НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

«На правах рукопису»
УДК _004.9_

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
«___»_____2024 р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою «Інформаційні управляючі системи та технології»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Інформаційна система для реферування наукових текстів»

Виконав:
студент 2 курсу, групи ІС-33мп
Толкунов Іван Сергійович

_____

Керівник:
доцент, к.ф.-м.н., доцент
Гавриленко Олена Валеріївна _____

Рецензент:
доцент кафедри ІПІ, к.т.н,
Олійник Юрій Олександрович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)
Спеціальність – 126 «Інформаційні системи та технології»
Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____Олександр РОЛІК
«___»_____20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Толкунов Івану Сергійовичу

1. Тема дисертації «Інформаційна система для реферування наукових текстів», науковий керівник дисертації Гавриленко Олена Валеріївна, к.ф.-м.н., доцент, затверджені наказом по університету від «08» 11 2024 р. № 5016-с
2. Термін подання студентом дисертації «09» 12 2024 р.
3. Об’єкт дослідження – інформаційна система для реферування наукових текстів на основі аналізу змісту та структурованих метаданих, що автоматично генеруються з наукових публікацій за допомогою спеціалізованих алгоритмів обробки тексту.
4. Вихідні дані – україномовний науковий стислий текст.
5. Перелік завдань, які потрібно розробити: розробити алгоритми автоматичного реферування наукових текстів, вибрати та реалізувати модель для аналізу

контексту та виділення ключової інформації, створити та підготувати набір даних для навчання моделі, провести тестування та аналіз помилок, порівняти ефективність створеної системи з існуючими аналогами, розробити структуру бази даних для збереження текстів і рефератів, розробити користувацький інтерфейс для зручної взаємодії з системою, структурна схема.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: UML діаграма послідовності взаємодії користувача та системи, діаграма класів, схема алгоритму BERT, схема алгоритму ланцюги Маркова, схема комбінації асоціативних правил та ланцюгів Маркова, схема алгоритму переклад тексту, схема алгоритму асоціативні правила, схема алгоритму TextRank.

7. Дата видачі завдання: 02.09.24

Календарний план

з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Огляд та аналіз існуючих рішень	05.09– 11.09	
2	Визначення завдання та цілі, аналіз існуючих алгоритмів і моделей	12.09 – 18.09	
3	Вибір відповідного алгоритму і моделі	19.09 – 25.09	
4	Створення набору даних, що використовуватиметься у навчанні моделі	26.09 – 01.10	
5	Розробка простого програмного застосунку	02.10 – 10.10	
6	Тестування та аналіз помилок при створенні застосунку	11.10 – 20.10	
7	Порівняння створеної системи з аналогами	21.10 – 22.10	
8	Оформлення щоденника, звіту і складання	23.10 – 26.10	

Студент Іван ТОЛКУНОВ

Керівник Олена ГАВРИЛЕНКО

РЕФЕРАТ

Інформаційна система для реферування наукових текстів: 123 с., 29 табл., 8 рис., 9 дод., 26 джерел.

TEXTRANK, РЕФЕРУВАННЯ, BERT, ЛАНЦЮГИ МАРКОВА, АСОЦІАТИВНІ ПРАВИЛА, ТЕКСТ

Новизна, полягає в адаптації існуючих алгоритмів реферування для роботи з україномовними науковими текстами.

Актуальність теми. Автоматизація реферування україномовних наукових текстів є актуальною задачею для сучасного наукового середовища України через зростання обсягів інформації та необхідність швидкого доступу до неї.

Мета дослідження. Створення інформаційної системи для автоматичного реферування текстів із високим рівнем точності та адаптивності до мовних і контекстуальних особливостей української мови.

Об’єкт дослідження. Україномовні наукові тексти.

Предмет дослідження. Інформаційна система реферування наукових україномовних текстів.

Завдання. Аналіз існуючих рішень та виявлення переваг та недоліків; додавання можливості перекладу реферату іноземними мовами; модифікація алгоритму реферування з використанням ланцюгів Маркова та асоціативних правил; інтеграція та вдосконалення моделей BERT і алгоритму TextRank для аналізу контексту та ключових фраз; тестування та аналіз ефективності алгоритмів.

Публікації. Толкунов І. С., Гавриленко О. В. Огляд методів реферування тексту// Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2023): матеріали V Міжнародної науково-практичної конференції (19-21 грудня 2023 р., Київ). – Київ: КПІ ім. Ігоря Сікорського, 2023. – С. 320-323.

Толкунов І. С., Бистріцький А. І., Гавриленко О. В., Богданова Н. В. Аналіз сучасних методів реферування текстів // Адаптивні системи автоматичного управління. – Вип. № 1 (46), 2025. – Рекомендовано до друку.

ABSTRACT

Information system for abstracting scientific texts: 123 pages, 29 tables, 8 figures, 8 appendices, and 26 references.

TEXTRANK, SUMMARIZATION, BERT, MARKOV CHAINS, ASSOCIATIVE RULES, TEXT

Novelty. The novelty lies in the adaptation of existing summarization algorithms for processing Ukrainian-language scientific texts.

Relevance of the Topic. Automating the summarization of Ukrainian-language scientific texts is a pertinent task for the modern scientific community in Ukraine.

Research Objective. The objective is to develop an information system for automatic summarization of texts with a high level of accuracy and adaptability to the linguistic and contextual features of the Ukrainian language.

Research Object. Ukrainian-language scientific texts.

Research Subject. An information system for summarizing Ukrainian-language scientific texts.

Tasks. Analyze existing solutions and identify their advantages and disadvantages; add the capability for translating summaries into foreign languages; modify the summarization algorithm using Markov chains and associative rules; integrate and improve BERT models and the TextRank algorithm for context and keyword analysis; test and evaluate the effectiveness of the algorithms.

Publications. Tolkunov I. S., Havrylenko O. V. Review of Text Summarization Methods // Software Engineering and Advanced Information Technologies (SoftTech-2023): Proceedings of the V International Scientific and Practical Conference (December 19-21, 2023, Kyiv). – Kyiv: Igor Sikorsky Kyiv Polytechnic Institute, 2023. – pp. 320-323.

Tolkunov I. S., Bystrytskyi A. I., Havrylenko O. V., Bohdanova N. V. Analysis of Modern Text Summarization Methods // Adaptive Automatic Control Systems. – Issue No. 1 (46), 2025. – Recommended for publication.

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

AI – Artificial Intelligence – штучний інтелект, система або програма, яка імітує людське мислення та поведінку.

API – Application Programming Interface – інтерфейс програмування застосунків.

BERT – Bidirectional Encoder Representations from Transformers – двонаправлені подання енкодера з трансформерів.

CSS – Cascading Style Sheets – каскадні таблиці стилів, що використовуються для оформлення веб-сторінок.

GPT – Generative Pretrained Transformer – генеративний попередньо навчений трансформер, модель для роботи з текстами.

HTML – HyperText Markup Language – мова розмітки гіпертексту для створення веб-сторінок.

HTTP – HyperText Transfer Protocol – протокол передачі гіпертексту, що використовується в Інтернеті.

JSON – JavaScript Object Notation – формат обміну даними, що базується на тексті.

ML – Machine Learning – машинне навчання, підгалузь штучного інтелекту, яка вивчає алгоритми для автоматичного навчання систем.

NLP – Natural Language Processing – обробка природної мови, що використовується для аналізу та синтезу текстів.

REST – Representational State Transfer – стиль організації веб-застосунків, що описує структуру і взаємодію компонентів.

SQL – Structured Query Language – мова структурованих запитів, за допомогою якої можна взаємодіяти із базами даних.

TF-IDF – Term Frequency-Inverse Document Frequency – частотність термів – обернена частотність у документах.

ЗМІСТ

ВСТУП

Сучасний науковий прогрес супроводжується значним зростанням обсягів наукової інформації, що ускладнює її аналіз та ефективну організацію. Особливо це стосується української науки, яка стрімко інтегрується у глобальний науковий простір, водночас стикаючись із необхідністю зберігати національні мовні та контекстуальні особливості. У таких умовах актуальним завданням стає розробка інструментів автоматизації обробки наукових текстів для створення точних, стислих і структурованих рефератів.

Існуючі системи автоматичного реферування текстів здебільшого орієнтовані на англомовні дані, що зумовлює їхню обмежену ефективність для української мови. Специфіка граматики, синтаксису та наукового стилю української мови вимагає адаптації алгоритмів і моделей, які застосовуються для автоматичного реферування текстів. Відсутність локалізованих рішень створює суттєві бар’єри для ефективного аналізу україномовних наукових текстів.

Мета дослідження полягає у створенні інформаційної системи, що забезпечує автоматичне реферування україномовних наукових текстів з використанням сучасних алгоритмів обробки мови. Також система повинна надавати переклад реферату, адже все більше робіт українських наукових діячів з’являються і обговорюються на всесвітніх конференціях.

Актуальність розробки полягає у вирішенні проблеми швидкої обробки великих обсягів текстів, підвищенні точності аналізу та створенні інструменту, який відповідає потребам українського наукового середовища. Запропоноване рішення сприятиме оптимізації роботи наукових установ, університетів та інформаційних центрів, забезпечуючи високу якість автоматизованих рефератів, адаптованих до лінгвістичних особливостей української мови.

Робота передбачає виконання наступних завдань:

– аналіз існуючих підходів до автоматичного реферування текстів;

– розробка алгоритмів, що інтегрують кілька підходів для підвищення точності та релевантності результатів;

– створення програмного застосунку з дружнім інтерфейсом для завантаження текстів і налаштування параметрів реферування;

– проведення тестування та оцінювання якості результатів, а також порівняння системи з аналогами;

– переклад рефератів на іноземні мови.

Результатом дослідження стане інтегрована інформаційна система для автоматичного реферування україномовних наукових текстів, що забезпечує високу точність, швидкість та адаптивність до потреб користувачів.

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Існуючі рішення систем для реферування текстів

Системи автоматичного узагальнення текстів є важливим кроком у розумінні сучасних тенденцій і технологій в цій галузі. З огляду на стрімке зростання обсягу текстової інформації в різних галузях, таких як новини, наукові публікації, соціальні мережі та корпоративна документація, виникає гостра необхідність у розробці систем, здатних автоматично аналізувати та узагальнювати великі масиви текстових даних. Це дозволяє не лише економити час на обробку інформації, але й підвищувати якість прийняття рішень на основі стислих і релевантних даних.

На сьогоднішній день існує широкий спектр рішень для автоматичного узагальнення текстів, які базуються на різних підходах, таких як статистичні методи, методи обробки природної мови (NLP), а також сучасні моделі на основі машинного навчання і глибокого навчання. Кожен з цих підходів має свої переваги і недоліки, залежно від специфіки застосування та типу текстових даних.

1.1.1 Система реферування текстів QuillBot

QuillBot є одним із провідних інструментів для автоматизації роботи з текстами, що базується на сучасних методах обробки природної мови (NLP) та глибокого навчання. Цей інструмент використовується для перефразування текстів, їх узагальнення, перевірки граматики та стилю, а також автоматизації цитування [1]. QuillBot підтримує інтеграції з популярними платформами, такими як Google Docs та Microsoft Word, що дозволяє його безперебійно використовувати у процесі створення текстових документів.

На рис 1.1. можна побачити інтерфейс QuillBot. Сервіс надає користувачам кілька ключових функцій:

– перефразування тексту;

– узагальнення текстів;

– перевірка граматики та стилю;

– генерація цитат.

