту вебсторінки сайту та розробити програмне забезпечення автоматичної перевірки тексту.

Матеріали, методи та технічні засоби: мова програмування Python, бібліотеки PyPi, машинне навчання для визначення тональності та рекламності тексту, фреймворк Django.

Результати. На основі аналізу предметної області та дослідження методів SEO-аналізу тексту розроблено програмний застосунок, призначений для підвищення оптимізації сайту та просування його в пошуковій системі.

Висновки. Розроблена система дозволяє автоматизувати перевірку тексту за рядом параметрів та покращити його якість, тим самим збільшити рейтинг сайту у пошукових системах.

Галузь використання – інтернет-магазини.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 91 pages, 6 tables, 44 figures, 2 appendices, 37 sources.

ANALYSIS, WEBSITE, WATER, WATERING, MACHINE LEARNING, FRAMEWORK, UNIQUENESS, TEXT QUALITY, DJANGO, PYTHON, SEO.

The object of research is SEO-analysis of the website.

The subject of research - SEO-analysis of the text of the web page.

The purpose of the work - to conduct research on SEO-analysis of the text of the web page of the site and to develop software for automatic text verification.

Materials, methods and technical means: Python programming language, PyPi libraries, scikit-learn (machine learning), Django framework.

Results. Based on the analysis of the subject area and research methods of SEO-analysis of the text developed a software application designed to improve site optimization and promotion in the search engine.

Conclusions. The developed system allows you to automate the verification of text on a number of parameters and improve its quality, thereby increasing the site's ranking in search engines.

Field of use - online stores.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ.....	9
1 Аналіз предметної області	12
1.1 Принципи пошуку вебсайтів	12
1.2 Алгоритми ранжування вебсайтів	15
1.3 SEO-оптимізація вебсайтів	19
1.4 Висновки до розділу	24
2 Дослідження методів SEO-аналізу	25
2.1 Внутрішня SEO-оптимізація вебсайту	25
2.2 Дослідження методів SEO-аналізу тексту вебсторінки	28
2.3 Висновки до розділу	38
3 Розробка архітектури сервісу	39
3.1 Клієнт-серверна архітектура.....	39
3.2 MVC-паттерн	47
3.3 Формальне визначення критеріїв SEO-аналізу	48
3.4 Висновки до розділу	53
4 Основні рішення щодо реалізації компонентів системи	54
4.1 Вибір мови програмування	54
4.2 Алгоритми обробки тексту	57
4.3 Машинне навчання для оцінки рекламності та тональності тексту	62
4.4 Висновки до розділу	64
5 Тестування SEO-аналізу	65
5.1 Інтерфейс сервісу	65
5.2 Параметри оцінок	67
5.3 Висновки до розділу	68
Висновки.....	70

	7
Перелік джерел посилання	72
Додаток А Програмна реалізація	77
Додаток Б Слайди презентації.....	82

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

HTML	–	Hyper Text Markup Language;
HTTP	–	Hyper Text Transfer Protocol;
SEO	–	Search Engine Optimization;
URL	–	Uniform Resource Locator;
ІТ	–	інформаційні технології;
ІЦ	–	індекс цитування;
ПЗ	–	програмне забезпечення;
ПО	–	пошукова оптимізація;
ПрО	–	предметна область;
ПР	–	пошуковий робот;
ПС	–	пошукові системи.

ВСТУП

SEO це комплекс заходів підвищення позиції сайту в пошукових системах (ПС) для збільшення цільової відвідуваності вебсайту [1]. Основним завданням SEO є залучення якнайбільшої кількості відвідувачів для подальшого досягнення певних цілей, які ставлять перед собою приватні особи або компанії. Пошукова оптимізація (ПО) – це процес, який має на меті збільшити рейтинг сайту в пошукових відповідях.

Останнім часом SEO значно перетворилося і стало своєрідним найпотужнішим інструментом для досягнення корпоративних цілей та завдань бізнесу. Якщо раніше цей інструмент розцінювався як особливий різновид інтернет-спаму, то зараз завданням SEO-оптимізаторів є досягнення реальної авторитетності та популярності ресурсу, а не штучна імітація зростання маси посилань [2]. Щороку способи обчислення неякісного інтернет-контенту посилюються та ускладнюються [3]. Через це методи SEO постійно змінюються і вдосконалюються.

При створенні нових ресурсів: сайтів, інтернет магазинів і т.д., - варто звернути увагу на той факт, яка аудиторія відвідуватиме даний ресурс, а також звідки приходитимуть нові користувачі. Створення нового ресурсу тільки для людей призводитиме до поганої індексації сайту та повільного просування в топ. В даному випадку потрібно знайти золоту середину, за якої інтернет-ресурс буде цікавий людям, а також роботам, які займаються індексацією сайтів (пошуковим роботам - ПР).

Для просування існують різні методи seo-оптимізації, зокрема seo-копірайтинг (від англ. search engine optimization copywriting), що є «технікою створення та редагування текстів для вебсайтів» [4]. Тому метою даної технології, як зазначають дослідники, є не тільки просування сайту в ПС за

запитами, а й зростання конверсії відвідувачів сайту, і, перш за все, якісне донесення інформації покупцям. У зв'язку з цим просування сайту в ПС здійснюється за запитами, основою яких є ключові слова. Правильно підібрані ключі, оптимальна їх щільність у тексті, правильно розставлені теги - запорука успіху. За рахунок цього відбувається збільшення конверсії відвідувачів сайту і реальних покупців. Це досягається тим, що має бути якісне донесення інформації та легкість сприйняття за допомогою грамотного використання мовних засобів [5].

В даний час ПС поступово відходять від ранжування сайтів за ознаками формальної релевантності та переходять до оцінки змістовної цінності, де акцент спрямований на користь ресурсу для аудиторії. Актуальним напрямом досліджень є seo-оптимізація тексту вебсторінки. Для того щоб seo-тексти мали успіх і ранжувалися ПР, вони повинні відповідати певним критеріям, за якими можна скласти правильний текст.

Мета даної магістерської роботи – розробка компонентів SEO-оптимізації сайту.

Реалізація поставленої мети передбачає вирішення таких завдань:

- провести аналіз методів SEO-оптимізації сайту;
- дослідити внутрішню SEO-оптимізацію сайту;
- дослідити принципи SEO-аналізу тексту вебсторінки;
- розробити концептуальну модель застосунку SEO-аналізу тексту вебсторінки;
- провести аналіз існуючого інструментарію проектування програмного забезпечення вебдодатків;
- розробити програмне забезпечення перевірки SEO-параметрів тексту вебсторінки;
- дослідити результати аналізу власного застосунку в порівнянні з

існуючими сервісами.

Предметом дослідження дипломної кваліфікаційної роботи магістра є SEO-аналізу тексту вебсторінки.

Практична цінність роботи полягає в тому, що на підставі досліджень розроблено програмне забезпечення для оцінки SEO-параметрів тексту. Розроблена система дозволяє підвищити продуктивність SEO-оптимізації сайту.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Принципи пошуку вебсайтів

Вдалий вебсайт - це надзвичайно ефективний інструмент, що здатний захоплювати увагу аудиторії. Як і будь-який інший маркетинговий інструмент, заснований на принципі безпосереднього відгуку, перш за все він повинен заінтригувати відвідувача, а потім з спонукати його на певні дії. Однак, багато хто ігнорує цю особливість головної сторінки, що часто призводить до того, що відвідувачі не затримуються на сайті надовго і залишають його, ледь зайшовши [6]. Такі вебсайти практично ніколи не досягають передбачуваного рівня відвідуваності, не кажучи вже про продажі. Зробивши лише кілька змін, простий вебсайт може перетворитися в більш надійний і ефективний інструмент. Важливо пам'ятати, що з дня на день на потенційних клієнтів обрушується потік інформації і різних рекламних повідомлень, і що в плані завоювання їхньої уваги існує гранично жорстка конкуренція. Вебсайт, здатний привернути увагу і викликати цікавість, спонукає клієнтів не тільки переглянути всі сторінки і зробити покупки, але і знову відвідати його через деякий час, а також рекомендувати своїм друзям і знайомим [1].

Для подальшої оптимізації вебсайту наведемо основні поняття ПС.

Індекс. ПС зберігає інформацію про всі проскановані сторінки в індексі. Кожен запис в індексі містить інформацію про контент та розташування (URL) відповідної сторінки. Проіндексувати – завантажити сторінку, обробити контент на ній та додати її до індексу [7].

Сканування – пошук нових та змінених вебсторінок. ПС знаходить URL-адресу, переходячи за посиланнями, зчитуючи інформацію з файлів Sitemap, а також іншими способами. ПС сканує інтернет, щоб знайти нові вебсторінки, а потім за потреби індексує їх [8].

Пошуковий робот (ПР) – спеціальна програма, яка сканує (переглядає) сторінки в інтернеті та індексує їх.

Пошукова оптимізація (ПО) – процес, який покликаний зробити структуру сайту та його контент доступнішим та зрозумілішим для ПС. SEO-фахівець – це фахівець із пошукової оптимізації [9].

Фізично ПС являють собою мережу з сотень тисяч і навіть мільйонів серверів, в буквальному сенсі слова зберігають у собі копію всього Інтернету [2]. Першим важливим компонентом ПС є пошуковий робот, що індексує:

- завантажує сторінки з Інтернету, так само, як це робить браузер;
- після скачування проводить початковий аналіз - чи містить ця сторінка якусь осмислену інформацію і чи потрібно помістити її в базу даних ПС (інакше званого «індексом»);
- якщо сторінка корисна, слід більш докладний аналіз. Сторінка розбирається на складові елементи: текст, html код, посилання, зображення тощо. і потім міститься в індекс.

Індекс ПС працює за таким принципом, що-для кожного слова є список документів, що його містять. Тільки на відміну від книжкового покажчика, індекс ПС значно більш повний і містить не тільки номери документів, але і багато додаткової інформації (як часто слово зустрічалося на сторінці, якими тегами було виділено і т.д.).

Індекс цитування (ІЦ) - показник ПС, обчислюваний на основі кількості посилань на даний ресурс з інших ресурсів мережі Інтернет. У простій різновидності індексу цитування враховується тільки кількість посилань на ресурс [3].

Першою великою пошуковою системою, яка почала активно використовувати індекс цитування, стала Google. ПС використовує дві складові: основний та додатковий індекс. В основному індексі знаходяться найбільш важливі та релевантні, на думку пошуковика, сторінки ресурсу, тобто ті, які

надають найбільш точну та корисну інформацію для користувача [10].

Додатковий індекс – база даних сторінок, які за певних причин не потрапили в основний індекс ПС. Такі сторінки практично не задіяні при видачі пошукової системи на запит користувача. Але, все ж таки, такі сторінки можна побачити, якщо запит користувача досить оригінальний та співпадає з ключовими словами запиту потрібної сторінки.

Основні причини, за яких сторінки ресурсу не потрапляють в основний індекс ПС:

- неунікальний контент;
- замалий розмір тексту на сторінці (менше 1000 символів);
- неякісні вхідні посилання на дану сторінку;
- вихідні посилання на неякісні сайти;
- дублювання тегу <title> на різних сторінках (категорично неприйнятно) та ін.

Зрозуміло, що чим більше сторінок ресурсу знаходяться в додатковому індексі ПС, тим менше шансів буде його знаходження на високих позиціях при запитах користувачів [11].

У ПС всі дані вже підготовлені для пошуку. 99% усієї роботи зроблено ще до отримання запита від користувача – список слів та документів підготовлений заздалегідь. А шукати за допомогою готового індексу в мільйони разів швидше, ніж переглядом інформації в реальному часі [5].

Після того, як первісний список результатів отримано, у справу вступають алгоритми ранжирування. Адже з мільйона знайдених сторінок потрібно вибрати 1000, які найбільше відповідають на запитання. Для відбору цієї тисячі використовуються безліч різних факторів. Текст сторінки, посилання на цю сторінку, поведінка користувачів і т.п. На основі цих даних і формується та 1000 результатів, які ми можемо побачити у своєму браузері [11].

Отже, ПС формує список документів, які найкраще відповідають на

заданий запит. Тепер потрібно їх відсортувати за якістю. На цьому етапі вступають машинні алгоритми обраної ПС. Саме ці алгоритми сортують отримані результати та формують остаточний варіант видачі вебсторінок у браузері.

1.2 Алгоритми ранжування вебсайтів

Ранжування - це процес сортування веб-сайтів у ПС, в результаті якого в топ потрапляють сторінки, які найбільше відповідають на пошуковий запит [9]. Основна мета ранжирування – визначити сайти, які мають максимальну практичну цінність для користувача. Додатково алгоритми ПС видаляють з видачі нецільові сторінки та анулюють сайти з дубльованим контентом.

Релевантність відображає відповідність сторінки, що просувається пошуковому запиту користувача, що дозволяє сайту потрапити в пошукову видачу. А за порядок сортування та позицію у топі відповідають фактори ранжування [12].

В даний час ПС поступово відходять від ранжування сайтів за ознаками формальної релевантності та переходять до оцінки змістовної цінності, де акцент спрямований на користь ресурсу для аудиторії.

Авторитетність, якість сайту та практична користь контенту визначаються за допомогою факторів ранжирування, які дозволяють просувати найцінніші ресурси у топ видачі. Найбільш релевантним пошуковим запитом вважається сайт на першій позиції видачі, далі релевантність ресурсу з оцінки пошукових алгоритмів знижується з кожною позицією пошуку [8].

Позиції ресурсу у видачі можна легко покращити, якщо знати, як працює ранжування сайтів у мережі, та які параметри враховуються ПС. Зараз докладно розберемо фактори ранжирування в Google.

Google дуже вимогливо ставиться до контенту сайтів і регулярно змінює

та вдосконалює свої алгоритми. Як і в будь-якій ПС, алгоритми Google постійно змінюються для того, щоб принести користь користувачам. ПС аналізує поведінку людей на конкретному сайті. Від того чи відвідують сайт, чи залишаються на ньому деякий час, чи діляться посиланнями, Google аналізує, чи гідний цей сайт місця в топі.

Істотний вплив на ранжування сайтів мають багато алгоритмів, запроваджених Google не випадково. Вони суттєво змінюють результати пошукової видачі. Унікальний контент, який зуміє не лише залучити, а й завоювати увагу цільової аудиторії, відіграє важливу роль у просуванні сайту [13].

У лютому 2011 року було запроваджено алгоритм ранжування Google Panda. Він оцінює якість змісту сайтів, визначає рейтинг релевантних сайтів, коригує та приховує спамлені сайти. Google Panda був розроблений для того, щоб автоматично видаляти з пошуку сторінки з неякісним та неоригінальним контентом. Завдяки фільтру пошукова система видає тільки ті сторінки, які точно відповідають запиту, користувач не витрачає час на перегляд сайтів, створених для агресивного seo-просування [14].

Чим корисніший і незаспамлений ресурс, тим вище місце він займає в топі, тоді як сайти з плагіатом і низькоякісним контентом - виключаються зі списку. Найважливішими показниками, на які звертає увагу Google Panda, є унікальність, грамотне оформлення текстів, обсяг контенту, посилання на активні сайти. Відіграє роль та поведінка користувачів на сайті, час, який користувач проводить на ресурсі, кількість переходів, перелінкування, відсоткове співвідношення відмов. Щоб уникнути фільтрації, потрібно публікувати змістовний контент, орієнтуватися на зацікавленість користувача, що він виявляє на сайті.

Penguin - алгоритм, що фільтрує якість посилань. Позбавляючись неякісних посилань, алгоритм збільшує шанси для органічної видачі

релевантних посилань. Запущений в дію 24 квітня 2012 року, часто оновлювався, а з 2016 року працює в режимі реального часу [15].

Алгоритм RankBrain запущено у жовтні 2015 року. Його мета – надати користувачеві релевантні результати [16]. Система машинного навчання дозволяє розшифровувати сенс і краще орієнтуватися у запитах, у яких немає ключових слів для пошуку. RankBrain – це система машинного навчання, що дозволяє Google краще розшифровувати та розуміти зміст запитів користувачів та надавати більш релевантні результати залежно від контексту. Мається на увазі, що штучний інтелект розпізнає не просто різні граматичні форми одного слова, а й синоніми і навіть смислове навантаження слова. Тепер пошукова система може відрізняти семантику слів, завдяки базі даних Knowledge Graph, в якій міститься вся інформація про можливе смислове навантаження слів і різні семантичні зв'язки. Безпосередній фактор ранжирування допомагає аналізувати, наскільки якісним є наповнення сайту. Система самонавчається, ґрунтуючись на популярному контенті користувача. Перевага отримує, звичайно ж, корисний та інформативний сайт. Ця інформація верифікується та вноситься до наступної версії.

Алгоритм Google BERT покращує релевантність за рахунок того, що здатний аналізувати не лише ключові слова чи фрази, а й цілі пропозиції. Раніше Google розпізнавав лише окремі слова, але зараз здатний аналізувати цілі пропозиції з огляду на прийменники. Для цього використовується нейронна мережа, користуючись якою ПС розуміє запити, введені різними мовами. В даний час алгоритм на стадії тестування в США його можна застосувати для 10% запитів. У перспективі буде доступним для більшості країн та мов. Якщо раніше Google аналізував кожне слово у запиті, то тепер він розумітиме не тільки ключові фрази, а й допоміжні, такі як словосполучення та прийменники [16].

Smith - цей алгоритм опрацьовує довгі об'ємні тексти. Удосконалення

цієї системи сприятиме оптимізації сайту. Модернізований інструмент зможе впоратися з набагато довгими пропозиціями, ніж його попередник BERT. Спочатку текст буде поділено на окремі частини, за якими в кожній з них визначатиметься відповідність пошукових запитів. Алгоритм здатний визначати, наскільки пропозиції фрагмента відповідають запиту, і наскільки текст відповідає запиту. Таким чином, SMITH покликаний розуміти весь текст загалом [17].

Алгоритм E-A-T ставить за мету боротьбу за якісний, точний і змістовний контент. Переклад аббревіатури E-A-T означає «Expertise, Authoritativeness, Trustworthiness», що означає «експертність, авторитетність, ймовірність». Для редакторів сайту чи копірайтерів цей алгоритм пропонує видавати лише ту інформацію, в якій людина є професійним фахівцем. Згідно з ним ділитися інформацією потрібно тільки тій, у правдивості якої ви впевнені. Якщо це медичний сайт, то статті має писати спеціаліст вузького профілю та надавати посилання лише на перевірені, оригінальні джерела [18].

Алгоритм YMYL тісно пов'язаний із EAT. Це аббревіатура від Your Money Your Life (ваші гроші – ваше життя). Цей фільтр перевіряє сайти, що публікують медичну, юридичну, фінансову тематику, невизначена інформація з яких загрожує вашому здоров'ю чи фінансовому стану. Вперше цей алгоритм набув чинності у 2018 році. За заявою керівництва, вони розширили теми, які можуть загрожувати безпеці життєдіяльності, завдавати шкоди та стати джерелом маніпуляцій. За концепцією Google може попросити автора статті пред'явити свою експертність у певній тематиці [19].

10 лютого 2021 року Google запустили в роботу Passage Ranking. Завдяки цьому алгоритму ПС видаватиме як результат пошуку всі сайти, що містять навіть часткову інформацію, що відповідає запиту. Навіть якщо решта тексту не буде цікавити користувача, але документ містить вичерпну відповідь на запит, такий ресурс займе більш високе місце у результатах пошуку. Отже, стаття, яку

видає Google на запит, не обов'язково має бути присвячена темі пошукового запиту. Це допоможе тим ресурсам, які раніше погано ранжувалися, адже пошук розпізнаватиме фрагменти тексту та видаватиме їх у рядку результатів [19]

Core Web Vitals (CWV) – складова пошукового алгоритму Google Page Experience. Метою цього фільтра є відокремлення ресурсів, з якими комфортно взаємодіяти користувачам. Створили її на реальному досвіді відвідувачів сайтів. Складовими алгоритму є: визначення швидкості, інтерактивність, індикатор стабільності верстки. Ряд досліджень показали, що швидкість завантаження сайту відіграє важливу роль, адже якщо сайт погано завантажується, користувачу нічого не варто перейти на інший сайт та скористатися інформацією на ньому. Середня швидкість відкриття сайту, прийнятна користувачам – 2,5 – 4 секунди. Сайти, які завантажуються довше 4 секунд, опускаються нижче за ранжуванням [20].

1.3 SEO-оптимізація вебсайтів

Методи ПО можна розділити на три класи відповідно до їх кольору (білий, сірий і чорний), проте останні події у світі ПС дають зрозуміти, що цей поділ дуже умовний - будь-яка маніпуляція певними параметрами вебсайту може бути розцінена ПС як вкрай небажана дія. Так, будь-яка спроба маніпулювання пошуковими результатами заборонена в ліцензії на використання пошукової системи Google [10].

Біла оптимізація, на відміну від поширених методів просування сайтів, виключає застосування будь-яких способів, спрямованих на штучне збільшення позиції в пошуковій видачі. Її процеси побудовані на природному розвитку проєктів, аналізі поведінки споживачів, проведенні PR-кампаній, збільшують кількість цільових відвідувачів та популярність ресурсу серед користувачів

мережі Інтернет.

Усі методи білої оптимізації можна розділити на два основні види: внутрішні та зовнішні. Внутрішня біла оптимізація спрямована на загальне покращення сайту. Сюди входить чітка структуризація, використання унікального контенту, створення Sitemap, внутрішня перелінковка, хороший трафік, високий траст сайту, створення RSS, відстеження неякісних посилань, грамотне складання семантичного ядра та інші подібні заходи [21].

Важливу частину серед них оптимізує контент, тобто розміщення в текстовому наповненні необхідного для оптимальної щільності ключових слів (як правило, 5-7%). Можливе збільшення ваги слів шляхом використання відповідних тегів. Всі ці маніпуляції виконуються зі збереженням якомога природнішого виду текстів. Інакше пошукові системи можуть розцінити їх як спам [14].

До методів білої оптимізації також зараховується робота над адресами сайтів та файлом robots.txt. Донедавна до неї належала і оптимізація мета-тегів. Однак у зв'язку з безліччю спроб оптимізаторів проспамити їх ключовими словами та фразами, які не мають відношення до змісту сторінок, більшість ПС їх не враховує.

Зовнішня біла оптимізація – це поширення інформації про існуючий сайт, що досягається за рахунок проведення рекламних кампаній та інших технологій маркетингу. Просування за допомогою статей.

Останній метод, всупереч поширеній думці, лише умовно ставиться до білого. На даний момент ПС не здатні розпізнавати куплені посилання в текстах, надають широкі можливості оптимізаторам проводити самостійну розкрутку сайтів без остраху накладення на них штрафних санкцій. Але за своєю суттю цей спосіб є штучним нарощуванням показників, порушує правила використання ПС [6].

Переваги. Так як білі методи спрямовані не на спамлювання ПС, а на

надання цікавої інформації, підвищення функціональності при розробці сайтів та рівня зручності роботи з ними, це сприятливо позначається на їхній популярності серед користувачів та забезпечує природний приплив відвідувачів. Але головною перевагою білої оптимізації все ж таки є досягнення стабільних показників, які не залежать від пошукових систем і не втрачають своїх значень при зміні їх алгоритмів.

Сіра оптимізація. Сірі методи ПО - методи оптимізації, які офіційно не заборонені ПС, але можуть бути розцінені як неприродні засоби просування сайту. Сіра оптимізація є основним інструментом для просування сайтів. На відміну від чорних методів, сірі рідко призводять до накладання бана або штрафних санкцій з боку ПС, цу дозволяє фахівцям застосовувати їх для виведення ресурсів на перші позиції за порівняно короткий час [20].

До сірих методів пошукової оптимізації відносяться:

- збільшення частоти ключових слів у текстах статей;
- дорвії без редиректу;
- купівля посилань та багато іншого.

При визначенні релевантності ПС в першу чергу звертають увагу, скільки разів на сторінці зустрічається фраза, тотожна запиту користувача. Цей параметр називається частотою ключового слова. Чим він вищий, тим релевантнішим вважається сайт. Донедавна оптимізатори спеціально збільшували частоту ключових слів до нечитабельності текстів. На даний момент ПС активно борються з подібними методами та знижують ранжування при їх виявленні.

Для визначення частоти ключових слів використовуються спеціальні математичні алгоритми, що вираховують кількість входження запитів на обсяг тексту. У цьому оптимальним співвідношенням вважається 3-5%. Так як ПР не здатні оцінювати тексти з точки зору читабельності, ця обставина і дозволяє оптимізаторам збільшувати частоту ключових слів до певної межі, що, з одного

боку, порушує правила використання ПС, а з іншого - не виходить за межі встановлених ними критеріїв [18].

Дорвей. Цим терміном позначають вебсайти, які створюються виключно з метою успішної індексації. Після попадання на головну сторінку цього вебсайту відвідувач автоматично перенаправляється на сайт, що просувається оптимізатором. Для перенаправлення використовується HTML-тег «redirect». Практика показує, що чорні оптимізатори створюють десятки дорвеїв, істотно навантажуючи апаратні ресурси кожної ПС окремо і підриваючи авторитет пошукових машин загалом. Тому створення дорвеїв жорстко карається, іноді навіть винятком із індексу ПС.

До методів сірої оптимізації належить створення «сірих» та «білих» дорвеїв:

- сірі дорвеї. Їхнім головним завданням є передача головному сайту накопичених показників, тобто виступати в ролі донорів. На відміну від чорних методів, сірі мають осмислений текстовий зміст та за своєю суттю є повноцінними вебсайтами;

- білі дорвеї. Цей вид насамперед несе рекламне навантаження та не порушує ліцензії ПС. При створенні використовується оригінальний контент і розробляється привабливе графічне оформлення. При попаданні на «білий дорвей» відвідувач бачить рекламний текст з посиланнями на сайт, що просувається і при бажанні може переходити на нього.

Купівля посилань є одним з основних способів просування сайтів у ПС і за своєю суттю є розміщенням посилань на сторонніх ресурсах з метою збільшення показників вебсайту. Як правило, для цього використовуються спеціалізовані біржі. ПС намагаються боротися з подібним методом, хоча використання такого методу не може призвести до бану [20].

Сіра оптимізація, на відміну чорної, входить в межі встановлених ПС критеріїв. Саме тому в деяких випадках ПС можуть тимчасово або постійно

заблокувати сайт. До того ж вимоги, які вони висувають до просування ресурсів, постійно змінюються і те, що раніше вважалося законним, може у майбутньому стати вже забороненим.

Чорні методи ПО – методи розкручування сайту, заборонені до застосування ПР. Дві переваги чорного просування (оптимізації):

- для власника сайту – це низькі витрати на самостійне просування сайту;
- для оптимізатора – швидка індексація з виведенням вебсайту у верхні рядки пошукової видачі.

Кількість загальноновживаних методів «чорного» просування та оптимізації сайтів не перевищує десяти. До найпоширеніших способів відносять спамдексинг, клоакінг, свопінг, створення фальшивих сайтів - дорвіїв, різні види спаму, наприклад, лінкбомбінг. Доцільно розглянути кожен із них хоча б коротко [13].

Спамдексинг називається методом чорної ПО, при якому сторінки сайту створюються тільки для індексації ПС. Наприклад, надмірно насичують сторінки пошуковими фразами, які до тематики сайту не мають жодного стосунку. Нетематичні пошукові фрази ховають від відвідувачів, набираючи їх дуже дрібним шрифтом, або літерами того ж кольору, що й фон сторінки. В результаті пошукову фразу бачать тільки роботи, але не бачать відвідувачі, які потрапляють за пошуковою фразою [20].

Клоакінг. Складнішим, а тому менш популярним, але не менш ефективним методом чорного ПО є клоакінг, який ще називають маскуванням. ПР пропонується сторінка з контентом, який повністю відрізняється від контенту, який бачать користувачі. Для цього оптимізатор використовує програми, які обчислюють IP-адреси пошукової машини. При зверненні до сторінки пошукача вміст сторінки підміняється на оптимізований саме під робота.

Свопінг. Це заміна вмісту сторінки після індексації. Сайти, що просувалися свопінгом, знаходяться у верхніх рядках ПС на пошуковий запит у первній індексації. Свопінг використовують оптимізатори, які не зацікавлені у тривалих проєктах.

Лінкбомбінг. У чистому вигляді лінкбомбінг є методом чорної ПО. Хоча спам в цілому, до якого належить лінкбомбінг, відноситься, швидше, до сірих методів. Адже посилальний спам при грамотній організації процесу не завжди карається ПС та ефективний для просування. Суть посилального спаму і, зокрема, лінкбомбінгу, полягає в просуванні сайту за нетематичними пошуковими запитами. Наприклад, якщо інтернет-магазин косметики просувається за допомогою пошукової фрази "онлайн-серіали", то пошуковик може занести магазин у бан, але якщо пошукова фраза вбудована в контент органічно, то негативних наслідків може і не бути.

Усі ПС борються із просуванням сайтів чорними методами оптимізації. Сайти, які виявили у використанні чорних способів оптимізації, найчастіше виключаються з індексу на якийсь термін або назавжди.

1.4 Висновки до розділу

В даному розділі розглянуті принципи пошуку вебсайтів та основні алгоритми ранжування. Наведені види SEO-оптимізації вебсайтів. В результаті аналізу можна зробити висновок, що актуальним фактором білої внутрішньої SEO-оптимізації вебсайтів для ПС є створення якісного, унікального, релевантного основній темі сайту тексту. Такий спосіб дасть змогу збільшити рейтинг сайту при ранжування результатів ПС.

2 ДОСЛІДЖЕННЯ МЕТОДІВ SEO-АНАЛІЗУ

2.1 Внутрішня SEO-оптимізація вебсайту

Внутрішня оптимізація сайту – це комплекс заходів, спрямованих на покращення сприйняття ресурсу ПС та користувачами. Вона включає оптимізацію коду та структури сайту, роботу з контентом та комерційними факторами, аналіз та покращення юзабіліті [22].

Технічне SEO - комплекс робіт, які забезпечують працездатність вебсайту, швидке завантаження та коректну роботу. Перелік робіт, для технічної оптимізації:

- видалення дублів вебсторінок;
- помилки в коді. Перевірити код на наявність помилок (валідність) можна за допомогою онлайн сервісів;
- прискорити завантаження сайту. На швидкість сайту впливає кількість та вага зображень, швидкість завантаження HTML, CSS, JS-файлів та інші фактори;
- налаштувати помилку 404 для неіснуючих сторінок. Це потрібно, щоб користувачі залишалися на вашому сайті після переходу через некоректну URL-адресу;
- сховати технічні сторінки індексації. Файли двигуна вебсайту найкраще закрити за допомогою robots.txt. А сторінки з конфіденційною інформацією – за допомогою метатегів robots зі значеннями «noindex, nofollow»;
- налаштувати заголовки Last Modified та If-Modified-Since. Вони повідомлятимуть пошуковим роботам дату останньої зміни сторінок, що пришвидшить переіндексацію;
- виявити та видалити змішаний контент;

- обрати надійний хостинг, адже він забезпечує працездатність та доступність сайту;

- використовувати мікророзмітки — це допоможе ПС точніше визначити тип контенту на сторінці.

Особливо важливою стає технічна оптимізація сайту у світлі швидких нововведень від Google. У травні 2021 року набувають чинності нові фактори оцінки якості - Core Web Vitals [21]. Це три основні показники, які потенційно впливатимуть на ранжування ресурсу в Google:

- LCP (Largest Contentful Paint) — швидкість завантаження основного (найбільшого та найзначнішого) контенту на сторінці. Оптимальний показник – до 2,5 сек;

- FID (First Input Delay) - час, який минає до можливості першої взаємодії з контентом (кліка або скролінгу). Оптимальний показник – до 100 мс;

- CLS (Cumulative Layout Shift) означає стабільність верстки. Іншими словами, як сильно зсувається верстка в процесі завантаження і чи підвантажуються при цьому блоки, що перешкоджають взаємодії з контентом або вводять в оману користувача. Оптимальний показник дорівнює значенню 0,1.

Структура сайту, орієнтованого на просування у ПС, заснована на семантичному ядрі та аналізі сайтів із топу. За такого підходу кожній групі (кластеру) запитів відповідає початкова сторінка. Найкращій варіант – розробка структури сайту на етапі його проєктування. Зрозуміла та логічна структура сайту важлива з двох причин:

- відвідувачам буде зручно користуватися вебсайтом;

- пошукові системи швидше індексуватимуть сторінки.

Залежно від типу ресурсу структура може бути:

- лінійною — вебсторінки можна переглядати лише послідовно. Це

сайти-презентації, слайд-шоу, фотовізитки тощо. У цьому випадку SEO погано працює, тому для просування використовують контекстну рекламу чи інші методи залучення трафіку;

- ієрархічною — більшість сайтів мають саме таку структуру. Двохрівневими можуть бути візитки, лендинги або спеціалізовані блоги. Найбільш поширена трирівнева структура, яка будується за принципом «Головна»-«Рубрика»-«Запис». Може бути і більше рівнів вкладеності, але з погляду SEO краще не використовувати більше 3-4 рівнів, інакше можуть бути проблеми з індексацією;

- гратчасту — її використовують великі інтернет-магазини з широкою номенклатурою товарів від різних брендів. І тут сторінки нижніх рівнів входять відразу у кілька категорій. Для правильної SEO-оптимізації важливо контролювати дублювання сторінок.

Для великих сайтів послуг та інтернет-магазинів з великим асортиментом найкраще підійде SILO-структура. Це метод проектування архітектури сайту, при якому вебсторінки об'єднуються у великі тематичні розділи та пов'язані усередині цих розділів перелінкуванням. При цьому сторінки нижнього рівня посилаються на сторінки верхнього рівня, забезпечуючи їм більшу вагу [23].

Контент має вирішувати проблеми користувачів, з якими вони прийшли на вебсторінку з пошуку. Він має бути корисним, унікальним та оптимізованим під ПС.

Перерахуємо актуальні на 2021 рік правила оптимізації контенту:

- одна сторінка – один заголовок h1;
- у заголовках h1, підзаголовках h2-h6 та тексті використані ключові фрази з семантичного ядра в природному входженні, без переспаму (зазвичай достатньо 2-4 входження ключів);
- у контенті сторінки використані не тільки високочастотні та

середньочастотні, а й низькочастотні запити, тематичні фрази, синоніми, які допомагають розкрити питання з усіх боків і вирішити проблему користувача;

- вміст сторінки відповідає контенту;
- для кожної сторінки складено метатеги title та description із входженням основних ключів;
- для зображень прописані теги alt з використанням ключових фраз;
- тексти на сайті унікальні;
- тексти структуровані: розбиті на абзаци, списки, цитати та інші елементи форматування;
- у текстах відсутні орфографічні та граматичні помилки;
- обсяг текстів для сторінок каталогу та карток товарів – мінімум 2000 символів без пробілів, для сторінок блогу – від 4000;
- якщо на одній сторінці надто багато контенту на різні теми, краще розподілити його на різні сторінки;
- кількість входжень ключових слів у різні зони документа (заголовки, текст, посилання та ін.) та обсяг контенту краще виявляти на основі аналізу сайтів конкурентів у топі.

2.2 Дослідження методів SEO-аналізу тексту вебсторінки

SEO-аналіз тексту - перевірка на відповідність (передбачуваним) вимогам ПС і, як правило, оцінка з точки зору читання. Він покликаний вирішувати дві основні завдання:

- залучати пошуковий трафік. Оптимізація тексту повинна сприяти покращенню ранжування в ПС, щоб сторінка показувалася на вищих позиціях і за максимальною кількістю різних запитів;
- допомагати користувачеві. Поведінка користувачів відіграє важливу

роль у SEO. Щоб люди довше залишалися на сайті та активніше з ним взаємодіяли, потрібно якісно відповідати на їхні пошукові запити. А зайві слова, загальні фрази, повтори тощо ускладнюють передачу інформації.

Розглянемо параметри «правильного» тексту:

- текст в першу чергу повинен писатися для людей, а не ПР. Якщо наповнити сайт набором ключових слів, без зв'язуючої складової, то даний текст буде заблокований і розцінюватиметься як спам;

- текст має бути унікальним. Якщо текст є копією тексту, взятого з іншого сайту, або ж сумішшю якихось двох текстів, то ПР в першу чергу розмістить вище оригінал, а не копію тексту;

- наявність ключових слів. У використанні ключових слів необхідно дотримуватися балансу. При невеликій кількості ключових слів ПР може не розпізнати текст і визначити текст нерелевантним для запиту. У той же час велика кількість призведе до того, що ПР розпізнає текст як спам і зменшить його рейтинг;

- тематичність. Важливо, щоб написаний текст відповідав тому, що пропонує конкретний сайт;

- об'єм. Для хорошої читабельності тексту і хорошого засвоєння інформації користувачем необхідно дотримуватися оптимального обсягу. Зазвичай - це 300-400 слів, але ці цифри можуть змінюватися в залежності від призначення тексту, а також від тематики сайту;

- структурованість. Достатньо об'ємний - понад 400 слів - текст, не розбитий на частини, що не містить підзаголовків, списків або таблиць, однаково погано сприймається як людьми, так і пошуковими системами.