Цей інструмент є одним із найпопулярніших для обробки текстів завдяки своїй простоті використання та універсальності. Вільний доступ до базових функцій робить його доступним для широкого кола користувачів, тоді як преміум-версія пропонує додаткові можливості, такі як розширені режими перефразування, збільшення обсягу тексту для обробки і налаштування рівня узагальнення. Його інтеграція з різними платформами для редагування текстів також забезпечує зручність у роботі над документами [1].

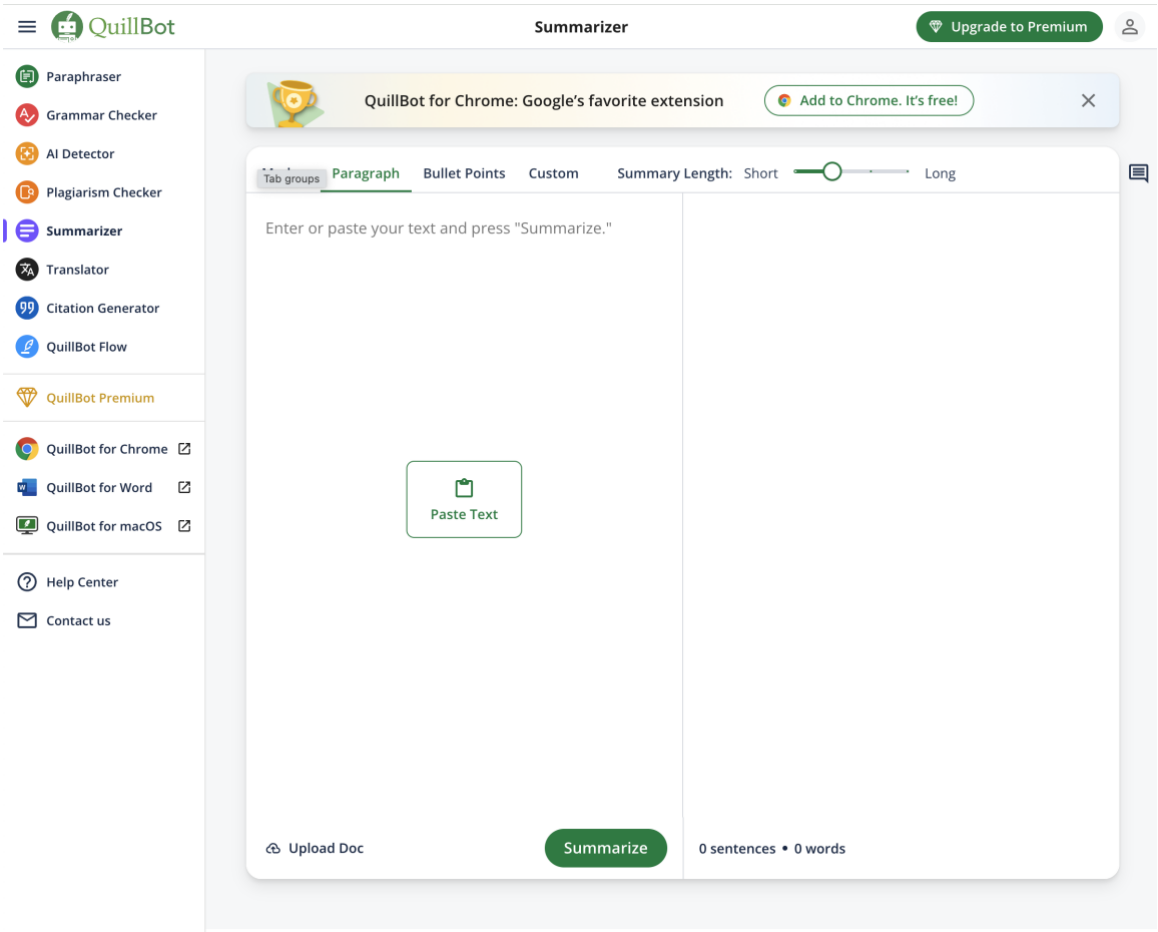

Рисунок 1.1 – Інтерфейс QuillBot

Основою QuillBot є нейронні мережі, які використовують техніки глибокого навчання для опрацювання і генерації тексту. Зокрема, використовуються трансформерні моделі, такі як BERT (Bidirectional Encoder Representations from

Transformers) і GPT (Generative Pretrained Transformer) [2]. Ці моделі можуть опрацьовувати великі масиви тексту, розуміючи контекст як попередніх, так і наступних слів у реченні, що дозволяє створювати перефразовані тексти або резюме з високою точністю.

Трансформери, на відміну від традиційних рекурентних нейронних мереж (RNN), ефективніше обробляють довгі послідовності слів завдяки механізму self-attention. Цей механізм дозволяє кожному слову в тексті взаємодіяти з іншими словами, враховуючи їхнє значення в загальному контексті. У випадку з QuillBot, це означає, що при перефразуванні інструмент не лише змінює слова на синоніми, але й зберігає зміст тексту на всіх рівнях: лексичному, синтаксичному і семантичному.

Моделі, що базуються на трансформерах (наприклад, BERT або GPT), не завжди здатні глибоко зрозуміти семантичні нюанси тексту. Незважаючи на здатність до обробки складних контекстів, ці моделі можуть втрачати важливі аспекти смислу при перефразуванні або узагальненні, особливо у випадках зі складними або двозначними текстами. Це може призвести до спотворення оригінального сенсу або втрати важливої інформації.

1.1.2 Система реферування текстів Scholarcy

Scholarcy – це інструмент на основі штучного інтелекту, призначений для автоматичного узагальнення та аналізу академічних текстів. Він розроблений спеціально для студентів, дослідників і науковців, щоб допомогти в обробці складних наукових публікацій, розділів книг, звітів та інших документів великого обсягу. Система автоматично виділяє ключову інформацію з текстів, полегшуючи сприйняття великих наукових робіт та економлячи час на їх аналіз.

Основні можливості:

– автоматичне узагальнення наукових статей;

– екстракція цитат та посилань;

– підсвічування ключових термінів та концепцій;

– інтеграція з іншими інструментами.

Технічно Scholarcy використовує методи обробки природної мови (NLP) і поєднує екстрактивні та абстрактивні методи узагальнення. Екстрактивні методи відбирають найважливіші речення з тексту, тоді як абстрактивні здатні переформулювати речення для стислого відображення основного змісту.

До переваг Scholarcy належить економія часу, зручний інтерфейс користувача рис 1.2, зручна робота з різними форматами документів (PDF, Word), а також спеціалізація на академічних матеріалах, що робить його корисним для дослідницьких робіт. Водночас інструмент має кілька недоліків. По-перше, Scholarcy може мати обмежену ефективність при роботі з дуже великими документами, такими як монографії або багатотомні праці, де можливості узагальнення можуть бути недостатніми. По-друге, як і всі системи на основі штучного інтелекту, він не є ідеальним і може пропускати важливі деталі або робити помилки. Тому результати, що генеруються Scholarcy, потребують людської перевірки та додаткового аналізу для забезпечення їхньої точності та повноти.

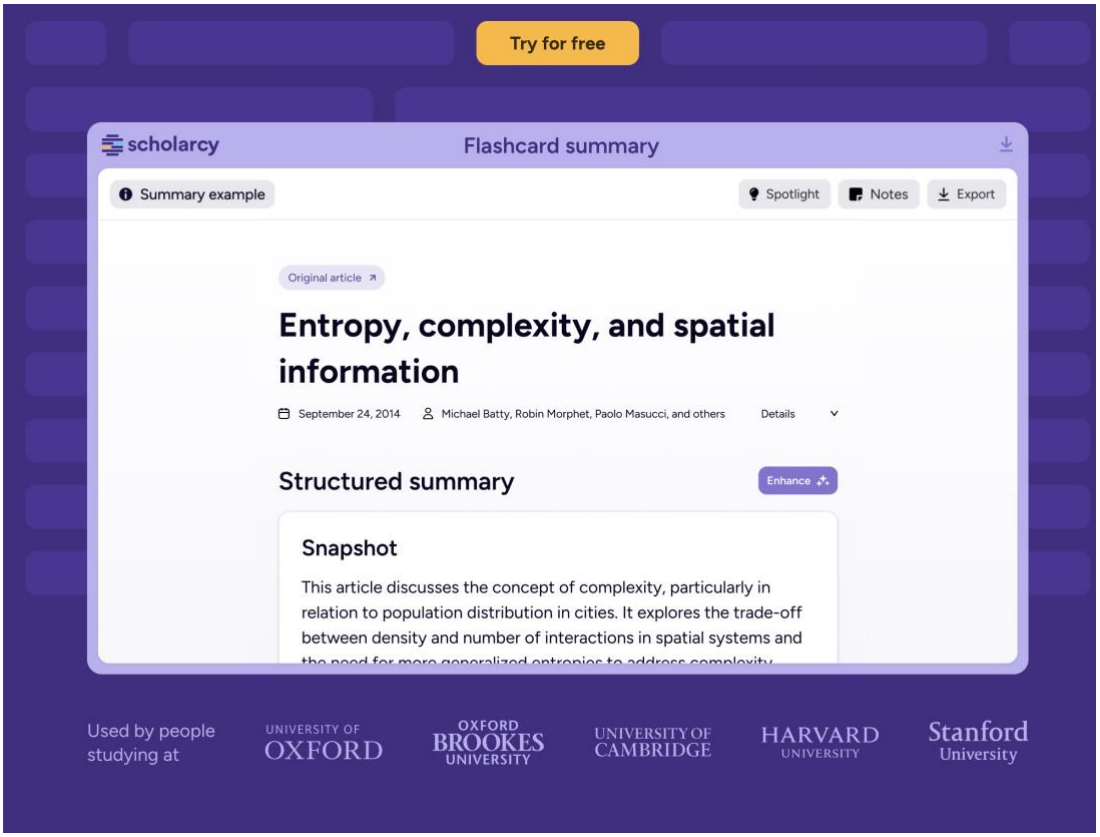

Рисунок 1.2 – Інтерфейс Scholarcy

1.1.3 Сервіс для скорочення тексту SMMRY

Простий і ефективний інструмент для автоматичного узагальнення текстів, що фокусується на екстракції найважливіших речень із введеного матеріалу. Основне завдання SMMRY – скорочення тексту до ключових моментів, забезпечуючи користувачів короткими і чіткими резюме. Інструмент використовує екстрактивний підхід до узагальнення, що означає вибір найбільш значущих речень без переформулювання оригінального змісту [3].

Одна з ключових функцій SMMRY – можливість регулювати довжину резюме, надаючи користувачам контроль над кількістю речень у підсумковому тексті. Інструмент підтримує вставку тексту вручну або через URL-адресу вебсторінки, інтерфейс системи можна побачити на рис 1.3. Гибкий набір способів надання тексту дає змогу легко створювати резюме для статей в Інтернеті. Крім того, SMMRY пропонує API для інтеграції в сторонні додатки, що робить його корисним для розробників.