Для того, щоб виділити сервіс серед існуючих при аналізі тексту, необхідно знати ряд параметрів успішного і якісного тексту [24]. Крім того, треба виділити ті параметри, які можна аналізувати автоматично (табл.2.1).

Таблиця 2.1 – Параметри якісного тексту

№	Показник	Діапазон норми, %	На що вказує	Як досягти оптимального значення
1	Унікальність	90-100	Показує оригінальність тексту	Використовувати новий контент або переробляти вже існуючий
2	Нудота	9	Співвідношення ключових слів до загального обсягу тексту	Видаляти або замінити повторювані слова
3	Водність	40-60	Наявність слів, що не відповідають наданій інформації	Видаляти фрази загального значення
4	Заспамленість	3-5	Частота вживання ключових слів	Використовувати ключові слова помірно

Види нудоти тексту. Класична — витягується квадратний корінь із кількості повторень ключового слова. Цей показник не враховує обсягу тексту, тому не дає повністю об'єктивну оцінку. Академічна — ставлення всіх повторів у тексті, зокрема у різних морфологічних формах, до сумарній кількості слів, помножене на 100% [25].

При надто високому показнику нудоти потрібно змінити текст одним із двох способів:

- поміняти деякі ключові слова на синоніми, займенники, видозмінити частину фраз для того, щоб зменшити нудоту до оптимальних

значень;

- збільшити кількість слів, щоб у результаті ключові слова були розподілені з меншою частотою.

Варто враховувати, що при зменшенні показника нудоти необхідно залишити достатню кількість ключових слів, інакше сайту не загрожуватимуть санкції ПС, але він і не потрапить до топ видачі.

У термінології SEO, водність – відсоткове співвідношення стоп-слів до загальної кількості слів у статті. Якщо коротко, стоп-слова – ті, що не враховуються ПР. Союзи, прийменники, числа, а також слова, які часто повторюються у мережі [26].

Об'єм тексту на сторінці обчислюється кількістю символів без пробілів. Запам'ятованість визначається відсотковим співвідношенням кількості ключових фраз до загального обсягу. Вважається, що показник спамленості не повинен бути вищим за 60%.

Розглянемо найпопулярніші сервіси оцінки тексту.

Text.ru - сервіс представляє цілий набір інструментів для перевірки текстів за багатьма параметрами, такими як унікальність, орфографія, SEO-аналіз, а також, вміє підбирати синоніми шляхом пошуку по вбудованій базі. Приклад результату перевірки через модуль SEO-аналізу наведений на рисунку 2.1. Сервіс має лише дві мови інтерфейсу: російську та англійську [27].

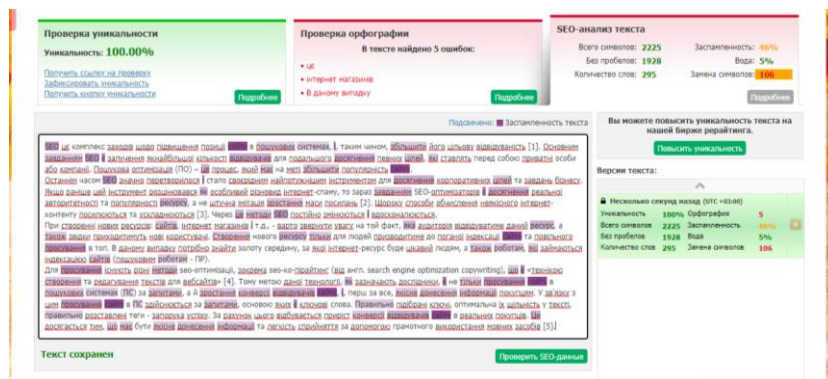


Рисунок 2.1 – Сервіс Text.ru

Як видно на скріншоті, виводяться такі загальні показники як кількість символів (без пробілів) та слів у фрагменті тексту, показники запам'ятованості та «води». Трохи нижче можна дізнатися які ключові слова і наскільки часто вживаються в документі (рис. 2.2).

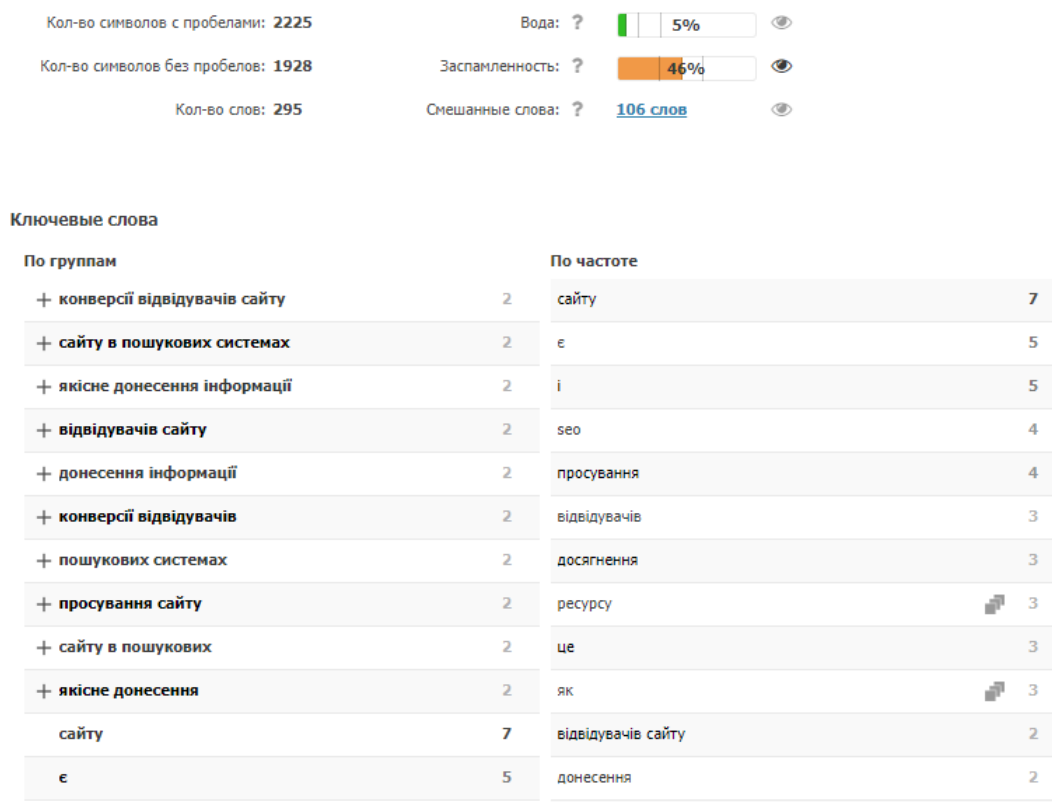


Рисунок 2.2 – Параметри SEO-аналізу тексту

Istio.com. Основна його перевага – семантичний аналіз. Цей сервіс одразу розроблявся для SEO. Можна вставляти потрібний текст у форму або просто скопіювати адресу сторінки на сайті, тоді її текст буде додано автоматично [28].

На рисунку 2.3 наведено приклад, як виглядають результати аналізу.

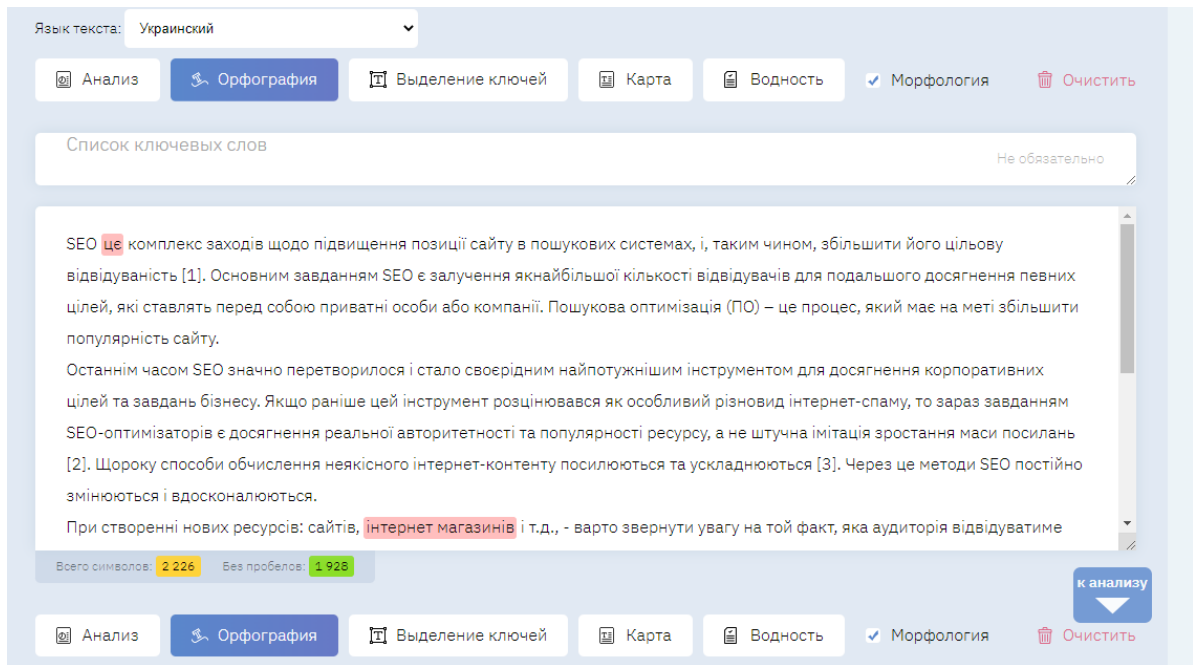


Рисунок 2.3 – Сервіс Istio.com

Перемикаючись між вкладками, можна активувати ті чи інші функції. Наприклад, вказати список ключових слів та Istio підсвітить усі їх входження на сторінці (рис. 2.4).

Показать 25 записей						Поиск:	
#	Слово	Кол-во	Релевантность	% в ядре	% в тексте		
1	сайт	9	3	4%	3.3%		
2	які	6	2	2.6%	2.2%		
3	seo	4	1.33	1.7%	1.5%		
4	пошукові	4	1.33	1.7%	1.5%		
5	ресурс	4	1.33	1.7%	1.5%		
6	просуванні	4	1.33	1.7%	1.5%		
7	дані	3	1	1.3%	1.1%		
8	завданні	3	1	1.3%	1.1%		
9	досягненні	3	1	1.3%	1.1%		
10	відвідувач	3	1	1.3%	1.1%		
11	створенні	3	1	1.3%	1.1%		
12	неякісні	3	1	1.3%	1.1%		
13	нові	3	1	1.3%	1.1%		
14	запит	2	0.66	0.8%	0.7%		
15	меті	2	0.66	0.8%	0.7%		
16	непопулярність	2	0.66	0.8%	0.7%		

Символов с пробелами	2226
Символов без пробелов	1928
Всего слов	265
Водность	15%
Классическая тошнота	3
Словарь	183 слов
Словарь ядра	163 слов

к тексту

Рисунок 2.4 – Перелік ключових слів

Важливий інструмент для оптимізації тексту – це «Карта», яка візуалізує слова, що найчастіше вживаються в тексті (рис. 2.5).

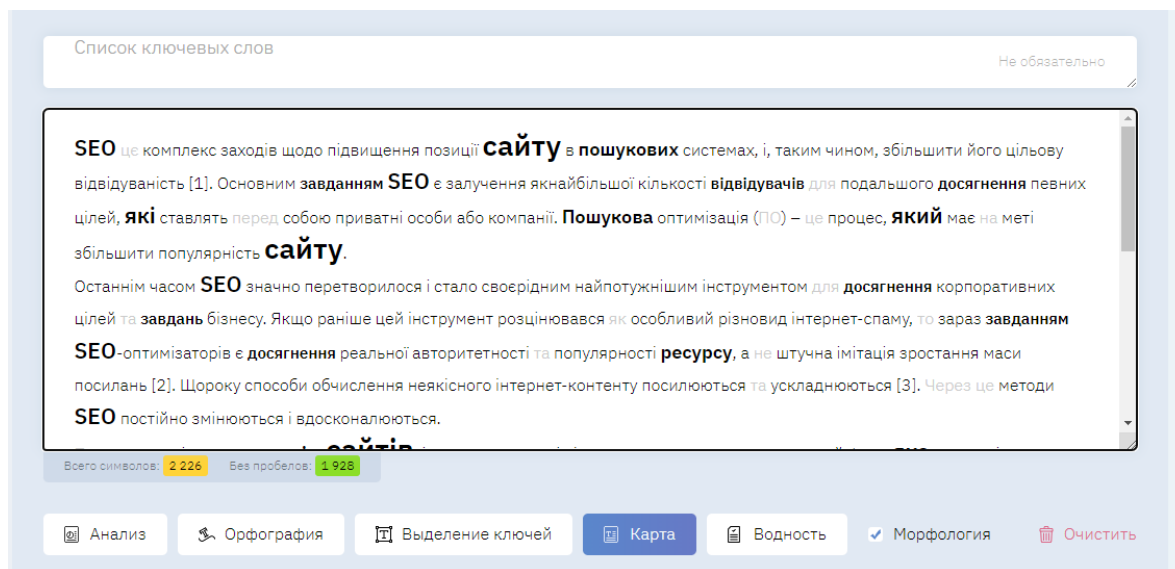


Рисунок 2.5 – Мапа тексту

SEOlub – цей сервіс пропонує багато різного функціоналу, протестуємо лише один – «Аналіз текстів» (рис. 2.6, рис. 2.7) [29]. З його допомогою можна отримати наступні критерії – дані про щільність ключових слів, обсяг та інші показники текстів на сторінках з топ-10 видачі ПС (рис. 2.7).

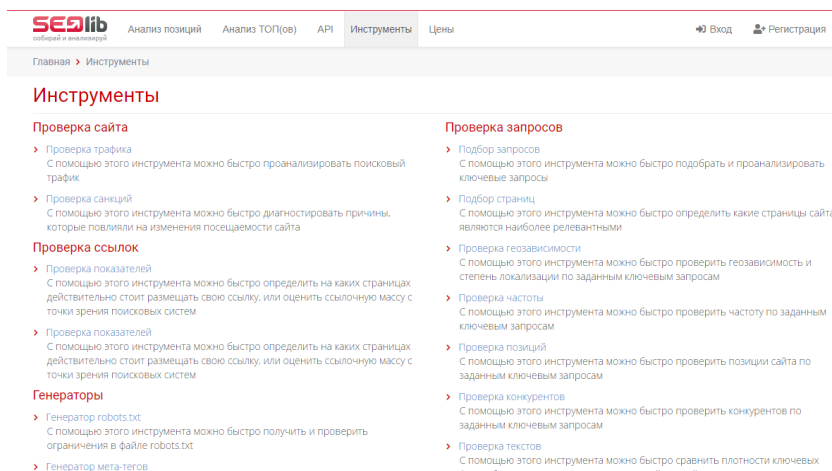


Рисунок 2.6 – Сервіс SEOlub

Анализ текстов

С помощью этого инструмента можно быстро сравнить плотности ключевых фраз, объема текстов и других показателей для сайтов

Форма проверки

Ключевые фразы:

SEO це комплекс заходів щодо підвищення позиції сайту в пошукових системах, і, таким чином, збільшити його цільову відвідуваність [1]. Основним завданням SEO є залучення якнайбільшої кількості відвідувачів для подальшого досягнення певних цілей, які ставлять перед собою приватні особи або компанії. Пошукова оптимізація (ПО) – це процес, який

Добавлено запросов : 4

Загрузить ключевые слова из файла в формате .txt или csv

Выбрать файл

Файл не выбран

Проанализировать страницы:

Из топ-10

Из топ-20

Указать вручную

Поисковые системы:

☐ Я Яндекс

☐ G Google

Что показать:

☒ Объем текста в словах и символах

☒ Точные вхождения ключевых фраз ?

☒ Вхождения ключевых фраз с учетом словоформ ?

☒ Тошнота ?

☒ Объем содержательной части страницы ?

Фраза в html-атрибутах:

☐ Title

☐ <H1>

☐ Description

☐ <H2>

☐ Keywords

☐ <H3>

☐ <alt>

☐ <a>

☐ Учитывать словоформы при анализе html-атрибутов

Какие блоки учитывать при анализе: ?

☒ Только содержательную часть ?

Рисунок 2.7 – Параметри перевірки тексту

В даному сервісі можна порівняти дані по топу вебсайтів з характеристиками тексту на своєму сайті, проводити аналіз по Яндекс або Google, а також виставляти додаткові налаштування для відображення іншої корисної інформації.

Однією з найважливіших характеристик тексту є показник унікальності. Система антиплагиату Content Watch дозволяє визначити показник унікальності конкретного фрагмента тексту, окремої веб-сторінки або запустити пакетну перевірку URL за списком [30].

Якщо фрагменти тексту є і на інших сайтах, сервіс показує адресу сторінки та відсоток збігу (рис. 2.8, рис. 2.9). У безкоштовному режимі доступно до 7 перевірок на день із обмеженням на 10 000 знаків без пробілів.

Проверить текст на уникальность

Длина текста: 2226 (без пробелов: 1928)

SEO це комплекс заходів щодо підвищення позиції сайту в пошукових системах, і, таким чином, збільшити його цільову відвідуваність [1]. Основним завданням SEO залучення якнайбільшої кількості відвідувачів для подальшого досягнення певних цілей, які ставлять перед собою приватні особи або компанії. Пошукова оптимізація (ПО) це процес, який має на меті збільшити популярність сайту. Останнім часом SEO значно перетворилося і стало своєрідним найпотужнішим інструментом для досягнення корпоративних цілей та завдань бізнесу. Якщо раніше цей інструмент розцінювався як особливий різновид інтернет-спаму, то зараз завданням SEO-оптимізаторів досягнення реальної авторитетності та популярності ресурсу, а не штучна імітація зростання маси посилань [2]. Щороку способи обчислення неякісного інтернет-контенту посилюються та ускладнюються [3]. Через це методи SEO постійно змінюються і вдосконалюються. При створенні нових ресурсів: сайтів, інтернет магазинів і т.д., варто звернути увагу на той факт, яка аудиторія відвідуватиме даний ресурс, а також звідки приходитимуть нові користувачі. Створення нового ресурсу тільки для людей призводитиме до поганої індексації сайту та повільного просування в топ. В даному випадку потрібно знайти золоту середину, за якої інтернет-ресурс буде цікавий людям, а також роботам, які займаються індексацією сайтів (пошуковим роботам ПР). Для просування існують різні методи seo-оптимізації, зокрема seo-ко-пірайтинг (від англ. search engine optimization copywriting), що «технічно створення та редагування текстів для вебсайтів» [4]. Тому метою даної технології, як зазначають дослідники, не тільки просування сайту в пошукових системах (ПС) за запитом, а й зростання конверсії відвідувачів сайту, і, перш за все, якісне донесення інформації покупцям. У зв'язку з цим просування сайту в ПС здійснюється за запитом, основою яких ключові слова. Правильно підібрані ключі, оптимальна їх щільність у тексті, правильно розставлені теги запорука успіху. За рахунок цього відбувається приріст конверсії відвідувачів сайту в реальних покупців. Це досягається тим, що має бути якісне донесення інформації та легкість сприйняття за допомогою грамотного використання мовних засобів [5].