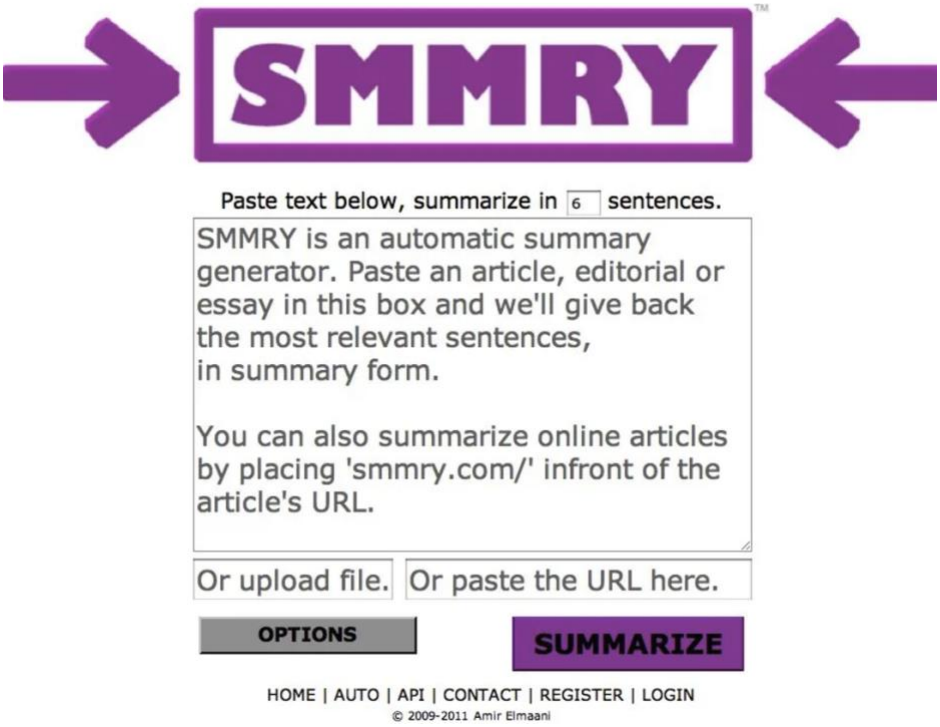

Рисунок 1.3 – Інтерфейс SMMRY

SMMRY має кілька ключових переваг, які роблять його популярним серед користувачів. Перш за все, це простота використання та швидкість роботи. Інтерфейс інструменту інтуїтивно зрозумілий і не вимагає особливих навичок для користування: достатньо вставити текст або URL-адресу для отримання швидкого резюме. Крім того, інструмент доступний безкоштовно, що дозволяє широкому колу користувачів скористатися його можливостями без необхідності купівлі ліцензії чи преміум-доступу. Завдяки своїм можливостям автоматичного видалення другорядних елементів, таких як приклади чи повторення, SMMRY може значно скоротити час на обробку тексту, особливо для користувачів, яким потрібно швидко ознайомитися з основними ідеями великого обсягу тексту.

Однак, незважаючи на свої переваги, SMMRY має певні недоліки. Один із них полягає в обмеженнях екстрактивного підходу, який використовує інструмент. Оскільки SMMRY лише виділяє найважливіші речення з тексту, він не переформулює їх і не узагальнює зміст у повному сенсі, що може призводити до появи тексту з недостатньою зв'язністю або до втрати важливих деталей, які не

були явно виражені у відокремлених реченнях [4]. Інструмент також має обмежену підтримку різних форматів, таких як PDF та Word, що вимагає від користувачів додаткових дій для конвертації файлів у текстовий формат. Крім того, SMMRY не здатний ефективно аналізувати складні текстові структури або зв'язки між частинами тексту, що може бути критично для наукових або технічних документів, де важлива глибока семантична обробка.

1.1.4 Інструмент для створення рефератів Resoomer

Інструмент Resoomer призначений для автоматичного узагальнення текстів, допомагаючи спрощувати складні документи шляхом виділення головних ідей та фактів. Він став популярним серед студентів, дослідників і журналістів, які потребують швидкого доступу до ключових тез у великих текстах. Алгоритми Resoomer використовують екстрактивний метод узагальнення, що означає вибір найважливіших речень чи фраз з оригінального тексту без зміни їх структури. Основне завдання інструменту – знаходження ключових слів та зв'язків між ними для визначення найважливіших частин тексту [5].

Алгоритм Resoomer базується на лексичному аналізі, який визначає важливість певних слів і фраз на основі їх частоти та розташування в тексті. Інструмент аналізує синтаксичну структуру речень, щоб відкидати незначні елементи, такі як перехідні фрази, пояснення або приклади, які не несуть основного смислового навантаження. Крім того, Resoomer використовує семантичний аналіз для розуміння контексту тексту і вибору речень, які найбільше відповідають загальній темі документа. Це дозволяє інструменту ефективно виділяти ключові моменти і позбуватися другорядних деталей, що є корисним при роботі з науковими, історичними або аргументативними текстами.

З технічного боку Resoomer працює на основі поєднання методів обробки природної мови (NLP) і статистичних моделей. Це забезпечує можливість обробки текстів різних тематик і складності. Проте, як і всі екстрактивні алгоритми,

Resoomer має певні обмеження. Інструмент не створює новий текст на основі прочитаного, а лише відбирає вже існуючі частини, що може призводити до недостатньої когерентності резюме або до втрати важливих деталей, якщо вони не явно виражені в окремих реченнях [5].

До основних переваг Resoomer можна віднести його висока швидкість обробки текстів і зручний інтерфейс який можна побачити на рис 1.4. Інструмент добре підходить для роботи з науковими та аргументативними текстами, де важливо швидко отримати основні ідеї. Водночас його екстрактивний підхід має свої недоліки, оскільки Resoomer не перефразовує речення, а лише відбирає важливі частини, він може створювати текст, який недостатньо зв’язаний або не відображає повну картину документа. Також інструмент може не враховувати важливі зв'язки між різними частинами тексту, що може вплинути на якість узагальнення, особливо у випадку складних документів.

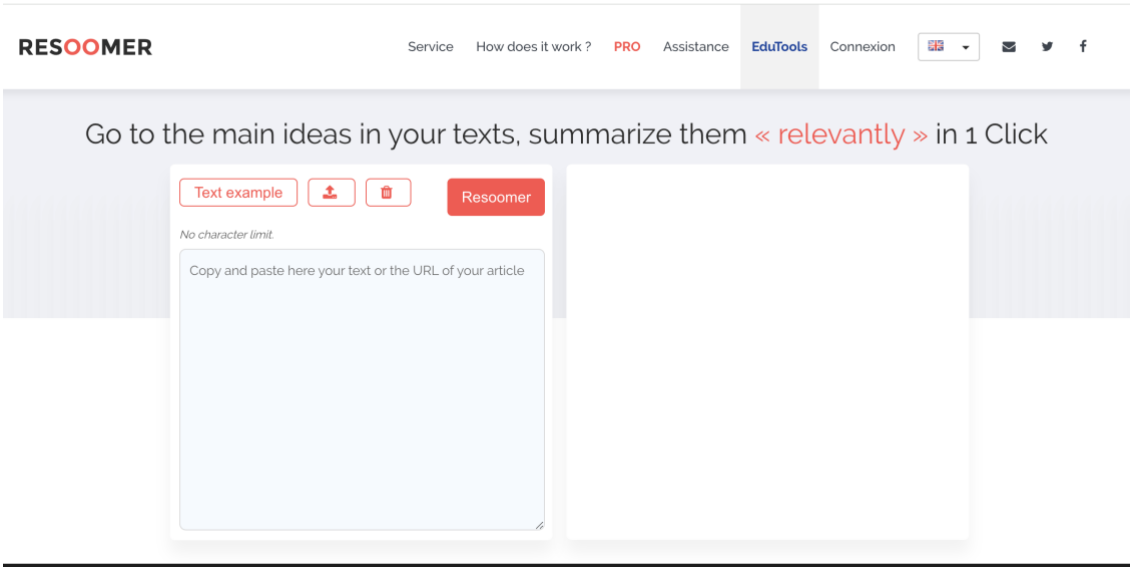

Рисунок 1.4 – Інтерфейс Resoomer

Висновки до розділу

Сучасні системи автоматичного узагальнення текстів значно покращують обробку великих обсягів інформації, пропонуючи користувачам ефективні інструменти для роботи з різними типами документів. Інструменти, такі як

QuillBot, Scholarcy, SMMRY та Resoomer, демонструють широкий спектр можливостей завдяки використанню різних методів обробки природної мови та моделей глибокого навчання. Кожен з них володіє унікальними функціями, що відповідають на різні потреби користувачів, будь то перефразування, узагальнення наукових статей або швидке створення резюме тексту.

QuillBot виділяється своєю універсальністю, пропонуючи широкий спектр функцій — від перефразування до перевірки граматики. Scholarcy фокусується на обробці наукових публікацій, автоматизуючи аналіз та виділення ключових елементів із академічних текстів. SMMRY надає прості та швидкі рішення для узагальнення текстів, хоча його екстрактивний підхід має обмеження в зв'язності резюме. Resoomer, зі свого боку, пропонує можливості для аналізу складних документів, проте також базується на екстрактивному підході, що не завжди дозволяє зберегти повний контекст.

Усі ці інструменти мають спільні риси та недоліки, пов’язані з їх екстрактивними підходами, що часто призводить до втрати частини контексту або важливих деталей. Більш детальний розгляд можна побачити у табиці 1.1.

Таблиця 1.1 – Переваги та недоліки розглянутих систем

Сервіс	Переваги	Недоліки	Проблеми з українською мовою
QuillBot	Універсальність, простота використання, доступ до широких функцій	Можливі втрати семантичних нюансів і спотворення змісту, потреба в перевірці результатів	Алгоритми QuillBot н завжди коректно обробляют українські тексти, щ призводить до помилок синтаксисі, стилі та втрат сенсу
Scholarcy	Оптимізація для наукових публікацій, зручність роботи з великими обсягами текстів, підтримка різних форматів	Обмеження в обробці надто великих документів, можливість втрати важливих деталей, необхідність	Алгоритми недостатньо адаптовані до української академічної термінології, що ускладнює точне виділення ключових елементів

Сервіс	Переваги	Недоліки	Проблеми з українською мовою
		перевірки результатів	
SMMRY	Швидкість, простота використання, можливість інтеграції через API	Відсутність узагальнень у повному сенсі, проблеми зі зв’язністю тексту, обмежена підтримка форматів документів	Екстрактивний підхід не завжди ефективно працює з українськими текстами, що призводить до втрати важливих контекстів і деталей
Resoomer	Висока швидкість обробки, ефективність для наукових та історичних текстів, зручний інтерфейс	Текст резюме може бути недостатньо зв’язним, можливість втрати важливих деталей, особливо у складних документах	Алгоритми не повністю враховують синтаксичну специфіку української мови, що може призводити до втрати змісту і когерентності тексту

ОГЛЯД ІСНУЮЧИХ АЛГОРИТМІВ РЕФЕРУВАННЯ ТЕКСТІВ

1.2 Існуючі алгоритми реферування текстів

1.2.1 Абстрактивні алгоритми

Абстрактивні алгоритми є одним із найпросунутіших підходів у автоматичному реферуванні текстів, що відрізняються від екстрактивних методів своєю здатністю не лише вилучати готові фрагменти з тексту, але й створювати нові речення. Це дозволяє таким алгоритмам генерувати резюме, яке виглядає більш природно і зв'язно, оскільки текст узагальнюється на основі розуміння його змісту, а не просто шляхом механічного відбору речень.

Принцип роботи абстрактивних алгоритмів полягає в застосуванні глибоких нейронних мереж та сучасних моделей обробки природної мови, таких як архітектури трансформерів, зокрема, моделі на зразок BERT, GPT або T5 [5]. Ці моделі аналізують текст як єдину структуру, враховуючи семантичні зв’язки між словами, реченнями та навіть абзацами. Такий підхід дозволяє алгоритмам краще розуміти контекст і генерувати нові фрази, що чітко відображають основну думку тексту, забезпечуючи зв'язність і логічність узагальнень. Основною перевагою абстрактивних алгоритмів є їхня здатність до глибшого аналізу тексту, що дозволяє створювати резюме, яке не тільки стисло передає зміст, але й робить це у природній мовній формі, що наближається до людського реферування.

Однією з головних сильних сторін абстрактивних алгоритмів є їхня здатність генерувати зв'язні та логічно послідовні тексти. Оскільки такі алгоритми не обмежуються копіюванням речень з оригіналу, вони можуть синтезувати нові, більш стисливі формулювання, що відображають головні ідеї документа. Це забезпечує високу якість узагальнення, особливо у випадках, коли важливо зберегти логіку та послідовність подання інформації. Гнучкість є ще однією перевагою абстрактивних алгоритмів [5], вони можуть переформулювати текст, виділяючи ключові ідеї навіть у випадках, коли екстрактивні методи не дають чіткої інформації через невідповідність фрагментів оригіналу.

Однак, абстрактивні алгоритми також мають свої слабкі сторони. Вони вимагають значних обчислювальних ресурсів для навчання і роботи, оскільки їхня складна архітектура потребує потужних процесорів або графічних процесорів, а також великих обсягів оперативної пам’яті. Це робить абстрактивні методи більш затратними в порівнянні з екстрактивними підходами, які є значно простішими і швидшими в реалізації. Крім того, навчання таких моделей вимагає великих наборів даних, які повинні містити достатньо прикладів різних текстових структур для досягнення високої точності.

Ще однією слабкою стороною абстрактивних алгоритмів є складність їх реалізації. Розробка таких моделей вимагає значних зусиль з боку інженерів і дослідників, оскільки потрібно не тільки правильно налаштувати модель, але й забезпечити адекватне навчання на відповідних даних [6]. Крім того, навіть після тренування абстрактивні алгоритми можуть генерувати текст із помилками, особливо коли стикаються зі складними або неоднозначними формулюваннями в оригінальному тексті. Це може призвести до того, що резюме буде містити семантичні неточності або невірно передавати основну ідею [5]. Підсумок розглянутих абстрактивних алгоритмів можна побачити у таблиці 2.1.

Таблиця 2.1 – Переваги та недоліки абстрактивних алгоритмів

Аспект	Переваги	Недоліки
Принцип роботи	Використання глибоких нейронних мереж (BERT, GPT, T5), які забезпечують аналіз тексту як єдиної структури.	Висока потреба в обчислювальних ресурсах для навчання і роботи.
Якість узагальнення	Генерація зв'язного та логічно послідовного тексту. Стисло і природно передає основний зміст документа.	Можливі семантичні неточності при обробці складних або неоднозначних формулювань.
Гнучкість	Здатність переформульовувати текст. Можливість виділяти ключові ідеї	Складність реалізації алгоритмів, що вимагає

Аспект	Переваги	Недоліки
	навіть у неструктурованих текстах.	високої кваліфікації інженерів і дослідників.
Ефективність	Висока якість узагальнення завдяки глибшому розумінню змісту тексту.	Залежність від великих обсягів даних для навчання.
Затратність	Забезпечує результати, наближені до людського реферування.	Значні витрати ресурсів (процесорних, графічних та оперативної пам'яті).
Сфери застосування	Узагальнення складних текстів (наукові статті, новини, аналітика). Підтримка зв'язності та логіки.	Не завжди забезпечують точність у специфічних або технічних галузях через недостатню навчальну вибірку.
Можливі помилки	Здатність до створення нових речень покращує зв'язність і природність тексту.	Можливі помилки у формулюваннях або пропуск важливих деталей через неправильне розуміння контексту.

1.2.2 Екстрактивні алгоритми

Екстрактивні алгоритми ґрунтуються на виборі найважливіших фрагментів або речень із вихідного тексту для створення резюме, зберігаючи при цьому їхню первісну форму та структуру. На відміну від абстрактивних методів, екстрактивні алгоритми не генерують нові речення, а використовують готові елементи тексту. Це робить їх швидкими та простими в реалізації, що особливо корисно для завдань, де потрібна висока продуктивність і швидкість обробки [7].

Екстрактивні методи зазвичай базуються на статистичних показниках, таких як частота слів або позиція речень у тексті. Наприклад, чим частіше певне слово зустрічається в документі, тим важливішим воно може бути для тексту, тому речення, яке його містить, може бути обране для резюме. Також часто використовується метод позиційного аналізу, коли важливими вважаються речення

на початку та наприкінці абзаців, оскільки вони зазвичай містять узагальнення або ключові висновки [6]. Одним з ефективних інструментів для оцінки важливості слів є TF-IDF, який дозволяє визначати унікальні терміни для конкретного тексту.

Перевагою даних алгоритмів є простота та ефективність. Вони не потребують глибоких мовних моделей або складних нейронних мереж, тому вимагають значно менших обчислювальних ресурсів ніж розглянутів вище алгоритми. Це робить їх придатними для використання в реальних умовах, де швидкість та економічність є важливими факторами. Крім того, екстрактивні алгоритми не вимагають значного навчання або налаштування, що спрощує їхнє впровадження у різні системи реферування текстів [8].

Одним із ключових недоліків є те, що резюме, створені за допомогою екстрактивних методів, можуть втрачати логічну зв'язність. Оскільки алгоритм просто витягує окремі речення з тексту, вони можуть бути вирвані з контексту, що іноді призводить до фрагментації резюме або втрати загальної смислової структури. Також екстрактивні алгоритми не враховують семантичні зв'язки між реченнями, що може ускладнити узагальнення складних текстів.

Інша слабка сторона полягає у обмеженій адаптивності цих методів до різних типів текстів. Алгоритми часто базуються на поверхневих ознаках, як-от частота слів або позиція речення, вони можуть не забезпечити належного узагальнення для текстів із складною або специфічною структурою, як, наприклад, художня література чи тексти з великою кількістю контекстуальних деталей. Детальний розгляд слабких і сильних сторін можна побачити у таблиці 2.2.

Таблиця 2.2 - Переваги та недоліки екстрактивних алгоритмів

Аспект	Переваги	Недоліки
Принцип роботи	Вибір найважливіших фрагментів або речень із тексту з їхньою первісною формою та структурою.	Не генерують нові речення, а лише вибирають готові елементи тексту.

Аспект	Переваги	Недоліки
Якість узагальнення	Швидкість і простота реалізації забезпечують високу продуктивність і економічність.	Можуть втрачати логічну зв'язність у створених резюме.
Гнучкість	Не потребують складних налаштувань або значного навчання.	Не враховують семантичні зв'язки між реченнями.
Ефективність	Можливість використання в умовах обмежених ресурсів.	Обмежена здатність до узагальнення складних текстів.
Затратність	Вимагають значно менших обчислювальних ресурсів порівняно з абстрактивними алгоритмами.	Можливість фрагментації резюме через виривання речень з контексту.
Сфери застосування	Придатні для автоматизації узагальнення текстів у реальному часі.	Складно забезпечити якісне узагальнення для текстів зі складною структурою, як художня література.
Можливі помилки	Збереження оригінальної форми речень забезпечує точність передання змісту.	Поверхневий підхід (частота слів, позиція речень) може бути неефективним для складних документів.

1.2.3 Ланцюги Маркова

Ланцюги Маркова є ймовірнісною моделлю, яка використовується для моделювання систем, де ймовірність переходу до наступного стану залежить лише від поточного стану, а не від усієї історії попередніх станів. Ця властивість, відома як "марковська властивість", робить ланцюги Маркова корисними для моделювання процесів, що розвиваються поступово, з чіткою залежністю між послідовними етапами. У контексті обробки текстів, ланцюги Маркова дозволяють прогнозувати ймовірність появи певних слів або речень на основі попереднього контексту.

Однією з найпростішої і водночас ефективної форми ланцюгів Маркова є двохмістні ланцюги, або біграми, які враховують тільки два послідовні стани, наприклад, два слова або речення. У текстових задачах це означає, що кожне наступне слово прогнозується на основі попереднього слова. Наприклад, якщо відомо, що слово "інформаційна" часто супроводжується словом "система", то двохмістний ланцюг Маркова може використати цю залежність для побудови ймовірного продовження тексту. Це підвищує зв'язність тексту, зберігаючи основні логічні зв'язки між його елементами [9].

У процесі роботи ланцюги Маркова використовують матрицю ймовірностей переходів між станами. Ця матриця відображає, з якою ймовірністю кожне слово, або інший елемент тексту, може бути наступним на основі попереднього. Завдяки такій структурі, система здатна визначати не лише найбільш ймовірні наступні елементи, але й будувати послідовності, які виглядають природними.

Ланцюги Маркова знайшли своє застосування в різних системах обробки природної мови, зокрема в задачах реферування текстів. Завдяки ймовірнісній природі, данна ймовірнісна модель дозволяють прогнозувати наступні слова або фрази, що допомагає зберігати логічну зв'язність під час скорочення тексту або створення його стислого резюме.

Основні аспекти використання ланцюгів Маркова в реферуванні текстів:

– прогнозування наступних слів або фраз;
− узгодженість тексту;
− обмежена обчислювальна складність.

Як можна побачити переваги ланцюгів Маркова полягають у їхній простоті та швидкості роботи. Вони легко реалізуються і мають порівняно низькі вимоги до обчислювальних ресурсів, що робить їх придатними для використання в умовах, де обмежені обчислювальні ресурси. Ймовірнісна модель добре підходять для моделювання короткострокових залежностей між словами, що корисно в задачах реферування текстів, оскільки дозволяє зберігати зв'язність між частинами тексту.

Головним недоліком ланцюгів Маркова можна вважати залежність між поточним і попереднім станами, це обмежує їхню здатність враховувати глобальний контекст тексту. У випадках складних документів або текстів із глибокими смисловими зв’язками це може призводити до втрати важливої інформації. Також ланцюги Маркова іноді призводять до повторюваності або генерувати менш різноманітні послідовності слів, що негативно впливає на якість автоматичного реферування. У таблиці 2.3 можна побачити слабкі та сильні сторони ймовірнісної моделі.

Таблиця 2.3 – Переваги та недоліки ланцюгів Маркова

Аспект	Переваги	Недоліки
Принцип роботи	Ймовірнісна модель, що прогнозує наступний стан лише на основі поточного (марковська властивість).	Залежність лише від поточного стану, ігнорування глобального контексту тексту.
Якість узагальнення	Забезпечує узгодженість тексту та логічність короткострокових залежностей.	Можуть втрачати важливі смислові зв'язки у складних текстах.
Гнучкість	Легко реалізуються, добре працюють з короткостроковими залежностями між словами.	Обмеження в роботі з текстами із глибокими смисловими зв'язками.
Ефективність	Швидкість роботи завдяки простоті алгоритму.	Іноді призводять до повторюваності слів або шаблонності тексту.
Затратність	Мають низькі вимоги до обчислювальних ресурсів, придатні для обмежених потужностей.	Не завжди забезпечують різноманітність у генерованих текстах.

Аспект	Переваги	Недоліки
Сфери застосування	Використовуються для прогнозування слів/фраз та реферування текстів.	Менш ефективні для документів із довгими контекстами.
Можливі помилки	Природність побудови коротких текстів завдяки врахуванню ймовірностей.	Можуть неадекватно узагальнювати складні або неоднозначні тексти.

1.2.4 Асоціативні правила

Асоціативні правила є ефективним методом для аналізу текстових даних та виявлення прихованих закономірностей між різними елементами тексту. Цей підхід, який активно використовується в дата-майнінгу, дозволяє виявляти взаємозв'язки між словами, фразами або концепціями, що зустрічаються разом у текстах. Алгоритм базуються на таких концепціях, як підтримка (support) та довіра (confidence), які вимірюють частоту одночасної появи елементів та ймовірність їхнього співіснування. У контексті автоматизованого реферування цей метод дає змогу знаходити важливі взаємозв'язки між словами або темами, що сприяє побудові більш точного та змістовного резюме тексту [11].