Унікальність текста: **100.0%**

[Править этот текст](#)

[Новая проверка](#)

Рисунок 2.8 – Сервіс Content Watch

Проверить текст на уникальность

Длина текста: 610 (без пробелов: 524)

На дворе 2021 год, а заказчики текстов оптимизаторы, веб-мастеры, предприниматели до сих пор:

требуют полную техническую уникальность лучше и по text.ru, и по Advego, и ещё по парочке других сервисов одновременно; заворачивают или отправляют на доработку тексты только из-за пары недостающих процентов зашпаленности, водности или ещё чего-то такого; считают точные и неточные вхождения ключевых слов и объём текста, ориентируясь на топ-10 выдачи Яндекса или Google по целевому запросу; плюс заставляют подгонять материалы под «Главред», «Тургенев» и, не дай бог, какой-нибудь анализатор по закону Ципфа.

Уникальность текста: **<10%**

[показать все совпадения](#)

Адрес страницы	Сколько совпало	Совпадения
https://vk.com/@wdesiguix-vse-cto-nuzhno-znat-o-seo-analiz-tekstov-v-2019-godu	84.3%	показать
https://mnogotolka.ru/info/seo-analiz-teksta-sajta/	84.3%	показать
https://prokompter.ru/analiz-teksta-na-unikalnost-seo/	84.3%	показать
https://4apple.org/cto-takoe-seo-analiz-teksta/	84.3%	показать
https://vk.com/@rubix.marketing-seo-analiz-tekstov-dlya-chego-nuzhen-i-kak-...	66.3%	показать

[Нужен уникальный контент?](#)

[Править этот текст](#)

[Новая проверка](#)

Рисунок 2.9 – Приклад перевірки російського тексту

За результатами дослідження онлайн сервісів перевірки тексту за параметрами SEO-оптимізації розробили таблицю 2.2.

Таблиця 2.2 – Результати дослідження функціональності сервісів

Критерії оцінки	Text.ru	Istio.com	SEOLib	ContentWatch
Пошук в мережі інтернет	+	+	+	+
Пошук у базах даних	+	-	+	-
Параметри оцінки	унікальність, нудота, заспамленість, обсяг тексту, заміна букв, орфографія, ключові слова	унікальність, нудота, заспамленість, обсяг тексту, орфографія, ключові слова, карта	унікальність, нудота, заспамленість, обсяг тексту, ключові слова	унікальність, ключові слова
Наявність звіту	+	+	+	-
Правила безкоштовного користування	Безкоштовно, але треба чекати в черзі на перевірку	Безкоштовно до 5000 знаків	-	У безкоштовному режимі доступно до 7 перевірок на день, і з обмеженням на 10 000 знаків
Завантаження файлу / Посилання на сайт	+/-	+/+	+/+	-/+
Вибір мови тексту	-	+	-	-

Слід зазначити, що перевірені інструменти лише виявляють унікальність текстів, знаходять джерела запозичення та вираховують параметри тексту за формулами. Іноді цитування оцінюється як дублювання, і тому в підсумкових зведених звітах користувача інформують лише про фрагменти, які потенційно можуть бути несанкціонованими. Остаточне рішення про результати аналізу залишається на розсуд користувача або seo-оптимізатора, тому потреба в ручній перевірці та оцінці (перегляді) матеріалу людиною все ще існує.

Оскільки кожен такий ресурс має свої переваги та недоліки щодо програмно-апаратних характеристик, не можна очікувати 100% точних

результатів від обробки. Основна мета використання цього класу програм – прискорення роботи над seo-оптимізацією вебсайту та виведення його на перші сторінки ПС.

2.3 Висновки до розділу

В розділі розглянуті показники внутрішньої оптимізації вебсайту, спрямовані на покращення сприйняття ресурсу ПС та користувачами. Актуальним покращенням SEO вебсайту є оптимізація контенту. Контент має вирішувати проблеми користувачів, з якими вони прийшли на вебсторінку з пошуку. Він має бути корисним, унікальним та оптимізованим під пошукові потреби. Розглянуті показники SEO-оптимізації тексту та методи їх обчислення. Досліджено основні сервіси онлайн перевірки тексту та зроблено висновки.

3 РОЗРОБКА АРХІТЕКТУРИ СЕРВІСУ

3.1 Клієнт-серверна архітектура

Архітектура «клієнт-сервер» передбачає поділ процесів надання послуг та надсилання запитів на них на різних комп'ютерах у мережі, кожен з яких виконує свої завдання незалежно від інших [31].

В архітектурі «клієнт-сервер» кілька комп'ютерів-клієнтів надсилають запити та отримують послуги від централізованої службової машини – сервера, яка також може називатися хост-системою. Клієнтська машина надає користувачеві «дружній інтерфейс».

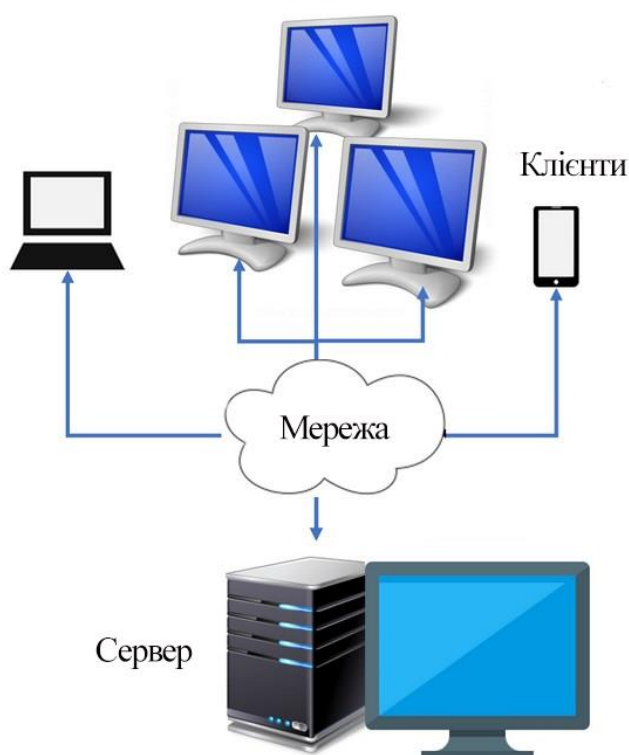


Рисунок 3.1 – Архітектура «Клієнт-Сервер»

Архітектуру «клієнт-сервер» прийнято розділяти на три класи: одно-, дво- та трирівневу.

Однорівнева архітектура «клієнт-сервер» (1-Tier) - така, де всі прикладні програми розосереджені по робочих станціях, які звертаються до спільного сервера баз даних або до спільного файлового сервера. Жодних прикладних програм сервер при цьому не виконує, лише надає дані. Загалом, така архітектура дуже надійна, проте, їй складно керувати, оскільки в кожній робочій станції дані будуть присутні у різних варіантах. Тому виникає проблема їхньої синхронізації на окремих машинах [31].

До дворівневої архітектури клієнт-сервер слід відносити таку, в якій прикладні програми зосереджені на сервері додатків, а в робочих станціях знаходяться програми-клієнти, які надають для користувачів інтерфейс для роботи з додатками на загальному сервері. Така архітектура є найбільш логічною для архітектури «клієнт-сервер» [31].

Переваги дворівневої архітектури:

- легко конфігурувати та модифікувати програми;
- користувачеві зазвичай легко працювати в такому середовищі;
- хороша продуктивність та масштабованість.

Проте, дворівнева архітектура має і обмеження:

- продуктивність може зменшуватися при збільшенні кількості користувачів;
- потенційні проблеми з безпекою, оскільки всі дані та програми знаходяться на центральному сервері;
- усі клієнти залежать від бази даних одного виробника.

У трирівневій архітектурі сервер баз даних, файловий сервер та інші є окремим рівнем, результати роботи якого використовує сервер додатків. Логіка даних та бізнес-логіка знаходяться на сервері додатків. Всі звернення клієнтів до бази даних відбуваються через проміжне програмне забезпечення, яке знаходиться на сервері програм. Внаслідок цього, підвищується гнучкість роботи та продуктивність [31].

Переваги трирівневої архітектури:

- цілісність даних;
- вища безпека порівняно з дворівневою архітектурою;
- захищеність бази даних від несанкціонованого проникнення.

Обмеження:

- більш складна структура комунікацій між клієнтами та сервером, оскільки в ньому також знаходиться middleware.

В окремий клас архітектури «клієнт-сервер» можна винести багаторівневу архітектуру, де кілька серверів додатків використовують результати роботи один одного, а також дані від різних серверів баз даних, файлових серверів та інших видів серверів. По суті, попередній варіант, трирівнева архітектура – не більше ніж окремий випадок багаторівневої архітектури. Перевагою багаторівневої архітектури є гнучкість надання послуг, які можуть бути комбінацією роботи різних додатків серверів різних рівнів та елементів цих додатків. Очевидним недоліком є складність, багатокомпонентність такої архітектури.

Характеристики архітектури «клієнт-сервер» [32]:

- асиметричність протоколів. Між клієнтами та сервером існують стосунки «один до багатьох». Ініціатором діалогу із сервером зазвичай є клієнт;
- інкапсуляція послуг. Після отримання запиту на послугу від клієнта сервер вирішує, як повинна бути виконана дана послуга. Модифікація сервера може проводитися без впливу на роботу клієнтів, оскільки це не впливає на опублікований інтерфейс взаємодії між ними. Іншими словами, максимум, що може при цьому відчувати користувач, – незначна затримка відгуку сервера протягом невеликого часу апгрейду;
- цілісність. Програми та загальні дані для сервера управляються централізовано, що знижує вартість обслуговування та захищає цілісність

даних. У той же час дані клієнтів залишаються персоніфікованими і незалежними;

– місцева прозорість. Сервер - це програмний процес, який може виконуватися на тій же машині, що й клієнт або на іншій машині, підключеній по мережі. Програмне забезпечення клієнт-сервер зазвичай приховує розташування сервера від клієнтів, перенаправляючи запит на послуги через мережу;

– обмін на основі повідомлень. Клієнти та сервер є слабо зв'язними процесами, які обмінюються повідомленнями: запитами на послуги та відповідями на них;

– модульний дизайн, здатний до розширення. Модульний дизайн програмної платформи «клієнт-сервер» надає їй стійкості до відмов, тобто відмова в якомусь модулі не викликає відмови всієї програми. У такій системі, один або більше серверів можуть відмовити без зупинки всієї системи в цілому, поки послуги сервера, що відмовив, можуть бути надані з резервного сервера. Інша перевага модульності в тому, що програма «клієнт-сервер» може автоматично реагувати на підвищення або зниження навантаження на систему шляхом додавання або відключення послуг або серверів;

– незалежність від платформи. Ідеальна програма «клієнт-сервер» не залежить від платформ обладнання або операційної системи. Клієнти та сервери можуть розгортатися на різних апаратних платформах та різних операційних системах;

– масштабованість. Системи «клієнт-сервер» можуть масштабуватися як горизонтально (за кількістю серверів та клієнтів), так і вертикально (за продуктивністю та спектром послуг);

– розділення функціоналу. Система «клієнт-сервер» - це співвідношення між процесами, що працюють на одній або різних машинах. Сервер – це процес надання послуг. Клієнт – це споживач послуг;

– загальне використання ресурсів. Один сервер може надавати послуги безлічі клієнтів одночасно, і регулювати їх доступ до спільно використовуваних ресурсів.

До переваг архітектури клієнт-сервер можна віднести:

– централізованість, оскільки всі дані та управління зосереджені на центральному сервері;

– інформаційна безпека, оскільки ресурси загального користування адмініструються централізовано;

– продуктивність використання виділеного сервера підвищує швидкість роботи ресурсів загального користування;

– масштабованість, кількість клієнтів та серверів можна збільшувати незалежно один від одного.

До недоліків архітектури клієнт-сервер слід віднести:

– перевантаження трафіку в мережі, що є головною проблемою в мережах клієнт-сервер. Коли велика кількість клієнтів одночасно запитують одну послугу на сервері, кількість запитів може створити перевантаження в мережі;

– наявність єдиної точки відмови у невеликих мережах з одним сервером. Якщо він відмовляє, то всі клієнти залишаються без обслуговування;

– перевищення меж ресурсів сервера, коли нові клієнти, які вимагають послугу, залишаються без обслуговування. У таких випадках потрібно термінове розширення ресурсів сервера;

– іноді клієнтські програми можуть не працювати на терміналах користувачів, якщо не встановлені відповідні драйвери. Наприклад, користувач надсилає запит на друк документа, а на сервері немає відповідного драйвера для друку цього формату документа на певному принтері.

Для розробки програмного забезпечення SEO-аналізу тексту обрана

клієнт-серверна архітектура. Клієнту буде запропонований дружній інтерфейс для завантаження текстів та перегляду результатів аналізу. Всі обчислення та аналіз будуть виконуватись на сервері. На рисунку 3.2 наведена початкова структура програми.

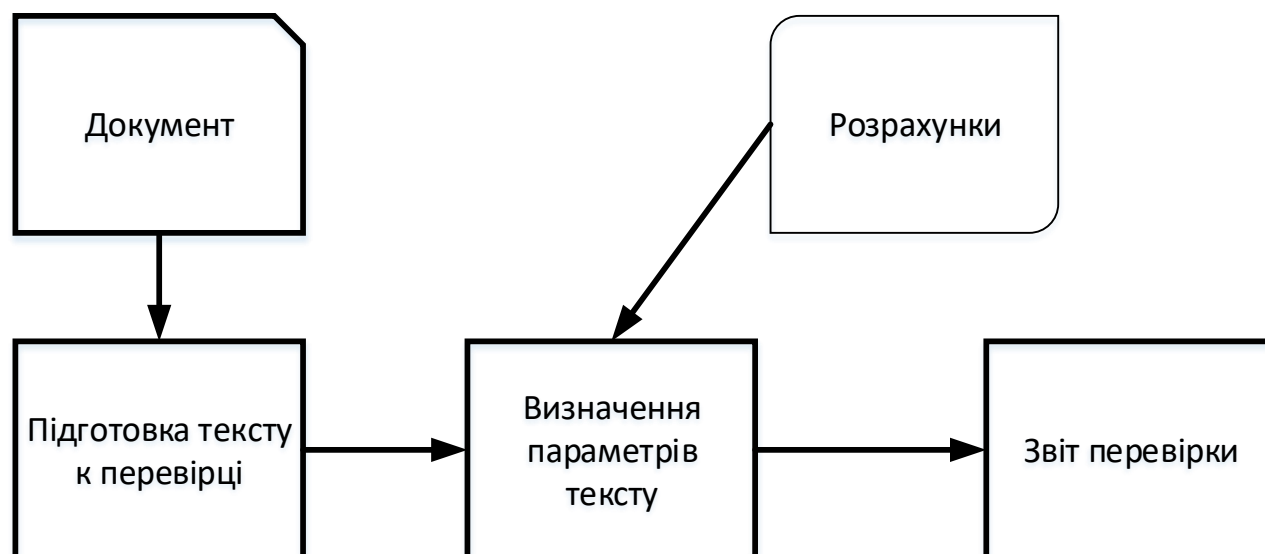


Рисунок 3.2 - Початкова структура програми

Вхідний текст піддається попередньої обробки, в яку входять:

- видалення із тексту розділових знаків і спецсимволів;
- перетворення регістру;
- видалення стоп-слів і розділових знаків (прийменники, прислівники і т.д.);
- стемінг (обробка закінчень слів).

Стемінг полягає у обробці закінчення слів. У нашому випадку вони просто відкидаються. Це дозволяє виключити вплив таких модифікацій тексту, як зміна однини і множини, чоловічого і жіночого роду, теперішнього часу і т.д.

Ми не ставили завдання перевіряти текст на плагіат. Перш за все нам цікаві параметри тексту для SEO-оптимізації (водність, заспамленість, орфографія) (рис. 3.3).

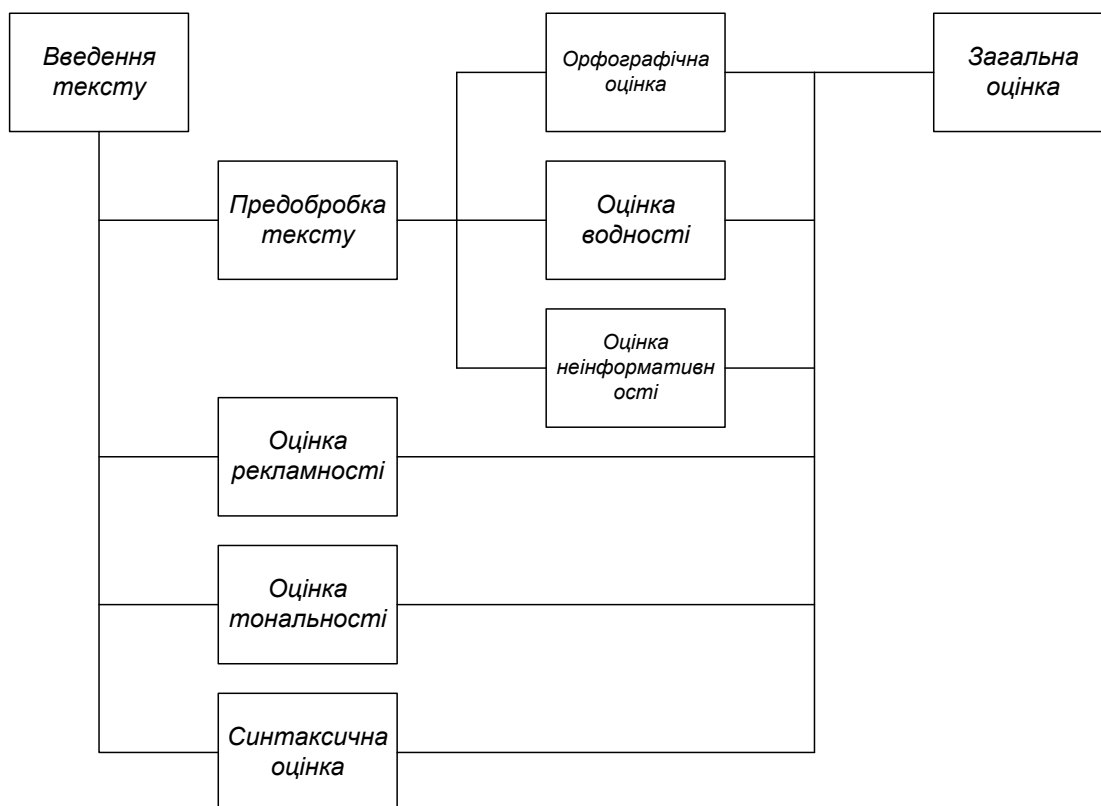


Рисунок 3.3 – Структура програми

Використання розробляемого додатку наведемо на UML-діаграмах. Діаграма класів «Клієнт» показую всі можливі варіанти використання системи (рис. 3.4).