Процес застосування асоціативних правил для реферування тексту починається з аналізу частотності елементів. Підтримка використовується для виявлення найбільш поширених слів або фраз, що зустрічаються в тексті. Далі, за допомогою довіри визначається, наскільки надійно один елемент тексту, наприклад, слово або фраза, може бути пов'язаний з іншим. Ці дві метрики дозволяють побудувати набори асоціативних правил, що ідентифікують ключові зв’язки між різними елементами тексту.

Окрім простого аналізу частоти, алгоритм дозволяють виявляти приховані зв’язки, які можуть не бути очевидними при звичайному лінійному аналізі. Наприклад, два ключові терміни можуть з'являтися в різних частинах тексту, але

завдяки високій кореляції між ними можна припустити, що вони мають важливий спільний контекст [11]. У цьому випадку алгоритм допоможе виявити ці взаємозв'язки та зробити резюме більш інформативним.

Головною перевагою асоціативних правил є їх здатність враховувати не лише безпосередню близькість елементів, але й їхню співзалежність у межах усього тексту. Це робить метод особливо корисним для великих текстових масивів, де важливо не лише виділити окремі ключові елементи, але й зрозуміти їхню взаємодію в загальному контексті. Застосування асоціативних правил дозволяє створювати реферати, що містять не тільки основні думки, а й пов’язану додаткову інформацію, яка доповнює основні теми.

Важливо зазначити, що асоціативні правила можуть бути використані не лише для виявлення простих зв'язків між словами, але й для побудови складніших мереж взаємодій між темами, концепціями або навіть цілими абзацами. Це дозволяє створювати багатовимірні реферати, що відображають не тільки основну тему тексту, але й додаткові аспекти, які можуть бути важливими для глибшого розуміння матеріалу. Такий підхід до автоматизованого реферування підвищує якість аналізу, особливо у випадках, коли тексти мають складну структуру або багаторівневу концептуальну побудову. Детальний опис переваг та недоліків можна побачити у таблиці 2.4.

Таблиця 2.4 – Переваги та недоліки асоціативних правил

Аспект	Переваги	Недоліки
Принцип роботи	Виявлення прихованих взаємозв'язків між словами, фразами або концепціями на основі частоти та співзалежності.	Потреба в налаштуванні метрик підтримки (support) та довіри (confidence).

Аспект	Переваги	Недоліки
Якість узагальнення	Дозволяє створювати точні та змістовні резюме з урахуванням ключових взаємозв'язків між елементами тексту.	Може упустити контекст, якщо ключові елементи з'являються рідко або не мають чіткої співзалежності.
Гнучкість	Може враховувати залежності не тільки між словами, але й між темами, концепціями чи цілими абзацами.	Не завжди виявляє точні взаємозв'язки для текстів із низькою кореляцією між елементами.
Ефективність	Ефективний для аналізу великих текстових масивів з багаторівневими зв'язками.	Потребує значних обчислень для великих текстових масивів із багатьма взаємозалежними елементами.
Затратність	Має помірні обчислювальні вимоги, особливо у випадках із середніми за обсягом текстами.	Менш ефективний для текстів із низькою частотою повторень ключових слів або фраз.
Сфери застосування	Застосовується для виявлення ключових взаємозв'язків у текстах, створення багатовимірних рефератів.	Іноді генерує правила, які не мають практичного значення, потребують додаткової перевірки.
Можливі помилки	Дозволяє враховувати контекст і взаємодію ключових тем у всьому тексті.	Може створювати складні та важкі для інтерпретації мережі взаємозв'язків.

Висновки до розділу

Було розглянуто кілька підходів до автоматизованого реферування текстів, зокрема абстрактивні та екстрактивні методи, а також використання ланцюгів

Маркова та асоціативних правил. Абстрактивні алгоритми надають можливість створювати зв’язні тексти на основі семантичного аналізу, тоді як екстрактивні методи відзначаються швидкістю та простотою. Ланцюги Маркова і асоціативні правила є ймовірнісними підходами, які можуть ефективно доповнювати один одного.

Використання ланцюгів Маркова дозволяє будувати моделі прогнозування наступних слів або фраз на основі короткострокових залежностей між елементами тексту, що сприяє збереженню зв'язності у стислому викладі. Водночас, асоціативні правила дозволяють виявляти приховані взаємозв’язки між елементами тексту, що допомагає краще зрозуміти його глибший зміст та взаємодію між концепціями. Порівння алгоритмів можна побачити у таблиці 2.5.

Таблиця 2.5 – Порівняння існуючих алгоритмів

Метод	Переваги	Недоліки	Сфери застосування
Абстрактивні алгоритми	Створюють зв’язні та природні тексти завдяки семантичному аналізу; здатність до глибокого розуміння контексту.	Високі обчислювальні витрати; складність реалізації; можливі помилки в складних текстах.	Реферування складних текстів (наукові статті, новини, технічні документи).
Екстрактивні алгоритми	Швидкість і простота реалізації; низькі обчислювальні витрати; точність передання змісту.	Можливість втрати логічної зв’язності; обмежена здатність працювати зі	Швидке створення резюме текстів; обробка великих обсягів текстової інформації.

Метод	Переваги	Недоліки	Сфери застосування
		складними текстами.	
Ланцюги Маркова	Ефективність у прогнозуванні короткострокових залежностей; швидкість і низькі обчислювальні вимоги.	Ігнорують глобальний контекст; можуть генерувати повторення або шаблонні послідовності.	Прогнозування слів/фраз; моделювання текстів із короткостроковими зв’язками.
Асоціативні правила	Виявлення прихованих взаємозв'язків між елементами тексту; здатність до аналізу великих текстових масивів.	Потреба в налаштуванні метрик; можливі складні для інтерпретації результати.	Аналіз текстів із прихованими взаємозв’язками; створення багатовимірних рефератів.

МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

1.3 Постанова задачі реферування текстів

Задача автоматизованого реферування текстів полягає у створенні системи, яка на основі вхідного тексту генерує скорочений реферат, що містить основну інформацію оригінального тексту.

На вхід системи подається текст *T*, який представлений у вигляді впорядкованої множини речень:

$$T = \{s_1, s_2, \dots, s_n\}, \tag{3.1}$$

де s_i – окреме речення тексту;

n – загальна кількість речень.

На виході система повинна формувати реферат *R*, який є підмножиною *T*:

$$R \subseteq T, \tag{3.2}$$

та задовольняє такі умови:

– реферат R має містити меншу кількість речень, ніж оригінальний текст;

– речення реферату повинні відображати основний зміст тексту T;

– речення реферату повинні зберігати початковий порядок слідування з тексту T.

Процес формування реферату можна описати наступними кроками:

– токенізація тексту на речення;

– виявлення ключових слів та фраз;

– нормалізація тексту шляхом приведення слів до початкової форми;

– для кожного речення обчислюється його вагова оцінка $W(s_i)$, яка визначає інформативність речення;

– оцінка $W(s_i)$ (ймовірності переходів між реченнями та важливості ключових слів речення);

– речення сортуються за значенням їхньої ваги $W(s_i)$ у спадному порядку;

– вибираються k речень із найбільшими значеннями $W(s_i)$ для формування реферату R.

Математично задача автоматизованого реферування формулюється як задача вибору множини R , яка максимізує загальну інформативність, що визначається як:

$$\sum_{s_i \in R} W(s_i), \tag{3.3}$$

де $W(s_i)$ – вагова оцінка речення s_i, яка визначається формулою:

$$W(s_i) = \alpha M(s_i) + \beta A(s_i), \tag{3.4}$$

де α і β – вагові коефіцієнти, які регулюють вплив факторів;

$M(s_i)$ – оцінка зв'язності речення s_i у контексті тексту, отримана на основі матриці переходів;

$A(s_i)$ – оцінка важливості речення s_i з точки зору ключових слів.

Система повинна забезпечувати такі властивості: релевантність, когерентність, збереження пропорційності. Таким чином, задача автоматизованого реферування полягає в оптимальному виборі речень для реферату, забезпечуючи баланс між скороченням тексту та збереженням його змісту.

1.3.1 Алгоритм реферування на основі ланцюги Маркова в поєднанні з асоціативними правилами

У розроблювальній системі використовується комбінація ланцюгів Маркова та асоціативних правил. Це забезпечує високу точність і релевантність результатів

за рахунок математично обґрунтованого підходу, що враховує як ймовірнісні, так і контекстуальні аспекти аналізу тексту.

Ланцюги Маркова є статистичними моделями, що описують системи, які переходять з одного стану в інший з певними ймовірностями. У контексті обробки природної мови, станами можуть бути слова або фрази. Основне припущення марковської моделі полягає в тому, що ймовірність появи слова залежить лише від певної кількості попередніх слів. У випадку ланцюга Маркова першого порядку умовна ймовірність появи слова w_i після слова w_{i-1} визначається за формулою:

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1},w_i)}{C(w_{i-1})}, \tag{3.5}$$

де $C(w_{i-1}, w_i)$ – кількість випадків, коли слова w_{i-1} та w_i зустрічаються послідовно в тексті, $C(w_{i-1})$ – загальна кількість появ слова w_i. Формула (3.5) дозволяє оцінити ймовірність переходу від одного слова до іншого на основі емпіричних даних тексту, що забезпечує моделювання локальних залежностей між словами.

Асоціативні правила використовуються для виявлення часто зустрічаючихся поєднань слів або термінів, які мають значущий взаємозв’язок. Формально, асоціативне правило записується у вигляді:

$$X \Longrightarrow Y, \tag{3.6}$$

де X та Y – множини слів або елементів. Метою є визначення ймовірності того, що множина Y з’явиться в тексті за умови, що з’явилася множина X. Для кількісної оцінки асоціативних правил використовуються міри підтримки (support) та довіри (confidence), які обчислюються за формулами:

$$support(X \Longrightarrow Y) = \frac{C(C,Y)}{N}, \tag{3.7}$$

$$\text{confidence}(X \Longrightarrow Y) = \frac{C(C,Y)}{C(X)}, \tag{3.8}$$

де C(X,Y) – кількість транзакцій, в яких одночасно зустрічаються X та Y, C(X) - кількість транзакцій з X , N - загальна кількість транзакцій, наприклад, речень або документів. Міра підтримки показує, наскільки часто правило застосовується в тексті, тоді як міра довіри відображає ймовірність появи Y за умови появи X.

Для підвищення точності моделювання зв'язків між словами, ланцюги Маркова поєднуються з асоціативними правилами. Це дозволяє враховувати як послідовні залежності між словами, так і більш складні асоціації. Комбінована ймовірність появи слова w_i після слова w_{i-1} визначається за формулою:

$$P_{comb}(w_i|w_{i-1}) = \lambda\, P(w_i|w_{i-1}) + (1-\lambda) confidence(w_{i-1} \Longrightarrow w_i), \quad (3.9)$$

де $P(w_i|w_{i-1})$ – умовна ймовірність за марковською моделлю формула (3.5), $confidence(w_{i-1} \Longrightarrow w_i)$ – міра довіри асоціативного правила між w_i та w_{i-1}, а λ – ваговий коефіцієнт ($0 \leq \lambda \leq 1$), що визначає вплив кожного з компонентів на комбіновану ймовірність.

Процес реалізації алгоритму починається з підготовки тексту. Виконується попередня обробка, яка включає токенізацію на слова та речення, видалення стоп-слів та пунктуації, а також лематизацію або стемінг для приведення слів до базової форми. Ці дії спрямовані на зменшення розмірності даних та усунення шуму, що покращує точність подальших статистичних оцінок.

Далі будується марковська модель. На основі підготовленого тексту обчислюються частоти появи слів та їхніх послідовностей. Використовуючи формулу (3.5), визначаються умовні ймовірності для всіх пар слів у тексті.

Наступним кроком є обчислення комбінованих ймовірностей. Для кожної пари слів обчислюється комбінована ймовірність за формулою (3.9). Ваговий коефіцієнт може бути обраний експериментально для оптимізації результатів.

На завершення відбувається оцінка та відбір ключових речень. Розраховується сумарна вага кожного речення шляхом агрегування комбінованих

ймовірностей слів, що входять до нього. Речення ранжуються за спаданням ваги, і відбираються ті, що мають найвищі значення, для формування реферату.

При застосуванні алгоритму до українськомовних текстів було враховано специфіку української мови, що дозволило підвищити ефективність реферування. Зокрема, було оптимізовано процес лематизації з використанням спеціалізованих словників та морфологічних аналізаторів для української мови. Це забезпечило коректне приведення слів до їх базових форм, що є критично важливим для точності обчислення частот та ймовірностей.

Крім того, було враховано особливості української граматики, такі як багатокореневі слова та складні відмінкові форми, які впливають на послідовності слів у реченнях. Для цього марковська модель була адаптована з урахуванням морфологічних ознак слів, що дозволило покращити моделювання ймовірностей переходів між словами різних граматичних категорій.

Також було проведено налаштування порогових значень для мір підтримки та довіри в асоціативних правилах, зважаючи на статистичні характеристики українських текстів. Це включало аналіз частотності використання специфічних українських слів та фраз, що дозволило більш точно виділяти релевантні асоціативні правила.

1.3.2 Модифікація мовної моделі BERT для україномовних текстів

Алгоритм реферування на основі моделі BERT представляє сучасний підхід до автоматичного узагальнення текстів. Використовуючи глибокі нейронні мережі та трансформери для моделювання контексту слів, він досягає високої точності в розумінні семантичних зв’язків.

Модель BERT базується на архітектурі трансформера, яка використовує механізм самоуваги (self-attention) для обробки послідовностей слів. Основною ідеєю є представлення кожного слова у вигляді вектору ознак, що враховує

контекст з обох боків слова. Це досягається шляхом навчання моделі на великому корпусі текстів з використанням задачі маскування слів.

Вхідний текст перетворюється у послідовність токенів $\{w_1, w_2, \dots, w_n\}$, які далі конвертуються у векторні представлення $\{x_1, x_2, \dots, x_n\}$ за допомогою ембеддингів. Потім ці вектори подаються на вхід трансформерного енкодера, який складається з багатьох шарів самоуваги та позиційних зсувів.

Механізм самоуваги обчислюється за формулою:

$$Attection(\ Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}}), \tag{3.10}$$

де Q, K, V – матриці запитів, ключів та значень, отримані шляхом лінійних перетворень вхідних векторів; d_k – розмірність ключів.

Після проходження через всі шари енкодера, отримуємо вихідні вектори $\{h_1, h_2, \dots, h_n\}$, які містять інформацію про контекст кожного слова в реченні.

Для задачі реферування використовується підхід на основі екстрактивного або абстрактивного узагальнення. У випадку екстрактивного реферування метою є вибір найбільш інформативних речень з тексту. Це можна формалізувати як задачу класифікації речень, де кожному реченню присвоюється ймовірність бути включеним до реферату.

Нехай $S = \{s_1, s_2, \dots, s_n\}$ – множина речень в тексті. Для кожного речення обчислюється векторне представлення шляхом агрегування векторів слів, що входять до нього. Це можна здійснити, наприклад, шляхом середнього значення векторів слів:

$$s_i = \frac{1}{n_i} \sum j = 1^{n_i} h_{ij}, \tag{3.11}$$

де n_i – кількість слів у реченні s_i, h_{ij} векторне представлення слова у реченні.

Далі для кожного речення обчислюється ймовірність:

$$P(y_i = 1 \mid s_i) = \sigma = (w^T s_i + b), \quad (3.12)$$

де w вектор ваг моделі, b – зміщення, а σ – сигмоїдна функція.

Модель навчається шляхом мінімізації функції втрат, наприклад, бінарної крос-ентропії:

$$L = -\frac{1}{m}\sum_{i=1}^{m}(y_i \log P(y_i = 1|s_i) + (1 + y_i)\log\ P(y_i = 0|s_i)), \quad (3.13)$$

де y_i – мітка класу для речення: 1, якщо речення повинно бути включеним до реферату, і 0 в іншому випадку.

При застосуванні моделі BERT до українських текстів важливо використовувати попередньо навчені моделі на українському корпусі, такі як Ukrainian BERT або мультимовні моделі. Це забезпечує коректне моделювання мовних особливостей української мови.

Було враховано специфіку української морфології та синтаксису, зокрема багатослівні вирази та складні граматичні конструкції. Для цього використовувалися спеціалізовані токенізатори, що коректно обробляють українські слова та враховують особливості алфавіту.

Модель BERT може бути тонко налаштована на конкретний корпус українських текстів для покращення якості реферування. Це досягається шляхом додаткового навчання моделі на задачі реферування з використанням спеціалізованих датасетів, що містять українські тексти та відповідні реферати. З таблиці 3.1 можна побачити основні аспекти BERT.

Таблиця 3.1 – Основні аспекти BERT

Аспект	Опис
Основна концепція	BERT навчається розуміти текст через двобічний аналіз контексту, що дозволяє ефективно працювати з багатозначними словами та складними реченнями.
Архітектура	BERT базується на трансформерах і використовує лише енкодерну частину. Self-attention допомагає аналізувати відносини між словами.
Завдання навчання	Навчання ґрунтується на двох завданнях: масковане моделювання мови (MLM) для передбачення слів і передбачення наступного речення (NSP).
Адаптація до української мови	BERT адаптовано для української шляхом перенавчання на корпусах, таких як 'ГРАК', що забезпечує врахування морфологічних і синтаксичних особливостей мови.
Універсальність	Модель може бути адаптована до різних задач: аналіз текстів, створення резюме, автоматичні відповіді на запитання тощо.
Обмеження	Великі обчислювальні витрати, обмеження довжини тексту до 512 токенів, відсутність спеціалізації без перенавчання на вузьких доменах.
Застосування	Аналіз текстів, реферування, робота з юридичними, літературними чи новинними текстами, інтерактивні запити.

1.3.3 Модифікація алгоритму TextRank для укріномовних текстів

TextRank базується на концепціях, запозичених з алгоритму PageRank, який використовується для ранжування веб-сторінок у пошукових системах. Основна

ідея полягає у побудові графової моделі тексту, де вузли відповідають словам або реченням, а ребра відображають зв'язки між ними. У цьому розділі детально описано математичне забезпечення алгоритму TextRank, його застосування до задачі реферування та оптимізацію для україномовних текстів.

Графова модель тексту представлена у вигляді ненаправленого або направленого зваженого графа G = (V, E) , де V – множина вузлів, що відповідають елементам тексту – словам або реченням, а E – множина ребер, що відображають зв'язки між цими елементами. Вага ребра w_{ij}між вузлами v_i та v_j відображає ступінь взаємозв'язку між відповідними елементами.

Побудова графа для реферування здійснюється шляхом представлення кожного речення s_i як вузла графа. Взаємозв'язок між реченнями визначається на основі міри подібності між ними. Однією з поширених мір є косинусна подібність між векторними представленнями речень. Нехай $S = \{s_1, s_2, \dots, s_N\}$- множина речень тексту. Кожне речення представляється у вигляді вектора s_i у багатовимірному просторі ознак. Тоді вага ребра між реченнями s_i та s_j обчислюється за формулою:

$$w_{ij} = \cos(s_i, s_j) = \frac{s_i s_j}{\|s_i\|\|s_j\|}, \tag{3.14}$$

де $s_i s_j$ – скалярний добуток векторів, а $\|s_i\|\|s_j\|$ – норми векторів. Формула 3.14 дозволяє кількісно оцінити подібність між реченнями на основі їхніх векторних представлень.

Після побудови графа алгоритм TextRank розраховує ранг кожного вузла, що відображає його важливість у тексті. Ранг вузла визначається рекурсивно за формулою:

$$R(v_i) = (1 - d) + d \sum_{v_j \in In(v_i)} \frac{w_{ji}}{\sum_{v_k \in Out(v_j)} w_{jk}}, \tag{3.15}$$

де d – коефіцієнт затухання (звичайно приймається значення d = 0.85), $\text{In}(v_i)$ – множина вузлів, що мають ребра до вузла v_i, $\text{Out}(v_j)$ – множина вузлів, до яких ведуть ребра з вузла v_j, а w_{ji}вага ребра від вузла v_j до v_i. Формула (3.15) аналогічна до рівняння розрахунку рангу в алгоритмі PageRank і відображає ідею, що важливість вузла визначається важливістю вузлів, які на нього вказують.

Розрахунок рангу виконується ітеративно, починаючи з початкового значення $R_0(v_i) = 1$ для всіх v_i. На кожній ітерації значення рангу оновлюються за формулою (3.2) до досягнення збіжності, тобто поки зміни рангу на наступній ітерації не стануть меншими за заданий поріг.

Після розрахунку рангу для всіх речень вони сортуються за спаданням значень $R(v_i)$. Вибираються верхні K речень з найбільшим рангом для формування реферату. Цей підхід дозволяє автоматично визначати найбільш інформативні речення, які найкраще відображають зміст оригінального тексту.

Для отримання більш точних векторних представлень використовувалися ембеддінги, навчені на українських текстах. Це могли бути моделі word2vec або GloVe, спеціально навчені на українських корпусах, або контекстуальні ембеддінги, такі як FastText чи BERT, адаптовані для української мови. Використання цих моделей забезпечує більш точне відображення семантичних зв’язків між словами та реченнями.

Українська мова характеризується багатою морфологією, що може ускладнювати обчислення подібності між реченнями. Було застосовано лематизацію або стемінг для приведення слів до базових форм, що дозволило зменшити розмірність векторного простору та покращити точність обчислення косинусної подібності за формулою (3.14).

Було складено список українських стоп-слів, які не несуть змістовного навантаження, та виключено їх з аналізу. Це дозволило зменшити шум у даних та підвищити якість побудови графа, оскільки ребра між реченнями встановлюються на основі інформативних слів.

Крім косинусної подібності, було розглянуто використання інших мір подібності, зокрема, Jaccard коефіцієнта або мір на основі спільних лемат, що можуть бути більш чутливими до лексичних особливостей української мови.

1.4 Постанова задачі оцінювання рефератів

Розробка механізму автоматизованого оцінювання та ранжування рефератів текстів ґрунтується на необхідності забезпечення об’єктивної, точної та прозорої оцінки якості текстових резюме. У сучасних умовах зростання обсягу інформації виникає проблема створення ефективних засобів скорочення тексту для полегшення його сприйняття, зберігаючи при цьому основний зміст і структуру. Розв’язання цієї задачі потребує розробки алгоритмів та моделей, які здатні генерувати резюме, а також здійснювати їх об’єктивне оцінювання на основі формалізованих метрик. Основною метою є створення механізму, який забезпечить автоматичне створення рефератів із використанням сучасних методів текстового аналізу, таких як TextRank, асоціативні ланцюги Маркова та BERT, та дозволить проводити багатокритеріальне оцінювання їхньої якості. Це передбачає аналіз та оцінку різних аспектів, зокрема змістовності, компактності, читабельності, когерентності, збереження ключових концепцій та щільності інформації. Для забезпечення коректного аналізу текстів перед виконанням оцінювання необхідно проводити етап попередньої обробки, що включає токенізацію, видалення стоп-слів, стемінг та нормалізацію векторного подання. Таким чином, механізм має забезпечити виконання повного циклу операцій, починаючи від попередньої обробки тексту до формування об’єктивної оцінки його якості. Його головним призначенням є інтеграція різних підходів для створення ефективного інструменту, який стане у нагоді в багатьох сферах, таких як наукові дослідження, освіта та інформаційний менеджмент. Механізм повинен забезпечувати високу точність і об’єктивність оцінювання через використання кількісних метрик, що дозволяють

порівнювати різні алгоритми генерації текстових рефератів, підтримуючи при цьому автоматизованість процесу та гнучкість застосування.