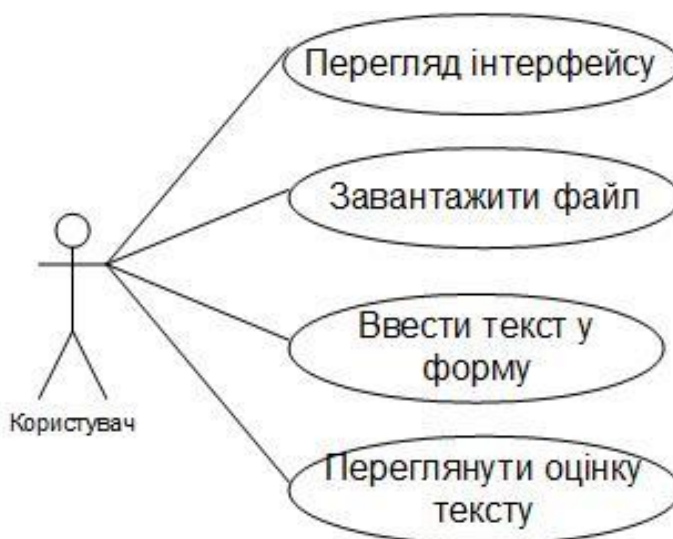


Рисунок 3.4 – Діаграма класів «Клієнт»

За допомогою діаграми сценарій показані всі стани системи в процесі аналізу тексту (рис. 3.5).

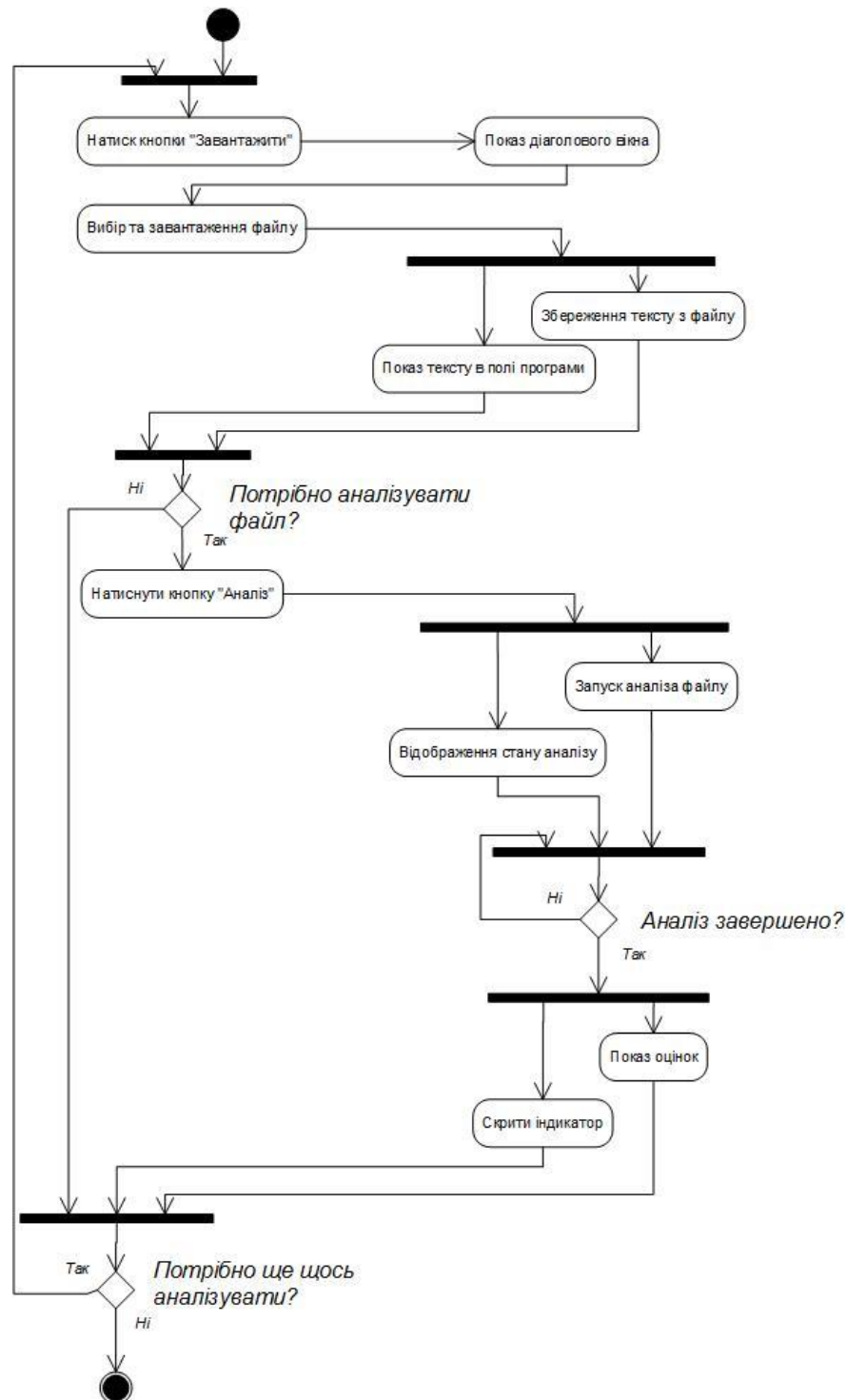


Рисунок 3.5 – Діаграма поведінки системи

3.2 MVC-паттерн

MVC – це шаблон програмування, який дозволяє розділити логіку програми на три частини [33]:

- model (модель). Отримує дані від контролера, виконує необхідні операції та передає їх у вигляд;
- view (вигляд або уявлення). Отримує дані від моделі та виводить їх для користувача;
- controller (контролер). Обробляє дії користувача, перевіряє отримані дані та передає їх моделі.

Якщо ж говорити про програми, то компоненти будуть наступні:

- вигляд – інтерфейс;
- контролер - обробник подій, що ініціюються користувачем (натискання на кнопку, перехід за посиланням, відправлення форми);
- модель — метод, який запускається обробником та виконує всі основні операції (отримання записів із бази даних, проведення обчислень).

Варто також відзначити, що реалізація патерну MVC може відрізнитися залежно від завдання. Наприклад, у веб-розробці модель і вид взаємодіють один з одним через контролер, а в додатках модель може сама повідомляти, що потрібно щось змінити.

Цей патерн розробки потрібен у тому, щоб розділити логічні частини програми та створювати їх окремо друг від друга. Тобто писати незалежні блоки коду, які можна як завгодно міняти, не торкаючись інших. Наприклад, щоб можна було переписати спосіб обробки даних, не змінюючи при цьому спосіб їхнього відображення. Це дозволяє ефективно працювати кільком програмістам – кожен займається своїм компонентом. При цьому розробнику не потрібно вникати в чужий код і його дії не вплинуть на інші фрагменти програми.

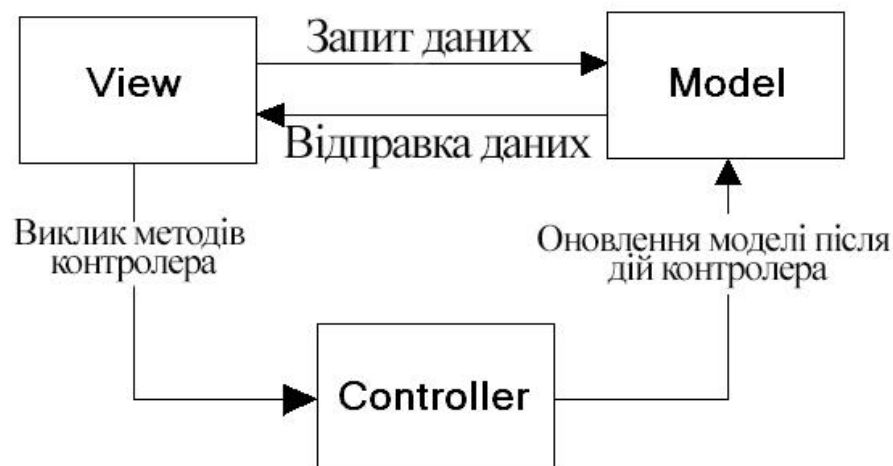


Рисунок 3.6 - Типова послідовність роботи MVC-програми

Типову послідовність роботи MVC-програми можна описати так (рис. 3.6):

- при заході користувача на веб-ресурс, скрипт ініціалізації створює екземпляр програми та запускає його на виконання. При цьому відображається вигляд головного вікна програми;
- програма отримує запит від користувача та визначає запитані контролер та дію;
- додаток створює екземпляр контролера та запускає метод дії, в якому містяться виклики моделі, які роблять розрахунки;
- після цього дія формує подання з даними, отриманими з моделі і виводить результат користувачеві.

3.3 Формальне визначення критеріїв SEO-аналізу

У розділі 2 було визначено критерії, якими оцінюватимуться тексти. У цьому пункті будуть описані формальні способи обчислення оцінок за вибраними критеріями.

В рамках даної кваліфікаційної роботи, водність тексту буде

розраховуватися за формулою (3.1):

$$V = \frac{\sum w_{stop}}{\sum w_t}, \quad (3.1)$$

де V – водність тексту;

W_{stop} – кількість стоп-слів;

W_t – загальний обсяг тексту.

Значення критерію може бути від «0» до «1». Значення «0» відповідає відсутності стоп-слів, «1» - що текст повністю наповнений стоп-словами. Враховуючи це, формулу для розрахунку оцінки можна переписати формулою (3.2):

$$V = \begin{cases} 2 * \frac{\sum w_{stop}}{\sum w_t}, & \text{if } \frac{\sum w_{stop}}{\sum w_t} < 0,5 \\ 1, & \text{if } \frac{\sum w_{stop}}{\sum w_t} \geq 0,5 \end{cases}, \quad (3.2)$$

Таким чином, якщо кількість стоп-слів більше половини від загальної кількості слів вираховується максимальна оцінка. А якщо кількість стоп-слів менше половини, значення буде подвійне відношення стоп-слів до загальної суми кількості слів.

Оскільки завданням роботи є аналіз якості текстів, а результатом формули (3.2) є водність тексту, то для отримання оцінки необхідно перетворити формулу у такий вигляд (3.3):

$$V = 1 - \begin{cases} 2 * \frac{\sum w_{stop}}{\sum w_t}, \text{if } \frac{\sum w_{stop}}{\sum w_t} < 0,5 \\ 1, \text{if } \frac{\sum w_{stop}}{\sum w_t} \geq 0,5 \end{cases}, \quad (3.3)$$

Орфографія буде розрахована за формулою (3.4):

$$OE = \frac{\sum err_o}{\sum w_t}, \quad (3.4)$$

де OE – значення правильності написання;

err_o – кількість орфографічних помилок;

Wt – загальний обсяг тексту.

Значення критерію від «0» до «1», де «0» - це всі слова в тексті орфографічно правильні, при «1» всі слова неправильні. Розрахунок слід доповнити, як і у випадку з водністю тексту, формула (3.5):

$$OE = \begin{cases} 2 * \frac{\sum err_o}{\sum w_t}, \text{if } \frac{\sum err_o}{\sum w_t} < 0,5 \\ 1, \text{if } \frac{\sum err_o}{\sum w_t} \geq 0,5 \end{cases}, \quad (3.5)$$

Оскільки якість тексту протилежна кількості помилок у тексті, то для отримання оцінки правильного критерію необхідно перетворити формулу відповідно (3.6):

$$OE = 1 - \begin{cases} 2 * \frac{\sum err_o}{\sum w_t}, \text{if } \frac{\sum err_o}{\sum w_t} < 0,5 \\ 1, \text{if } \frac{\sum err_o}{\sum w_t} \geq 0,5 \end{cases}, \quad (3.6)$$

Синтаксична правильність тексту розрахована формулою (3.7):

$$SE = \frac{\sum err_s}{\sum w_t}, \quad (3.7)$$

де SE – значення синтаксичної правильності;

err_s – кількість синтаксичних помилок;

Wt – загальний обсяг тексту.

Подібно до попередніх критеріїв формулу (3.8) було доповнено:

$$SE = 1 - \begin{cases} 2 * \frac{\sum err_s}{\sum w_t}, \text{if } \frac{\sum err_s}{\sum w_t} < 0,5 \\ 1, \text{if } \frac{\sum err_s}{\sum w_t} \geq 0,5 \end{cases}, \quad (3.8)$$

Оцінка інформативності розраховується за такою формулою (3.9):

$$I = \frac{\sum w_{diff}}{\sum w_t}, \quad (3.9)$$

де I – значення інформативності;

w_{diff} – кількість слів, що відрізняються;

W_t – загальний обсяг слів у тексті.

Враховуючи, що значення «0» рідко зустрічається в текстах, формулу було доповнено (3.10):

$$I = \begin{cases} \frac{\sum w_{diff}}{\sum w_t}, \text{if } 0,3 < \frac{\sum w_{diff}}{\sum w_t} < 0,8 \\ 1, \text{if } \frac{\sum w_{diff}}{\sum w_t} \geq 0,8 \\ 0, \text{if } \frac{\sum w_{diff}}{\sum w_t} \leq 0,3 \end{cases}, \quad (3.10)$$

Точної формули, пов'язаної з частотним словником, для визначення рекламності тексту немає. Для розрахунку реклами вибирають машинне навчання. У рамках класифікації завдань необхідно підготувати тексти для навчання та тестування, а також визначити класи для виконання завдань. Можливі значення класифікатора: 0, 1 (табл. 3.1.).

Таблиця 3.1 – Опис рекламного індикатора

Вихідні значення	Опис
0	Не рекламний текст (<5%)
1	Рекламний текст, практично ніякої корисної інформації, крім опису товарів з кращих сторін (>80%)

Тональність тексту визначається за допомогою машинного навчання. У рамках класифікації завдань необхідно підготувати базу даних для навчання та

тестування. Можливі вихідні значення класифікатора: 0, 0.5, 1 (табл. 3.2).

Таблиця 3.2 – Опис критерію тональності

Вихідні значення	Опис	Приклад
0	негативна	Жахливий товар! Більше не буду купувати
0,5	нейтральна	Купив товар у подарунок
1	позитивна	Найкращий товар! Користуюсь їм щодня

Тональність та рекламність несуть парний характер оцінки тексту. По окремо вони не визначають якість тексту.

3.4 Висновки до розділу

SEO-аналіз тексту дозволяє оцінити контент з різних факторів. Його зазвичай проводять автоматично за допомогою онлайн-сервісів у рамках аудиту сайту. Ці програми прораховують показники запам'ятованості, водності, виводять частоту вживання тих чи інших слів. Автоматична перевірка допомагає швидко виявити слабкі місця будь-якої статті. Переваги SEO-аналізу тексту за допомогою клієнт-серверного додатку: не потрібно інсталиувати додаткове програмне забезпечення на комп'ютері, перевірка займає мінімум часу, можливість редагування тексту у формі, щоб налаштувати його під потрібні параметри.

4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

4.1 Вибір мови програмування

Для вирішення поставленої задачі було обрано мову програмування Python. Це скриптова мова програмування розроблена для запису «сценаріїв», послідовних операцій, які користувач може виконувати на комп'ютері [34].

Python – найпопулярніша мова програмування, згідно запитів від Stack Overflow (рис. 4.1).

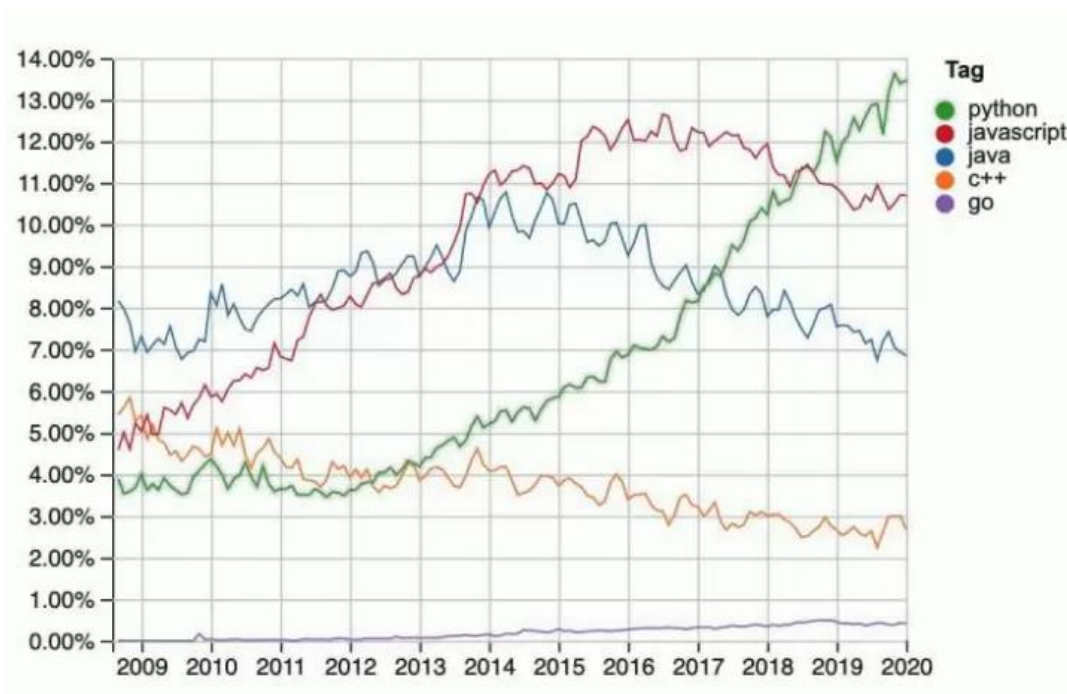


Рисунок 4.1 – Популярність мови Python

По індексу PYPL, який порівнює кількість запитів на посібники в Google, Python також лідує з великою перевагою (табл. 4.1).

Таблиця 4.1 – Кількість запитів на посібники в Google

Позиція	Мова програмування	Відсоток запитів	Тенденція
1	Python	31,02%	+2,2%
2	Java	16,38%	-2,38%
3	JavaScript	8,41%	+0,4%
4	C#	6.52 %	0.6 %
5	PHP	5,83	-0.4 %
6	C/C++	5.56 %	-0.4 %
7	R	4.26 %/	+0.4 %
8	Objective-C	3.48 %	+0,8%
9	Swift	2.37 %	-0.1 %
10	TypeScript	1,9%	+0,1%

Аналогічні результати в індексах від ТЮВЕ і Бюро статистики праці США. Питон займає 1–3 місце для всіх авторитетних рейтингів популярності мов [30].

Популярність мови грає істотну роль за наступними причинами:

- затребуваність – індикатор якості Python;
- велика культивість молодих спеціалістів гарантує розвиток розробок

ПЗ;

- багатомільйонне співтовариство допомагає постійно розвивати і покращувати мову;
- більше людей займається розробкою готових оптимізованих бібліотек і модулів для різних технічних і бізнес-задач.

Синтаксис Python доволі простий, він схожий на природну мову. Лаконічний, читабельний код легше підтримувати, перевіряти та виправляти. Порівняємо розмір міні-програми, що малює піраміду, на Java та Python (рис. 4.2).

JAVA	vs	PYTHON
<pre> public class JavaPyramid 1 { public static void main(String[] args) { for(int i=1; i<=5; i++){ for(int j=0; j<i; j++){ System.out.print(" * "); } //generate a new line System.out.println(" "); } } } </pre>		<pre> def create_pyramid (rows): for i in range (rows): print (' * ' * (i+1)) </pre>

Рисунок 4.2 – Порівняння програми на мові Python та Java

Також, на швидкість розробки впливає те, що мова Python – інтерпретована. Інтерпретатор читає вихідний текст програми по одному рядку за раз, виконує цей рядок і лише після цього переходить до наступного рядка.

Коли говорять про веб-розробку на Python, найчастіше передбачається розробка на Django [35]. Django це набір перевірених інструментів для створення всіх компонентів вебдодатку.

Розглянемо переваги використання фрейворку:

- швидкість розробки;
- універсальність. Django включає ключові інструменти веброзробки: міграція баз даних, автентифікація користувачів, форми, адміністративну панель. У багатьох інших фреймворках для роботи з цими функціями необхідно підключати окремі бібліотеки;
- гнучкість. Django — фреймворк, що легко настраюється;
- модулі та програми. До проектів Django можна підключати готові модулі;
- архітектура фреймворку слідує патерну проектування MVC. Він має на увазі розмежування бізнес-логіки та її візуалізації. Це дозволяє розробляти компоненти паралельно, змінювати їх окремо. А також по-різному надавати дані, залежно від контексту;
- масштабованість. Django підходить для створення високонавантажених сайтів із великим трафіком;
- безпека. Django за замовчуванням підтримує захист від таких видів атак, як використання SQL-коду та міжсайтових підробок запитів.

Недоліки:

- через величезну кількість вбудованих можливостей Django складніше освоїти, ніж інші фреймворки;
- для невеликих проектів із простими вимогами функціональність може бути надмірною.

Як бачимо, мінуси Django дуже умовні. Це стабільний та багатофункціональний фреймворк без яскраво виражених слабких сторін. Чим і пояснюється його вибір для розробки проекту.

4.2 Алгоритми обробки тексту

На етапі постановки завдання відомо, що на вході програми подається

текст. Вихідними параметрами є набір оцінок за рядом перерахованих вище критеріїв. Текст повинен бути проаналізований в обробленому вигляді, в якому всі слова наведені до вихідної форми, тому що в іншому випадку можуть виникнути ситуації, коли програма не прийматиме різні словоформи однієї і тієї ж лема [36].

У першій частині алгоритму дообробки відбувається розбір тексту на слова, приведені до початкової форми засобами бібліотеки *рурі*. Результатом роботи даного блоку є частотний словник тексту. Алгоритм роботи показаний на рисунку 4.3. Програмний опис реалізації алгоритму наведений на рисунку 4.4.

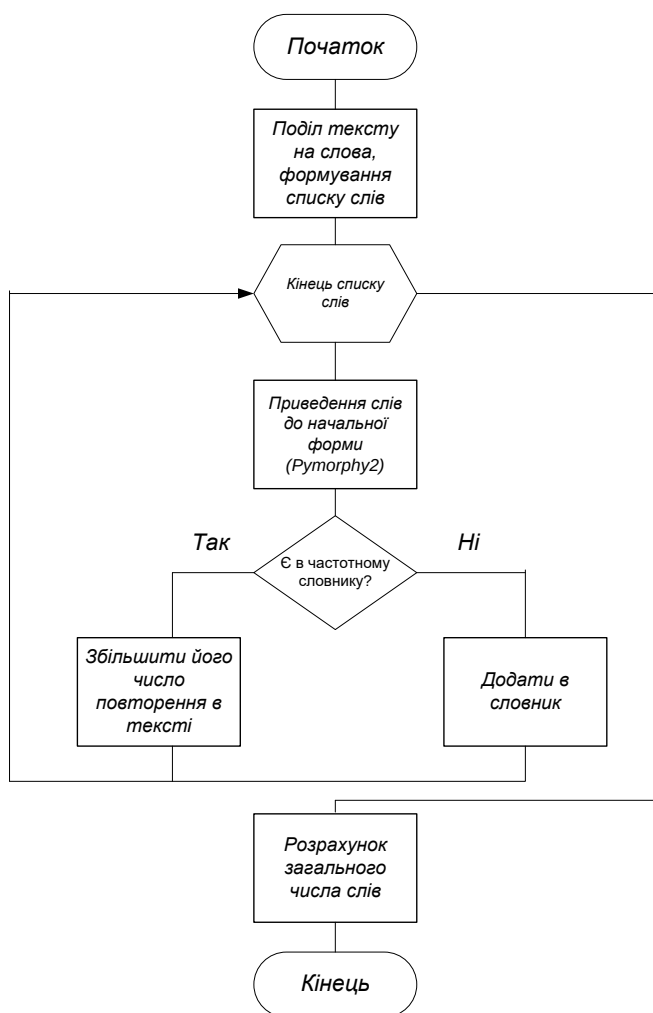


Рисунок 4.3 – Алгоритм дообробки тексту

```

1 def review_to_wordlist(review, remove_stopwords=False):
2     # Розбиваємо речення на список слів (list of words)
3     # 1. Видаляємо html теги та URL
4     review = re.sub(r'^https?:\/\/.*[\r\n]*', '', review, flags=re.MULTILINE)
5     review_text = BeautifulSoup(review, "html.parser").get_text()
6     # 2. Залишаємо тільки літери
7     review_text = re.sub("[^а-яА-Я]", " ", review_text)
8     # 3. Переводимо текст на нижній регістр
9     words = review_text.lower().split()
10    morph = pymorphy2.MorphAnalyzer()
11    words = [morph.parse(word)[0].normal_form + "_" +
12             OpenCorpora2Universal.get(morph.parse(word)[0].tag._POS, morph.parse(word)[0].tag._POS)
13            for word in words]
14    # 4. Видаляємо стоп-слова
15    if remove_stopwords:
16        # stops = set(stopwords.words("english"))
17        stops = set(stopwords.words("russian"))
18        stops = set(stopwords.words("ukraine"))
19    words = [w for w in words if not w in stops]
20    # 5. Повертаємо слова
21    return (words)
22    @staticmethod

```

Рисунок 4.4 – Програмна реалізація дооброби тексту

Синтаксичний аналіз тексту проходить за допомогою бібліотеки LanguageTool. На рисунку 4.5 показано алгоритм роботи блоку. Програмний код наведений на рисунку 4.6.

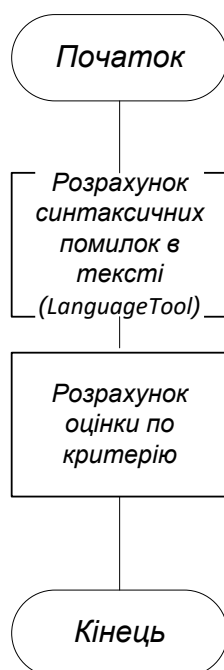


Рисунок 4.5 - Алгоритм оцінки синтаксису тексту

```

1 from language_check import LanguageTool
2 class GrammMark:
3     def __init__(self, obj):
4         self.__value = 0.0
5         self.__obj = obj
6         self.__txt = obj.get_txt()
7         self.__commas(obj)
8     def __commas(self, obj):
9         tool = LanguageTool("ru-RU")
10        matches = tool.check(self.__txt)
11        self.__value = len(matches) / obj.get_totalword()
12        if self.__value >= 0.5:
13            self.__value = 1.0
14        else:
15            self.__value *= 2 69
16        def get_mark(self):
17            return 1 - self.__value
18        def __str__(self):
19            return "%.2f" % self.get_mark()

```

Рисунок 4.6 – Програмна реалізація оцінки синтаксису тексту

Орфографічний аналіз тексту показує наявність орфографічних помилок за допомогою бібліотеки PyEnchant (рис. 4.7, рис. 4.8).

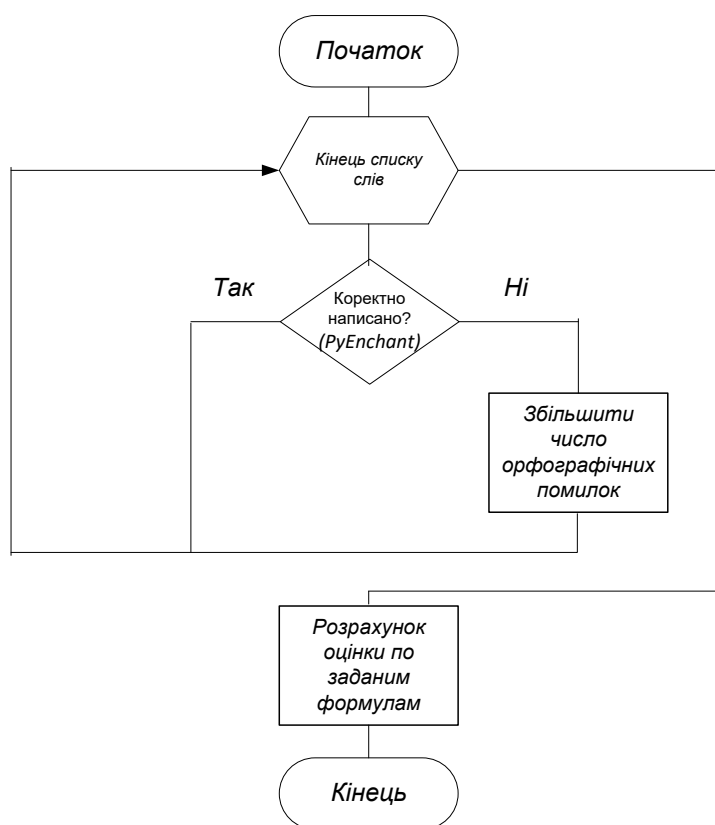


Рисунок 4.7 – Алгоритм оцінки орфографії тексту

```

1  import enchant
2  import pymorphy2
3  class Orthography:
4  def __init__(self, obj):
5  self.__orf_err = 0
6  self.__value = 0.0
7  self.__obj = obj
8  self.__vocab = obj.get_words()
9  self.__wateri(obj)
10 def __wateri(self, obj):
11 morph = pymorphy2.MorphAnalyzer()
12 c = enchant.Dict("ru_RU")
13 for key in self.__vocab:
14 tmp1 = morph.parse(key)[0]
15 if c.check(tmp1.normal_form) == False:
16 self.__orf_err += 1
17 self.__value = self.__orf_err / obj.get_totalword()
18 if self.__value >= 0.5:
19 self.__value = 1.0
20 else:
21 self.__value *= 2
22 def get_mark(self):
23 return 1 - self.__value
24 def __str__(self):
25 return "%.2f" % self.get_mark()

```

Рисунок 4.8 – Програмна реалізація оцінки орфографії тексту

Обчислення показника водності наведений на алгоритмі 4.9. Програмний код наведений на рисунку 4.10.

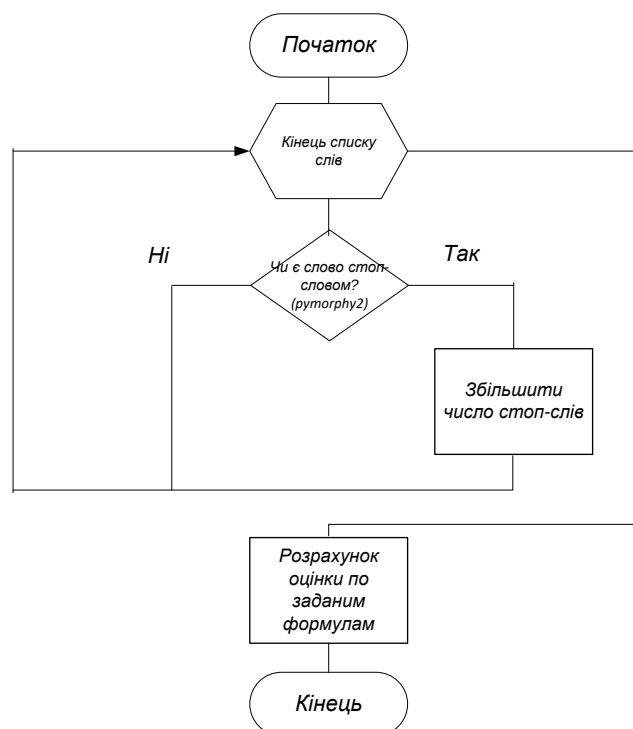


Рисунок 4.9 – Алгоритм обчислення водності тексту

```

1 import pyphi
2 class Water:
3     def __init__(self, obj):
4         self.__value = 0.0
5         self.__stop_word = 0
6         self.__obj = obj
7         self.__vocab = obj.get_words()
8         self.__wateri(obj)
9     def __wateri(self, obj):
10        morph = pyphi.MorphAnalyzer()
11        for key in self.__vocab:
12            tmp1 = morph.parse(key)[0]
13            if tmp1.tag.POS == "INITJ" or tmp1.tag.POS == "PRCL" or tmp1.tag.POS == "CONJ" or tmp1.tag.POS == "FRED":
14                self.__stop_word += 1
15        self.__value = self.__stop_word / obj.get_totalword()
16        if self.__value >= 0.5:
17            self.__value = 1.0
18        else:
19            self.__value *= 2
20    def get_mark(self):
21        return 1 - self.__value
22    def __str__(self):
23        return "%.2f" % self.get_mark()

```

Рисунок 4.10 – Програмна реалізація розрахунку критерію водність

4.3 Машинне навчання для оцінки рекламності та тональності тексту

Для реалізації моделі Word2Vec була обрана бібліотека gensim. Ця реалізація є переписаним на Python алгоритмом, розробленим Google, реалізованим на мові програмування C [37]. Бібліотека gensim дозволяє зберігати навчені моделі Word2Vec в окремому файлі. Результатом роботи якої є файл з навченою моделлю. Для реалізації класифікатора була обрана бібліотека scikit-learn [32].

Алгоритм навчання моделей Word2Vec і Random forest винесено в окремий блок.

Модель Word2Vec приймає список слів в початковій формі (лемі), який вона вивчає. Кожне слово замінюється відповідним вектором з Word2Vec уявлення, потім вектора, отримані в рамках одного тексту, підсумовуються і результат ділиться на кількість слів. Таким чином, навчальною базою для класифікатора є список векторів (рис. 4.11).

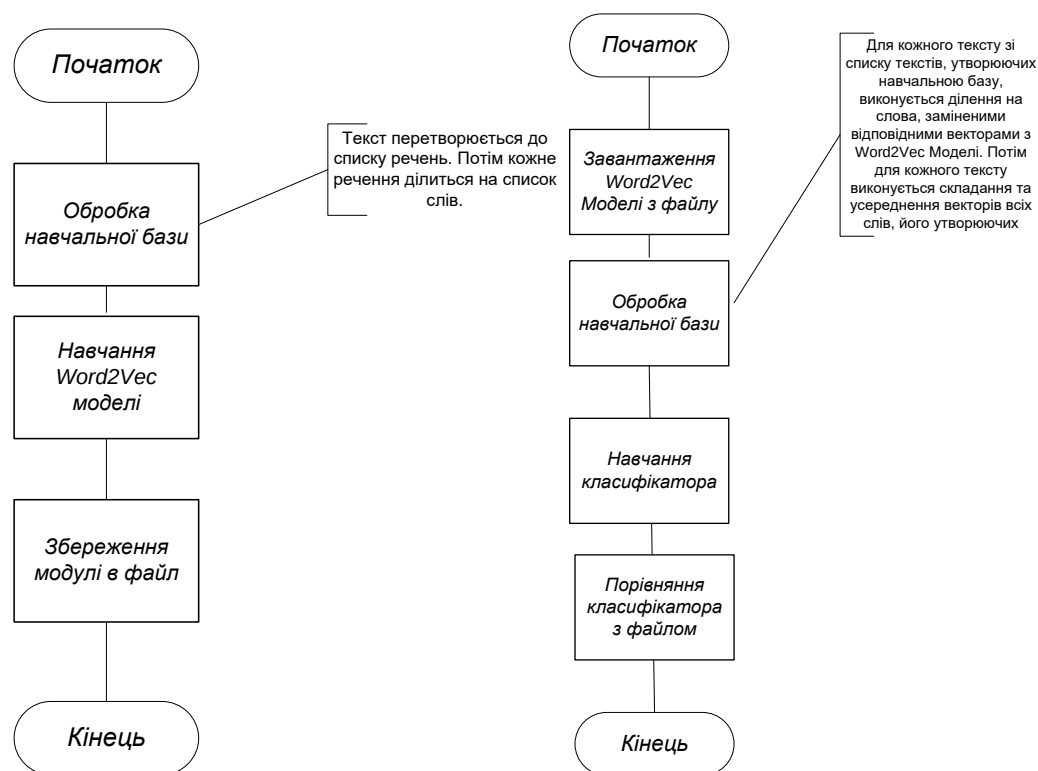


Рисунок 4.11 – Алгоритм навчання

Для обчислення показника рекламності на вхід подається текст, який перетворюється до певного формату, після чого оцінюється класифікатором. Алгоритм оцінки тональності тексту повторює алгоритм обчислення рекламності, але в ньому використовуються інші моделі Word2Vec і Random forest.