1.4.1 Алгоритми оцінювання рефератів

Розроблена система для автоматизованого оцінювання та ранжування рефератів текстів базується на використанні багатокритеріального підходу, що дозволяє врахувати різні аспекти якості текстових резюме. Основна задача полягає не лише у створенні стислих рефератів за допомогою різних алгоритмів, таких як TextRank, асоціативні ланцюги Маркова та BERT, але й у об'єктивній оцінці їхньої якості за допомогою набору формалізованих метрик.

Для забезпечення коректного аналізу тексту перед виконанням оцінювання проводиться етап попередньої обробки. У тексті виконуються токенізація, видалення стоп-слів та стемінг. Токенізація розбиває текст на окремі елементи (слова чи речення), що полегшує подальшу обробку. Видалення стоп-слів усуває службові слова, які не несуть смислового навантаження, наприклад, «але», «вже», «тільки». Стемінг видаляє суфікси, характерні для українських слів, такі як «-ання», «-ення», «-ість», залишаючи лише основу слова. Це дозволяє системі узагальнювати різні форми одного і того ж слова, забезпечуючи кращу узгодженість у подальшій обробці.

Для проведення точного аналізу тексту його необхідно представити у вигляді числових векторів. Векторизація здійснюється шляхом створення векторного подання кожного тексту на основі частоти появи слів у ньому. Нехай T — текст, що складається з слів $w_1, w_2, \dots w_n$. Терм-фреквенція (TF) кожного слова обчислюється за формулою:

$$TF(w_i) = \frac{f_{w_i}}{\sum_{j=1}^{n} f_{w_j}}, \tag{3.16}$$

де f_{w_j} – кількість появ слова w_i у тексті T. Для врахування значущості слів використовується зважена терм-фреквенція – обернена документна частота.

Вектор нормалізується, що забезпечує однаковий масштаб для порівняння між різними текстами. Нормалізація вектора виконується за допомогою L2-норми за формулою:

$$v_{norm} = \frac{v}{\|v\|_2}, \tag{3.17}$$

Цей етап є критично важливим для метрик, які використовують подібність між текстами, наприклад, косинусну схожість.

Кожен згенерований реферат оцінюється за шістьма основними метриками, які враховують різні аспекти якості тексту.

Метрики оцінювання якості рефератів включають кілька ключових аспектів, які дозволяють об’єктивно аналізувати, наскільки добре резюме відображає зміст оригінального тексту.

Першою є покриття змісту (content coverage), яке вимірює, наскільки ефективно реферат охоплює основні ідеї тексту. Для цього використовується косинусна схожість між векторами оригінального тексту v_{orig} та реферату v_{ref}, яка обчислюється за формулою:

$$ContentCoverage = \cos\left(v_{orig} v_{ref}\right) = \frac{v_{orig} v_{ref}}{||v_{orig}||\,||v_{ref}||}, \tag{3.18}$$

Високе значення цієї метрики свідчить про те, що резюме передає основний зміст тексту без втрати важливих деталей.

Щільність інформації (information density), яка є другою метрикою, визначається як співвідношення унікальних слів у рефераті до загальної кількості слів, кориговане на основі покриття змісту. Формула для розрахунку щільності інформації:

$$InformationDestiny = \frac{|\text{Унікальні слова в рефераті}|}{|\text{Загальна кількість слів у рефераті}|} ContentCoverage, \quad (3.19)$$

Ця метрика показує, наскільки компактно та ефективно передано інформацію, уникаючи зайвих повторень і наповнюючи резюме лише суттєвими деталями.

Третя метрика, ключові концепції (key concepts), аналізує, чи зберігаються у рефераті найбільш важливі ідеї та терміни, присутні в оригінальному тексті. Нехай K_{orig} – множина ключових концепцій оригінального тексту, а K_{ref} – множина ключових концепцій реферату. Тоді метрика обчислюється за формулою:

$$KeyConcepts = \frac{|K_{ref} \cap K_{orig}|}{|K_{orig}|}, \quad (3.20)$$

Високий показник за цією метрикою означає, що резюме ефективно передає основний зміст, зберігаючи ключові ідеї.

Наступна метрика, читабельність (readability), оцінює легкість сприйняття тексту. Вона базується на середній довжині речень L_{sent} та середній довжині слів L_{word}. Читабельність може бути оцінена за формулою:

$$Readability = \alpha \left(1 - \frac{L_{sent}}{L_{optimal}}\right) + \beta(1 - \frac{L_{word}}{L_{max}}), \quad (3.21)$$

де $L_{optimal}$ – оптимальна середня довжина речення (наприклад, 30 слів), L_{max} – максимальна допустима довжина слова (наприклад, 12 символів), α та β – вагові коефіцієнти, що визначають внесок кожного компонента у загальну оцінку читабельності.

П’ята метрика, когерентність (coherence), оцінює зв’язність тексту шляхом аналізу схожості між сусідніми реченнями. Для цього використовується середня

косинусна схожість між векторами речень s_i та $s_i + 1$, що обчислюється за формулою:

$$Coherence = \frac{1}{N-1}\sum_{i=1}^{N-1} \cos(s_i, s_i + 1), \quad (3.22)$$

де N – кількість речень у рефераті.

Кожна метрика має ваговий коефіцієнт, який визначає її значущість у загальній оцінці. Наприклад, покриття змісту має найбільшу вагу (0.25), тоді як когерентність та баланс розділів оцінюються із меншою значущістю (0.1). Загальний бал для реферату розраховується як зважена сума значень усіх метрик.

1.5 Постанова задачі перекладу реферату

Задача автоматичного перекладу рефератів полягає у створенні механізму, який дозволяє точно та ефективно перетворювати текст резюме з однієї мови на іншу, зберігаючи при цьому його змістовність, лексичну насиченість, граматичну правильність та стилістичну відповідність. Основною метою є розробка алгоритму, який забезпечить переклад тексту реферату таким чином, щоб цільовий текст залишався максимально наближеним до оригіналу як за змістом, так і за структурою.

Алгоритм має враховувати особливості реферативного стилю, який характеризується високим рівнем узагальнення, компактністю та відсутністю зайвих повторень. Переклад рефератів ускладнюється необхідністю збереження ключових концепцій і термінів, адекватного відображення міжмовних еквівалентів та забезпечення когерентності тексту. Завдання ускладнюється, коли мова йде про мови з багатою граматичною структурою, наприклад, українську, або мови з різними культурними та мовними традиціями.

1.5.1 Алгоритм перекладу реферату

Алгоритм автоматичного перекладу текстів, реалізований за допомогою моделі MarianMT, є сучасним рішенням у галузі машинного перекладу, яке базується на трансформерних архітектурах. Модель, частина бібліотеки Hugging Face Transformers [16], дозволяє виконувати якісний переклад між великою кількістю мовних пар. Завдяки своїй універсальності та адаптивності цей алгоритм забезпечує точність і ефективність перекладу навіть для мов із багатою граматичною структурою, таких як українська.

Основна ідея алгоритму полягає у використанні нейронних трансформерних моделей для аналізу тексту та його перетворення в іншомовний еквівалент. Робота алгоритму розпочинається з вибору мовної пари, де користувач задає мову джерела та цільову мову. На основі цього обирається відповідна модель перекладу із колекції Helsinki-NLP, наприклад, модель для перекладу з української на англійську має назву Helsinki-NLP/opus-mt-uk-en. Після цього завантажується токенізатор, який розбиває текст на токени $T = \{t_1, t_2. \ldots, t_n\}$ та перетворює їх у числові вектори за допомогою ембеддінг-функції:

$$e_i = E(t_i), \tag{3.22}$$

де E – матриця ембеддінгів, а e_i - векторне представлення токена t_i. Це забезпечує правильне форматування тексту для обробки. Паралельно завантажується модель MarianMT, яка використовується для генерації перекладеного тексту на основі токенізованого вводу.

Вхідний текст проходить попередню обробку, яка включає токенізацію, доповнення тексту для уніфікації його довжини, а також усічення тексту, якщо його розмір перевищує допустимі межі моделі. Цей підготовлений текст передається до моделі MarianMT, яка за допомогою механізму self-attention прогнозує

послідовність токенів у цільовій мові. Формально, механізм self-attention визначається як:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \quad (3.23)$$

де Q – матриця запитів, K – матриця ключів, V – матриця значень, а d_k – розмірність ключів. Завершальним етапом є декодування отриманих токенів у текстовий формат, що забезпечується токенізатором. При цьому згенерований текст очищується від спеціальних символів, таких як <s> або </s>, які додаються для технічних потреб під час генерації.

Особливістю алгоритму є його адаптація для української мови, що включає підтримку специфічних моделей, таких як Helsinki-NLP/opus-mt-uk-en та Helsinki-NLP/opus-mt-en-uk. Ці моделі враховують лексичні, морфологічні та синтаксичні особливості української мови. MarianMT здатний ефективно працювати з контекстами будь-якої складності, що є критично важливим для мов із вільним порядком слів, таких як українська [16]. Завдяки механізму self-attention модель аналізує взаємозв'язки між словами в реченні, навіть якщо порядок слів змінюється. Використання спеціального токенізатора, адаптованого до української мови, забезпечує коректну обробку відмінків, чисел і граматичних форм. Крім того, алгоритм підтримує обробку довгих текстів через автоматичне усічення, що дозволяє уникнути обмежень на довжину вхідних даних.

MarianMT вирізняється високою універсальністю, оскільки підтримує широкий спектр мовних пар, а його трансформерна архітектура дозволяє враховувати як локальний, так і глобальний контекст тексту. Завдяки цьому модель забезпечує високоякісний переклад, який є релевантним до особливостей обох мов. Для української мови це особливо важливо, адже зберігається точність перекладу навіть за наявності складних граматичних конструкцій чи багатозначних слів. Хоча модель має певні обмеження, такі як обмеження на кількість токенів у тексті та

потреба у значних обчислювальних ресурсах для роботи з довгими текстами, її ефективність у вирішенні завдань міжмовної комунікації є незаперечною.

Висновки до розділу

Проаналізувавши існуючі системи автоматичного реферування текстів, можна зробити висновок, що їхня адаптованість до української мови є вкрай обмеженою. Більшість алгоритмів і моделей орієнтовані на англомовний контент, що ускладнює їхнє застосування до україномовних текстів через різницю у граматиці, морфології, синтаксисі, а також специфіку наукової термінології. Це призводить до зниження ефективності реферування українських текстів, що особливо актуально для наукових статей, де якість аналізу є критично важливою. Розроблена система спрямована на подолання цих викликів, інтегруючи три підходи: ланцюги Маркова з асоціативними правилами, BERT та TextRank. Кожен із цих методів має свою специфіку і забезпечує структурний або семантичний аналіз тексту, що дозволяє підвищити точність і гнучкість реферування.