Для розробленого додатка був навчений набір класифікаторів на різних Word2Vec моделях. При тестуванні використовувалися моделі Word2Vec на основі скачаного корпусу текстів.

Корпус текстів є електронним набором текстових даних (dataset) за різними тематиками або за конкретною, що відображає організацію лексичної системи конкретної мови за цими тематиками. На жаль, в україністиці корпусів текстів, створених для навчання текстових класифікаторів, пошукових машин, систем текстового аналізу та вирішення інших лінгвістичних завдань, практично немає. Підходячи до питання тематичної класифікації, що відповідає

задачі інформаційного пошуку, ми вважаємо за можливе виділити чотири проекти:

- Браунівський корпус української мови [32];
- великий електронний словник української мови [33];
- російсько-український словник [34];
- LanguageTool API NLP UK [35].

Браунівський корпус української мови (БрУК) знаходиться у вільному доступі. Це дає можливість безперешкодної роботи із накопиченою базою. Корпус налічує 730 документів - файлів формату .txt з текстом і описом інформації про нього. Також кожен файл починається з певної літери, що означає клас, до якого належить текст. Кількість класів БрУК фіксована і дорівнює дев'яти. Для кожної з наведених вище моделей були навчені класифікатори для оцінки показників тональності та рекламності тексту.

4.4 Висновки до розділу

У цьому розділі визначені технології розробки вебдодатку: Python, PyPi, gensim (Word2Vec), scikit-learn (машинне навчання), Django (фреймворк). Були детально описані алгоритми обчислення SEO-показників тексту за такими критеріями як водність, інформативність, граматична і орфографічна коректність, рекламність та тональність. Розроблені алгоритми навчання моделі Word2Vec і класифікатора Random forest.

5 ТЕСТУВАННЯ SEO-АНАЛІЗУ

5.1 Інтерфейс сервісу

Інтерфейс був розроблений із використанням фреймворка Django (рис.5.1). Вебдодаток виконує всі вимоги діаграми використання розглянутою у 3 розділі:

- головна сторінка;
- можливість завантажити файл;
- введення тексту у вікно програми;
- перегляд параметрів оцінки тексту.

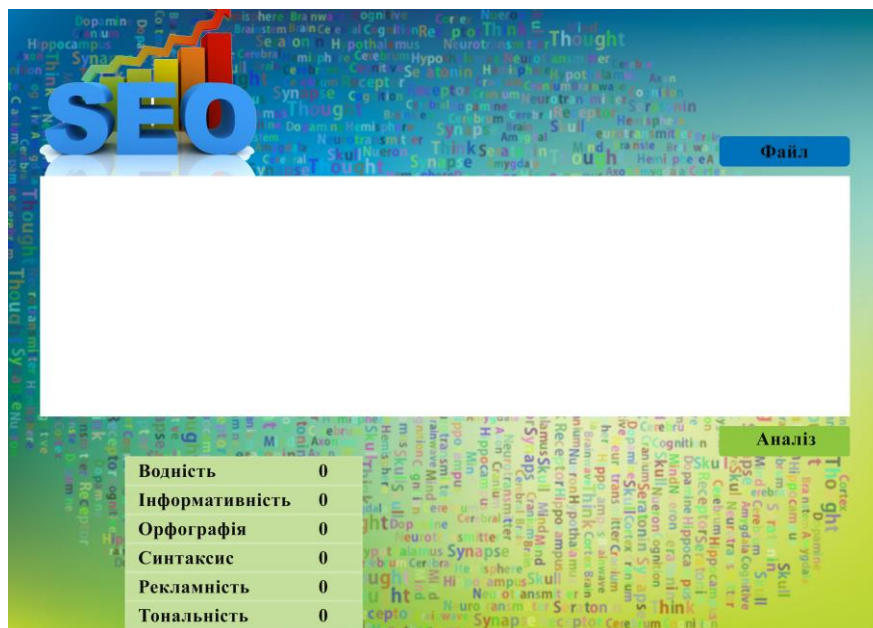


Рисунок 5.1 – Інтерфейс застосунку

Для завантаження доступні лише файли *.txt. Після завантаження обраний текст відображається в полі нижче. Щоб почати процес аналізу тексту, необхідно натиснути відповідну кнопку (рис. 5.2).

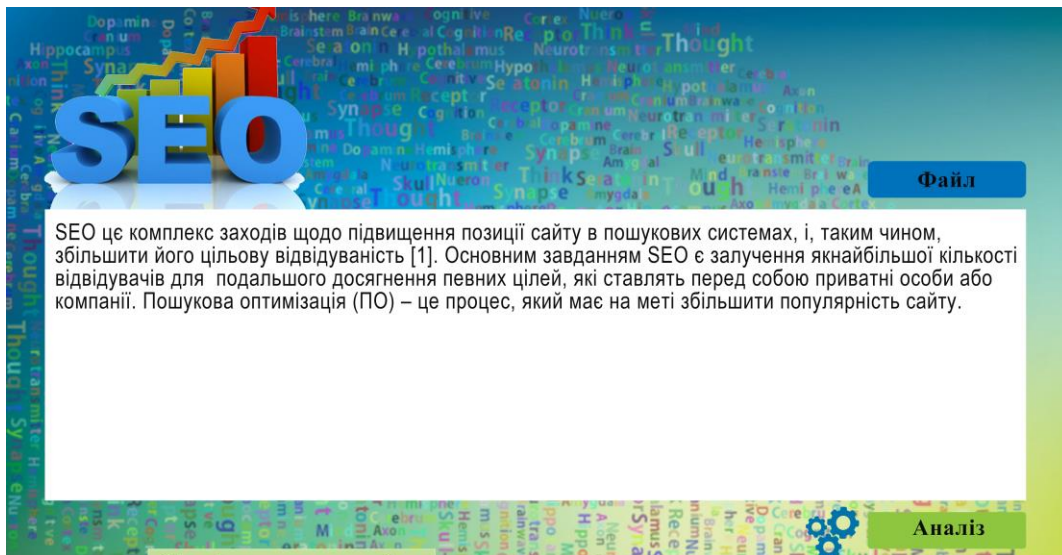


Рисунок 5.2 – Початок аналізу тексту

Після завершення аналізу тексту та розрахунку параметрів SEO-дані показників з'являться в таблиці (рис. 5.3).



Рисунок 5.3 – Приклад аналізу тексту

5.2 Параметри оцінок

Для того, щоб перевірити коректність роботи застосунку та ефективність навчених моделей було проведено порівняння оцінки різних текстів на різних мовах. Для порівняння обрано сервіс Istio.com, тому що тільки в цьому сервісі можна обрати для перевірки українську мову, та сервіс Text.ru, тому що він вираховує кількісне значення показників текстів.

Текст для тестування був обраний з моєї пояснювальною записки з глави «вступ». Результат перевірки текстів наведені в таблиці 5.1 для української мови.

Таблиця 5.1 – Результати перевірки українського тексту

Показники	Власний застосунок	Istio.com	Text.ru
Водність	8%	15%	5%
Інформативність (нудота)	100%	97% (нудота 3%)	-
Орфографія	90%	93% (7% помилок)	95% (5% помилок)
Синтаксис	96%	-	-
Заспамленість	-	-	46%
Рекламність	50%	-	-
Тональність	100%	-	-

Результат перевірки текстів наведені в таблиці 5.2 для російської мови. Російський текст був взятий зі сторінки енциклопедії «Вікіпедія» російською мовою з інформацією про SEO.

Таблиця 5.2 – Результати перевірки російського тексту

Показники	Власний застосунок	Istio.com	Text.ru
Водність	30%	34%	12%
Інформативність (нудота)	100%	97% (нудота 2.64%)	-
Орфографія	100%	93% (7% помилок)	95% (5% помилок)
Синтаксис	96%	-	-
Заспамленість	-	-	45%
Рекламність	47%	-	-
Тональність	100%	-	-

За результатами дослідження можна зробити висновок, що власний застосунок здатний оцінювати якість текстів з великою точністю. Параметр рекламність можна порівнювати з SEO-критерієм заспамленість. Перевагою застосунку між іншими є можливість оцінки текстів на українській мові. Серед недоліків треба звернути увагу на розширення інтерфейсу сервісу (додати перевірку на унікальність текстів), можливістю зберігати результати перевірки в окремі файли та можливість редагувати текст безпосередньо у вікні програми.

5.3 Висновки до розділу

В даному розділі розглянуто інтерфейс розробленого застосунку. Це односторінковий вебдодаток, який має можливість вводити текст або завантажити його з файлу з розширенням .txt. Інтерфейс досить простий, що дозволяє його використання без навчання. Додаток проводить аналіз тексту за

наступники SEO-показниками: водність, інформативність, орфографія, синтаксична коректність, рекламність, тональність тексту.

За результатами порівняння власного додатку з сервісами text.ru та Istio.com можна зробити висновок, що система здатна оцінювати якість текстів з великою точністю. Параметр рекламність можна порівнювати з SEO-критерієм запам'ятовуваність. Перевагою застосування є оцінка текстів на українській мові.

ВИСНОВКИ

В кваліфікаційній роботі розглянуті принципи пошуку вебсайтів та основні алгоритми їх ранжування. Проаналізовані різні види SEO-оптимізації вебсайтів. Досліджені показники внутрішньої оптимізації вебсайту, спрямовані на покращення сприйняття ресурсу ПС та користувачами. В результаті аналізу зроблено висновок, що актуальним фактором білої внутрішньої SEO-оптимізації вебсайтів для ПС є створення якісного, унікального, релевантного основній темі вебсайту тексту.

У другому розділі проаналізовані показники SEO-оптимізації тексту такі, як водність, нудота, заспамленість, унікальність. SEO-аналіз тексту дозволяє оцінити контент з різних факторів. Його зазвичай проводять автоматично за допомогою різних онлайн-сервісів у рамках аудиту сайту. Досліджено основні сервіси онлайн перевірки тексту (text.ru, istio.com, SEOlib, ContentWatch). Автоматична перевірка допомагає швидко виявити слабкі місця будь-якої статті. Переваги SEO-аналізу тексту за допомогою клієнт-серверного додатку: не потрібно інсталиувати додаткове програмне забезпечення на комп'ютері, перевірка займає мінімум часу, можливість редагування тексту у формі, щоб налаштувати його під потрібні параметри. Для автоматичного розрахунку параметрів тексту формально визначені всі критерії оцінки якості тексту.

Програмне забезпечення розроблено за допомогою наступних технологій: мова програмування Python, бібліотеки PyPi, gensim (перетворення модулі Word2Vec), scikit-learn (машинне навчання), фреймворк Django. Розроблено односторінковий вебдодаток, який має можливість вводити текст або завантажити його з файлу з розширенням .txt. Додаток проводить аналіз тексту за наступники SEO-показниками: водність, інформативність, орфографія, синтаксична коректність, рекламність, тональність тексту.

Проведено порівняння власного додатку з сервісами text.ru та Istio.com можна зробити висновок, що система здатна оцінювати якість текстів з великою точністю.

В даній кваліфікаційній роботі можна зробити висновок, що поставлені задачі були вирішені.

Практична цінність роботи полягає в тому, що на підставі досліджень розроблено програмне забезпечення для оцінки SEO-параметрів тексту. Розроблена система дозволяє підвищити продуктивність SEO-оптимізації сайту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Шамина, И. Total SEO часть 1. Полное практическое руководство по продвижению сайтов / И. Шамина, А. Носаченко. – Инфра-Инженерия., 2021. – 110 с.
2. Barbar, A. Search engine optimization (SEO) for websites / A. Barbar, A. Ismail // Proceedings of the 2019 5th international conference on computer and technology applications. – 2019. – Vol. 3.– P. 51-55.
3. Yuniarthe, Y. Application of artificial intelligence (AI) in search engine optimization (SEO) / Y. Yuniarthe // 2017 International conference on soft computing, intelligent system and information technology (ICSIT). – IEEE., 2017. – P. 96-101.
4. 20 Professional SEO Copywriting Tips for Writing influential Web Pages [Electronic resource]. – Access Mode: <https://www.problogbooster.com/2012/04/seo-copywriting-tips-tricks-for-professional-content-writing-web-blogs-posts-beginners.html>
5. Лебеденко, М.С. Вебометричний ранг як показник ефективності електронного ресурсу підприємства / М.С. Лебеденко // Економічний вісник Національного технічного університету України "Київський політехнічний інститут". – 2014. – № 11. – С. 401-408.
6. Смирнов, Е.А. Наглядное пособие по SEO-копирайтингу / Е.А. Смирнов. – М., 2014. – 30 с.
7. Patil, V.M. SEO: On-page+ off-page analysis / V.M. Patil, A.V. Patil // 2018 International Conference on Information, Communication, Engineering and Technology (ICICET). – IEEE, 2018. – P. 1-3.
8. Канн, С. К. Оценка параметров веб-репутации библиотечных сайтов с помощью инструментов SEO анализа / С.К. Канн // Вестник научных

конференций. – 2015. – № 2-6(2). – С. 48-51.

9. Пестерев, П. В. Анализ нововведений в алгоритмах поисковых систем Yandex и Google / П.В. Пестерев // Вестник Белгородского государственного технологического университета им. В.Г Шухова. – 2017. – №. 3. – С. 24-37

10. Иванова, А.В. Популярность поисковой системы / А.В. Иванова, Г.П. Селюкова // Актуальные вопросы науки и хозяйства: новые вызовы и решения. – 2019. – Т.3 - С. 427-431.

11. Лымарь, В.А. Эволюция алгоритмов крупных поисковых систем и принцип ранжирования / В.А. Лымарь, М.А. Елифанова // StudNet. – 2020. – Т. 3. – №. 10. – С. 96.

12. Павлов, А.В. Некоторые особенности внутренней и внешней SEO-оптимизации сайта / А.В. Павлов, Д.И. Васюнин // Труды международного симпозиума «Надежность и качество». – 2018. – Т. 1. – С. 268-271.

13. Ашанов, И. Оптимизация и продвижение сайтов в поисковых системах / И. Ашанов, А. Иванов. – Питер, 2018. – 467 с.

14. Rahan, J. Using SEO techniques Google Panda to Improve the Website / J. Rahan, M. Raanking // International Journal of Engineering Works Kambohwel Publishers Enterprise. – 2014. – Vol. 1. – P. 6-9.

15. Patil, A. Comparative Study Of Google Search Engine Optimization Algorithms: Panda, Penguin and Hummingbird [Electronic resource] / Akshita Patil; Jayesh Pamnani; Dipti Pawade // 2021 6th International Conference for Convergence in Technology (I2CT). – IEEE, 2021. – Access mode: <https://ieeexplore.ieee.org/abstract/document/9418074>

16. Blakeman, K. The future of Google is not about search [Electronic resource] / K. Blakerman // ELUCIDATE: autumn 2018. – Vol. 15. – No. 1-2. – Access mode: <https://elucidate-ukeig.org.uk/index.php/elucidate/article/view/250/245>

17. Энж, Э. SEO. Искусство раскрутки сайтов / Э. Энж, С. Спенсер, Р. Фишкин и др. – СПб. : Бхв-Петербург, 2014. – 668 с.

18. Воробьева, Д.С. Новые алгоритмы поиска Google: технология BERT / Д.С. Воробьева // Рекламный вектор - 2020: smart-коммуникации: сборник материалов XIV Всероссийской научно-практической конференции, Москва, 08–09 апреля 2020 года. – Москва: Российский университет дружбы народов (РУДН), 2020. – С. 263-264.

19. Королев, В.О. Современные особенности и методы подготовки SEO-специалиста / В.О. Королев // Наука XXI века: актуальные направления развития. – 2020. – № 1-2. – С. 115-120.

20. Шабает, М.Б. Базовая SEO оптимизация / М.Б. Шабает, И.А. Магомедов // Тенденции развития науки и образования. – 2020. – № 68-1. – С. 145-148. – DOI 10.18411/lj-12-2020-43.

21. Тапхарова, О. А. Техники поисковой оптимизации (SEO) для повышения видимости веб-сайта в поисковых системах / О.А. Тапхарова // Бизнес-образование в экономике знаний. – 2021. – № 2(19). – С. 108-111.

22. Jacson, S. The best ever SEO optimization tool online [Electronic resource] / S. Jacson. – Access mode: <https://digital-selling.org/seo-optimization-tool-online/>

23. Энж, Э. SEO – искусство раскрутки сайтов / Э. Энж, Д. Стрикчиола, С.спенсер [3-е изд. переаб. и доп.] – СПб.: БХВ-Петербург, 2017. – 816 с.

24. Михайлов, С.Н. Способ тематической кластеризации текстовых документов на основе их инфологической обработки / С.Н. Михайлов // Наукоемкие технологии. – 2012. – № 9. – С. 48-51.

25. A domain keyword analysis approach extending Term Frequency-Keyword Active Index with Google Word2Vec model / Hu K. et al. // Scientometrics. – 2018. – Vol. 114. – P. 1031-1068.

26. Поповський, О. І. Огляд програм порівняльного аналізу на збіг / О.І. Поповський // Збірник тез доповідей 2-го Кіровоградського соціально-економічного форуму «Інформаційне суспільство і влада». – Кіровоград. – 2013.

– С. 99—100.

27. Сервис онлайн проверки текста на уникальность Text.ru [Электронный ресурс]. – Режим доступа: <https://text.ru/>.

28. SEO анализ текста онлайн Istio.com [Электронный ресурс]. – Режим доступа: <https://istio.com/>.

29. SEOLib: Анализ сайтов онлайн. SEO-сервис для вебмастеров [Электронный ресурс]. – Режим доступа: <https://seolib.ru/>.

30. Content Watch [Электронный ресурс]. – Режим доступа: <https://content-watch.ru/>.

31. Вырмаскин, Н.А. Влияние структуры сайта на ранжирование в поисковых системах / Н.А. Вырмаскин // Студенческий. – 2018. – № 13-1 (33). – С. 36-38.

32. Евдокимов, Н. В. Раскрутка веб-сайта: практическое руководство по SEO 3.0 / Н.В. Евдокимов, И.В. Лебединский. – М. : Вильямс, 2013. – 288 с.

33. Design an MVC Model using Python for Flask Framework Development / Mohammad Robihul Mufid; Arif Basofi; M. Udin Harun Al Rasyid; Indhi Farhandika Rochimansyah; Abdul Rokhim // 2019 International Electronics Symposium (IES) [Electronic resource]. – Access mode: <https://ieeexplore.ieee.org/abstract/document/8901656>

34. Дронов, В. Python 3. Самое необходимое / В. Дронов. – СПб.: БХВ-Петербург, 2016. – 464 с.

35. Головатый, А. Django. Подробное руководство / А. Головатый. – М.: Символ-Плюс, 2010. – 560 с.

36. Ali, A.M. Overview and Comparison of Plagiarism Detection Tools [Electronic resource] / A. M. El T. Ali, H. M. D. Abdulla, V. Snasel // CEUR Workshop Proceedings. – 2011. – Vol. 706.— P. 161-172.— Access mode: <http://ceur-ws.org/Vol-706/poster22.pdf>.

37. Clarke, A. SEO 2018 Learn Search Engine Optimization With Smart

Internet Marketing Strateg: Learn SEO with smart internet marketing strategies / A. Clarke. - CreateSpace Independent Publishing Platform, 2017. - 224 p.

ДОДАТОК А
Програмна реалізація

Лістинг А.1 – Дообробка тексту

```

import re
import pymorphy2
from collections import OrderedDict

class PrepText:
    def __init__(self, f):
        self.__flag = False
        self.__txt = f
        self.__total_words = 0
        self.__work_file = ""
        self.__file = ""
        self.__morph = pymorphy2.MorphAnalyzer()
        self.__lsWord = {}
        self.__make_dict()
    def __make_dict(self):
        self.__txt.strip("\n")
        p, p1 = re.compile("([а-яА-Я-']+)"), re.compile("([!?!])")
        res, res1 = p.findall(self.__txt), p1.findall(self.__txt)
        if len(res) > 0:
            self.__flag = True

        for key in res:
            if key in self.__lsWord:
                value = self.__lsWord[key]
                self.__lsWord[key] = value + 1
            self.__total_words += 1
        else:
            self.__lsWord[key] = 1
            self.__total_words += 1

        for key in res1:
            key = key.lower()
            if key in self.__lsWord:
                value = self.__lsWord[key]
                self.__lsWord[key] = value + 1
            else:
                self.__lsWord[key] = 1
            self.__normalform()
    def __normalform(self):
        cr = {}

```

```

for key in self.__lsWord:
    words = self.__morph.parse(key)[0].normal_form
    cr[words] = self.__lsWord[key]
self.__lsWord = OrderedDict(sorted(cr.items()))

def get_lsWord(self):
    return self.__lsWord

def get_flag(self):
    return self.__flag

def get_words(self):
    bfr = re.sub(r'^https?:\V.*[\r\n]*', '', self.__txt, flags=re.MULTILINE)
    bfr_text = BeautifulSoup(bfr, "html.parser").get_text()
    bfr_text = re.sub("[^a-яA-Я]", " ", bfr_text)
    words = bfr_text.lower().split()
    return words

def get_txt(self):
    return self.__txt
def get_totalword(self):
    return self.__total_words

@staticmethod
def makeFtrVec(words, model, num_ftrs):
    ftrVec = np.zeros((num_ftrs,), dtype=np.dtype(float))
    nwords = 0.
    index2word_set = set(model.index2word)
    for word in words:
        if word in index2word_set:
            nwords = nwords + 1.
    ftrVec = np.add(ftrVec, model[word])
    ftrVec = np.divide(ftrVec, nwords)
    return ftrVec

```

Лістинг А.2 – Навчання класифікатора

```

import os
import nltk
from gensim.models import KeyedVectors

```

```

from gensim.models import Word2Vec
import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import Imputer
from blog.text_analysis.preparat import PrepText
from sklearn.externals import joblib

class Word2VecUsage_advert:
    __num_ftrs = int(300)
    def __init__(self):
        trains = os.path.join(BASE_DIR, "data")
        self.train = pd.read_csv(trains, header=0, delimiter="\t",
            quoting=3, low_memory=False)

        self.__model = KeyedVectors.load_word2vec_format(
            os.path.join(BASE_DIR, 'text/word2vec/test.bin'), binary=True)

        self.__forest_train()
        joblib.dump(self.__forest, 'randomforestTRAINED_.pkl')
        self.__forest = joblib.load('randomforestTRAINED_advert.pkl')

    def pred(self, txt):
        return self.__forest.predict(self.__makeFtrVec(PrepText.bfr_to_wordlist(txt),
            self.__model, self.__num_ftrs))

    def __forest_test(self):
        result = self.__forest.predict(self.__test_bfrs())
        output = pd.DataFrame(data={"id": self.test["id"], "sentiment": result})
        output.to_csv("D:/Diplom/Word2Vec_ru.tsv", index=False, sep="\t", quoting=3)

    def __forest_train(self):
        self.__forest = RandomForestClassifier(n_estimators=100)
        print("Fitting a random forest to labeled training data...")
        tr=Imputer().fit_transform(self.__train_bfrs())
        self.__forest = self.__forest.fit(tr,
            np.asarray(self.train["sentiment"], dtype=np.dtype(float)))

    def __test_bfrs(self):
        clean_test_bfrs = []
        for bfr in self.test["bfr"]:
            clean_test_bfrs.append(PrepText.bfr_to_wordlist(bfr,

```

```
remove_stopwords=True))
return self.__getAvgFtrVecs(clean_test_bfrs, self.__model, self.__num_fts)
```

```
def __train_bfrs(self):
    print("Creating average ftr vecs for train bfrs")
    clean_train_bfrs = []
    for bfr in self.train["bfr"]:
        clean_train_bfrs.append(PrepText.bfr_to_wordlist(bfr,
            remove_stopwords=False))
    self.trainDataVecs = self.__getAvgFtrVecs(clean_train_bfrs, self.__model,
        self.__num_fts)
    print(self.trainDataVecs)
    return self.trainDataVecs
```

```
def __makeFtrVec(self, words, model, num_fts):
    ftrVec = np.zeros((num_fts,), dtype=np.dtype(float))
    nwords = 0.
    index2word_set = set(model.index2word)
    for word in words:
        if word in index2word_set:
            nwords = nwords + 1.
            ftrVec = np.add(ftrVec, model[word])
    ftrVec = np.divide(ftrVec, nwords)
    return ftrVec
```

```
def __getAvgFtrVecs(self, bfrs, model, num_fts):
    counter = int(0)
    bfrFtrVecs = np.zeros((len(bfrs), num_fts), dtype=np.dtype(float))
    for bfr in bfrs:
        if counter % 1. == 0.:
            print("overview %i of %d" % (counter, len(bfrs)))
        bfrFtrVecs[counter] = self.__makeFtrVec(bfr, model, num_fts)
        counter = counter + 1
    return bfrFtrVecs
```

ДОДАТОК Б
Слайди презентації

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ SEO- АНАЛІЗУ ВЕБСАЙТУ

АВТОР: КНТ-210М

МОСКАЛЕНКО О.О.

КЕРІВНИК: К.Т.Н. ДОЦЕНТ

КАСЬЯН М.М.

Рисунок Б.1 – Слайд №1

Мета, об'єкт, предмет і задачі дослідження

Мета даної магістерської роботи – розробка компонентів SEO-оптимізації сайту.

Реалізація поставленої мети передбачає вирішення таких завдань:

- провести аналіз методів SEO-оптимізації сайту;
- дослідити внутрішню SEO-оптимізацію сайту;
- дослідити принципи SEO-аналізу тексту вебсторінки;
- розробити концептуальну модель застосунку SEO-аналізу тексту вебсторінки;
- провести аналіз існуючого інструментарію проектування програмного забезпечення вебдодатків;
- розробити програмне забезпечення перевірки SEO-параметрів тексту вебсторінки;
- дослідити результати аналізу власного застосунку в порівнянні з існуючими сервісами.

Предметом дослідження дипломної кваліфікаційної роботи магістра є SEO-аналізу тексту вебсторінки.

Рисунок Б.2 – Слайд №2

Аналіз предметної області

Методи SEO-оптимізації	Опис
Біла оптимізація	виключає застосування будь-яких способів, спрямованих на штучне збільшення позиції в пошуковій видачі. Усі методи білої оптимізації можна розділити на два основні види: внутрішні та зовнішні. Сюди входить чітка структуризація, використання унікального контенту, створення Sitemap, внутрішня перелінковка, хороший трафік, високий траст сайту, створення RSS, відстеження неякісних посилань, грамотне складання семантичного ядра
Сіра оптимізація	збільшення частоти ключових слів у текстах статей; дорвії без редиректу; купівля посилань
Чорна оптимізація	методи розкручування сайту, заборонені до застосування пошуковими системами. До найпоширеніших способів відносять спамдексинг, клоакінг, свопінг, створення фальшивих сайтів - дорвіїв, різні види спаму, наприклад, лінкбомбінг

4/15

Рисунок Б.3 – Слайд №3

Аналіз предметної області

Методи SEO-оптимізації	Опис
Біла оптимізація	виключає застосування будь-яких способів, спрямованих на штучне збільшення позиції в пошуковій видачі. Усі методи білої оптимізації можна розділити на два основні види: внутрішні та зовнішні. Сюди входить чітка структуризація, використання унікального контенту, створення Sitemap, внутрішня перелінковка, хороший трафік, високий траст сайту, створення RSS, відстеження неякісних посилань, грамотне складання семантичного ядра
Сіра оптимізація	збільшення частоти ключових слів у текстах статей; дорвії без редиректу; купівля посилань
Чорна оптимізація	методи розкручування сайту, заборонені до застосування пошуковими системами. До найпоширеніших способів відносять спамдексинг, клоакінг, свопінг, створення фальшивих сайтів - дорвіїв, різні види спаму, наприклад, лінкбомбінг

4/15

Рисунок Б.4 – Слайд №4

Параметри якісного тексту

№	Показник	Діапазон норми, %	На що вказує	Як досягти оптимального значення
1	Унікальність	90-100	Показує оригінальність тексту	Використовувати новий контент або переробляти вже існуючий
2	Нудота	9	Співвідношення ключових слів до загального обсягу тексту	Видаляти або замінити повторювані слова
3	Водність	40-60	Наявність слів, що не відповідають наданій інформації	Видаляти фрази загального значення
4	Заспамленість	3-5	Частота вживання ключових слів	Використовувати ключові слова помірно

5/15

Рисунок Б.5 – Слайд №5

Найпопулярніші сервіси оцінки тексту

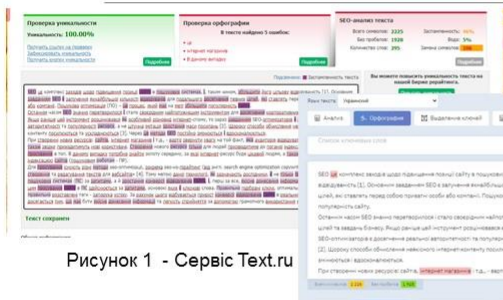


Рисунок 1 - Сервіс Text.ru

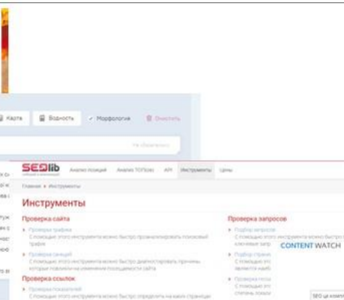


Рисунок 2 – Сервіс Istio.com

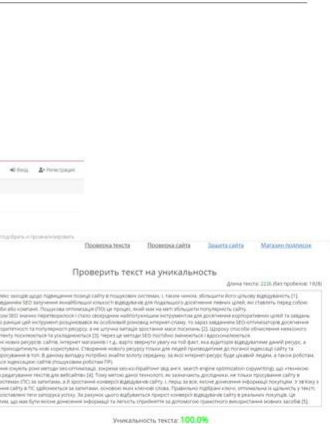


Рисунок 3 – Сервіс SEOlib



Рисунок 4 – Сервіс Content Watch

6/15

Рисунок Б.6 – Слайд №6

Результати дослідження функціональності сервісів

Критерії оцінки	Text.ru	Istio.com	SEOlub	ContentWatch
Пошук в мережі інтернет	+	+	+	+
Пошук у базах даних	+	-	+	-
Параметри оцінки	унікальність, нудота, заспамленість, обсяг тексту, заміна букв, орфографія, ключові слова	унікальність, нудота, заспамленість, обсяг тексту, орфографія, ключові слова, карта	унікальність, нудота, заспамленість, обсяг тексту, ключові слова	унікальність, ключові слова
Наявність звіту	+	+	+	-
Правила безкоштовного користування	Безкоштовно, але треба чекати в черзі на перевірку	Безкоштовно до 5000 знаків	-	У безкоштовному режимі доступно до 7 перевірок на день, із обмеженням на 10 000 знаків
Завантаження файлу / Посилання на сайт	+/-	+/+	+/+	-/+
Вибір мови тексту	-	+	-	-

7/15

Рисунок Б.7 – Слайд №7

Структура програми

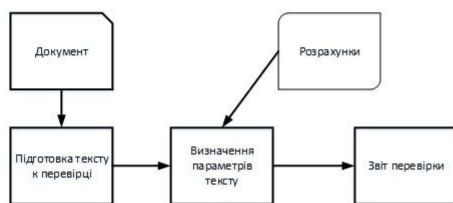


Рисунок 5 – Структура програми

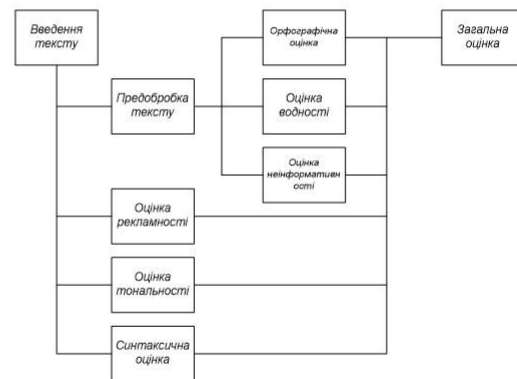


Рисунок 6 – Структура модулів програми

8/15

Рисунок Б.8 – Слайд №8

Моделювання програми



Рисунок 7 – Діаграма класів «Клієнт»

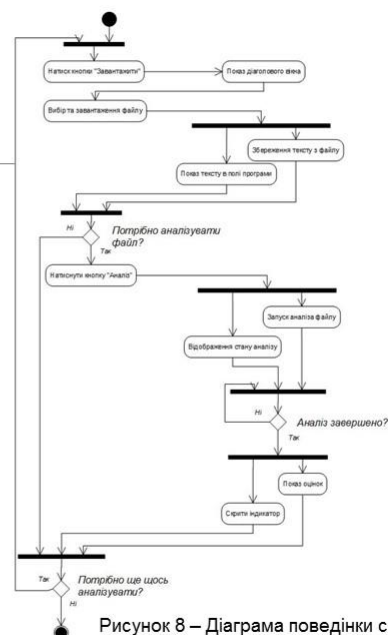


Рисунок 8 – Діаграма поведінки системи

9/15

Рисунок Б.9 – Слайд №9

Формальне визначення критеріїв SEO-аналізу

$$V = \frac{\sum w_{stop}}{\sum w_t} \quad (1)$$

де V - водність тексту, W_{stop} - кількість стоп-слів, W_t - загальний обсяг тексту.

$$OE = \frac{\sum err_o}{\sum w_t} \quad (2)$$

де OE - значення орфографії, err_o - кількість орфографічних помилок, W_t - загальний обсяг тексту.

$$SE = \frac{\sum err_s}{\sum w_t} \quad (3)$$

де SE - значення синтаксичної правильності, err_s - кількість синтаксичних помилок, W_t - загальний обсяг тексту.

$$I = \frac{\sum w_{diff}}{\sum w_t} \quad (4)$$

де I - значення інформативності, w_{diff} - кількість слів, що відрізняються, W_t - загальний обсяг слів у тексті.

10/15

Рисунок Б.10 – Слайд №10

Розробка системи

- для вирішення поставленої задачі було обрано мову програмування Python;
- в якості фреймворку обрано Django;
- дообробка тексту проводиться засобами бібліотеки рурі;
- синтаксичний аналіз тексту проходить за допомогою бібліотеки LanguageTool;
- орфографічний аналіз тексту показує наявність орфографічних помилок за допомогою бібліотеки PyEnchant;
- обчислення показника водності засобами rutmorphu2;
- тональність та рекламність розраховується засобами машинного навчання (бібліотека gensim та алгоритм Random forest).

11/15

Рисунок Б.11– Слайд №11

Інтерфейс сервісу



Рисунок 9 – Інтерфейс застосунку

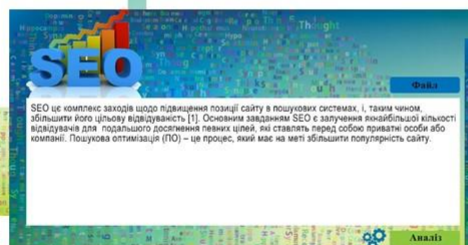


Рисунок 10 – Початок аналізу тексту



Рисунок 11 – Приклад аналізу тексту

12/15

Рисунок Б.12 – Слайд №12

Результати перевірки українського тексту

Показники	Власний застосунок	Istio.com	Text.ru
Водність	8%	15%	5%
Інформативність (нудота)	100%	97% (нудота 3%)	-
Орфографія	90%	93% (7% помилок)	95% (5% помилок)
Синтаксис	96%	-	-
Заспамленість	-	-	46%
Рекламність	50%	-	-
Тональність	100%	-	-

13/15

Рисунок Б.13 – Слайд №13

Результати перевірки російського тексту

Показники	Власний застосунок	Istio.com	Text.ru
Водність	30%	34%	12%
Інформативність (нудота)	100%	97% (нудота 2.64%)	-
Орфографія	100%	93% (7% помилок)	95% (5% помилок)
Синтаксис	96%	-	-
Заспамленість	-	-	45%
Рекламність	47%	-	-
Тональність	100%	-	-

14/15

Рисунок Б.14 – Слайд №14

Висновки

- проаналізовані різні види SEO-оптимізації вебсайтів. Досліджені показники внутрішньої оптимізації вебсайту, спрямовані на покращення сприйняття ресурсу пошуковими системами та користувачами;
- проаналізовані показники SEO-оптимізації тексту вебсторінки такі, як водність, нудота, заспамленість, унікальність;
- досліджено основні сервіси онлайн перевірки тексту (text.ru, istio.com, SEOlib, ContentWatch) та зроблено порівняльний аналіз;
- програмне забезпечення розроблено за допомогою наступних технологій: мова програмування Python, бібліотеки PyPi, gensim (Word2Vec), scikit-learn (машинне навчання), фреймворк Django;
- Проведено тестування та дослідження розробленого додатку.

В даній дипломній кваліфікаційній роботі можна зробити висновок, що поставлені задачі були вирішені.

Практична цінність роботи полягає в тому, що на підставі досліджень розроблено програмне забезпечення для оцінки SEO-параметрів тексту. Розроблена система дозволяє підвищити продуктивність SEO-оптимізації сайту.