Однією з головних переваг системи є можливість налаштування параметрів реферування, таких як бажана кількість слів у рефераті, що дозволяє адаптувати результати до потреб користувача. Крім того, впроваджені метрики оцінювання текстів, які враховують релевантність, повноту та точність, дають змогу автоматично оцінювати якість згенерованих рефератів та оптимізувати їхній вміст. Це забезпечує користувачам високий рівень індивідуалізації і зручності під час роботи із системою.

Комбінація ланцюгів Маркова та асоціативних правил є основою для моделювання послідовностей речень та аналізу ключових фраз. Ланцюги Маркова використовують ймовірнісні матриці переходів між реченнями, дозволяючи визначати стійкі стани, які репрезентують важливі сегменти тексту. Асоціативні правила допомагають виявити часто повторювані шаблони фраз і зв’язків між словами, що додає семантичної глибини до реферування. У результаті система

створює реферат, що враховує як ймовірнісну структуру тексту, так і семантичні особливості української мови.

Інтеграція BERT дозволяє системі ефективно працювати із семантичним аналізом тексту завдяки двобічному аналізу контексту. Модель BERT, перенавчена на україномовних корпусах, таких як “ГРАК”, демонструє високу ефективність у врахуванні морфологічних і синтаксичних особливостей української мови. Вона здатна правильно інтерпретувати багатозначні слова, складні граматичні конструкції та змінний порядок слів, що є важливим для роботи з україномовними текстами. Незважаючи на високі обчислювальні витрати, BERT забезпечує точне та релевантне реферування, яке відповідає специфіці наукового контексту.

TextRank, адаптований до української мови, додає можливості графового аналізу для виділення ключових речень та слів у тексті. Алгоритм враховує семантичну та граматичну подібність між реченнями, доповнену спеціальними модифікаціями для обробки морфології та варіативного порядку слів в українській мові. Використання косинусної схожості між векторними поданнями речень дозволяє системі генерувати логічно узгоджені реферати, які точно передають основний зміст тексту.

Для оцінювання якості рефератів було впроваджено багатокритеріальну систему метрик, яка включає покриття змісту, щільність інформації, збереження ключових концепцій, читабельність, когерентність та баланс розділів. Ці метрики дають змогу комплексно оцінювати реферати, враховуючи як точність, так і структурну якість тексту. Кожна метрика має ваговий коефіцієнт, що забезпечує її вплив на загальну оцінку, дозволяючи автоматично ранжувати згенеровані резюме.

Особливістю системи є також алгоритм автоматичного перекладу текстів на англійську мову за допомогою MarianMT. Це розширює функціонал системи, дозволяючи працювати з багатомовними текстами. Модель MarianMT забезпечує точний переклад українських текстів, враховуючи багатослівність, складну граматику та специфічну лексику, що є характерними для української мови.

У підсумку, розроблена система поєднує кілька методів і технологій, які забезпечують високу якість реферування українських текстів. Вона враховує специфіку мови, адаптована для роботи з різними типами контенту, включаючи наукові статті, та пропонує користувачам широкий функціонал. Унікальність системи полягає в її здатності ефективно працювати з україномовними текстами, що робить її цінним інструментом для автоматизації роботи з текстами у науковій та освітній сферах.

ВИБІР ТА ОБГРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ

1.6 Клієнтська частина

Клієнтська частина системи є важливим компонентом, який відповідає за взаємодію користувача із застосунком. Вона забезпечує зручний і зрозумілий інтерфейс, через який користувачі можуть вводити текст, завантажувати файли для аналізу та отримувати результати у вигляді згенерованих підсумків і метрик якості. Основне завдання клієнтської частини — спростити процес взаємодії між користувачем і сервером, забезпечуючи швидку передачу даних, адаптивний дизайн та інтуїтивний досвід.

1.6.1 Опис JavaScript-бібліотеки React.js

React.js – це потужна JavaScript-бібліотека, створена компанією Facebook у 2013 році, яка широко використовується для розробки сучасних веб-додатків. Вона орієнтована на створення інтерактивних користувацьких інтерфейсів (UI) та дозволяє будувати складні веб-системи завдяки своїй компонентній архітектурі. Основною перевагою React є можливість розділення інтерфейсу на незалежні, багаторазово використовувані компоненти, що значно спрощує підтримку та масштабування додатків [17].

React.js працює за принципом односпрямованого потоку даних, що означає, що дані в додатку передаються від батьківських компонентів до дочірніх через так звані “пропси” (props). Це гарантує передбачуваність поведінки додатка, оскільки зміни в даних автоматично відображаються на інтерфейсі. Додатково, React використовує технологію Virtual DOM (віртуального дерева документів), що дозволяє ефективно оновлювати тільки ті елементи інтерфейсу, які зазнали змін. Це значно підвищує продуктивність, особливо у випадках роботи з великими обсягами динамічних даних.

Ключовою особливістю React.js є його декларативний підхід. Розробники описують, яким має бути стан інтерфейсу, а React автоматично забезпечує відображення цього стану відповідно до змін у даних. Такий підхід спрощує процес розробки, зменшує ймовірність помилок та підвищує читабельність коду [17].

У цьому проєкті React.js було обрано для реалізації клієнтської частини з кількох причин. Перш за все, React надає чудові можливості для роботи з динамічними інтерфейсами. Система передбачає введення тексту, завантаження файлів, обробку отриманих результатів і відображення метрик якості аналізу, тому необхідно було забезпечити ефективну взаємодію користувача із застосунком. Використання компонентної архітектури React дозволило розділити інтерфейс на незалежні модулі, такі як форми введення, вкладки для перемикання між режимами роботи (текст/файл) та блоки відображення результатів. Це спрощує розробку, тестування та подальшу підтримку коду.

Ще однією важливою перевагою React є його інтеграція з іншими бібліотеками. У даному проекті для стилізації інтерфейсу було використано Tailwind CSS, що дозволяє швидко створювати адаптивний дизайн для різних пристроїв. Крім того, інтеграція з Formik забезпечила зручність роботи з формами, включаючи валідацію введених даних. Також була реалізована інтеграція з Radix UI, яка дозволила створити інтуїтивно зрозумілі вкладки для перемикання між режимами роботи системи.

React.js також забезпечує високу продуктивність роботи інтерфейсу. Завдяки використанню Virtual DOM оновлення елементів інтерфейсу відбувається миттєво, навіть при обробці складних запитів або великого обсягу даних. Це особливо важливо у випадках, коли користувач завантажує великий текстовий файл, оскільки завдяки оптимізованій роботі з DOM [18] інтерфейс залишається швидким і чутливим.

Окрім цього, React.js має велику та активну спільноту розробників, що гарантує доступність документації, навчальних матеріалів та готових рішень.

Завдяки цьому спрощується інтеграція нових функціональностей у систему та вирішення можливих технічних проблем.

Обґрунтовуючи вибір React.js у даному проекті, слід зазначити, що ця бібліотека забезпечує гнучкість, масштабованість і зручність у використанні. Вона дозволяє легко додавати нові функції без значних змін у базовому коді. У цьому проекті React виконує роль основи для побудови інтуїтивного інтерфейсу, який відповідає сучасним вимогам до веб-додатків. Користувач отримує можливість легко взаємодіяти з системою, вводити дані, переглядати результати аналізу текстів та оцінювати їх якість.

1.6.2 Опим JavaScript-фреймворку Remix

Remix – це сучасний JavaScript-фреймворк, створений для розробки швидких, масштабованих і SEO-оптимізованих веб-додатків. Він тісно інтегрується з React.js, забезпечуючи ефективну маршрутизацію, серверний рендеринг та гнучкість в обробці даних. Remix дозволяє будувати веб-застосунки, які можуть виконувати код як на сервері, так і на клієнті, забезпечуючи оптимальне використання ресурсів та швидке завантаження сторінок [19]. Його основною перевагою є декларативна маршрутизація, де структура додатка організовується через файли маршруту, що спрощує підтримку та масштабування коду.

Особливостю Remix є серверний рендеринг, який дозволяє формувати HTML-контент на сервері перед його передачею до клієнта [20]. Це значно знижує час завантаження сторінок, особливо для користувачів із повільним інтернетом, і забезпечує готовність контенту для індексації пошуковими системами. Крім того, фреймворк підтримує ефективне управління станом через маршрути, дозволяючи передавати дані між клієнтом і сервером без необхідності використання сторонніх бібліотек, таких як Redux. Це зменшує складність проєкту і підвищує продуктивність за рахунок мінімізації зайвих HTTP-запитів.

У проєкті Remix використовується для реалізації клієнтської частини системи. Основними завданнями є організація маршрутизації, інтеграція з серверною частиною для отримання результатів аналізу текстів та відображення цих результатів у зрозумілому для користувача форматі. Наприклад, за допомогою Remix було реалізовано маршрути для стартової сторінки, форми введення тексту, завантаження файлів та перегляду результатів аналізу. Така структура дозволяє розділити функціонал системи на логічні блоки, що спрощує підтримку і подальший розвиток додатка.

Ще однією важливою перевагою Remix є його інтеграція з React.js, що дозволяє використовувати всі переваги цієї бібліотеки для побудови компонентів інтерфейсу. Завдяки цьому в проєкті реалізовано зручні та інтерактивні елементи, такі як вкладки для перемикання між режимами роботи, форми введення тексту з валідацією, а також блоки для відображення метрик якості аналізу. Крім того, Remix легко інтегрується з бібліотеками стилів, такими як Tailwind CSS, що дозволяє створювати адаптивний дизайн, оптимізований для різних пристроїв.

На додачу перевагою використання Remix є його здатність забезпечувати швидке завантаження сторінок і зручність взаємодії користувача із системою. Серверний рендеринг дозволяє зменшити час очікування для користувача навіть при обробці великих текстових файлів, а можливість попередньої обробки даних на сервері забезпечує стабільність роботи додатка. Додатково, Remix автоматично формує чистий HTML, що позитивно впливає на SEO-оптимізацію, дозволяючи результатам аналізу тексту бути доступними для пошукових систем.

1.6.3 UI бібліотека Formik

Однією з ключових задач у будь-якому веб-додатку є зручна та надійна робота з формами. У цьому проєкті для вирішення цього завдання була використана бібліотека Formik. Вона надає інтуїтивний API, який дозволяє ефективно управляти станом форм, відстежувати введені користувачем дані та

забезпечувати їх валідацію. Formik підтримує як просту валідацію за допомогою JavaScript, так і інтеграцію зі сторонніми бібліотеками, такими як Yup, що дозволяє реалізувати складні перевірки даних [21].

У цьому проєкті Formik було використано для роботи з формами введення тексту, завантаження файлів і встановлення ліміту слів для аналізу. Завдяки Formik кожна форма автоматично зберігає свій стан, перевіряє обов'язковість полів та інтерактивно відображає помилки введення. Ця бібліотека також спростила інтеграцію з кастомними стилями, створеними за допомогою Tailwind CSS. Таким чином, Formik дозволив швидко та ефективно реалізувати важливу частину функціоналу клієнтської частини системи.

1.6.4 CSS-фреймворк Tailwind CSS

Для забезпечення сучасного вигляду інтерфейсу та його адаптивності було використано Tailwind CSS. Цей утилітарний CSS-фреймворк дозволяє розробляти стилі швидко й ефективно, не створюючи окремі стилі вручну. Tailwind CSS надає великий набір класів для дизайну елементів, таких як кнопки, форми, вкладки тощо. Завдяки цьому стилі додатку можна змінювати або розширювати без написання додаткових CSS-файлів [21].

У розроблювальній системі Tailwind CSS використовується для створення адаптивного дизайну, який коректно відображається на різних пристроях. Він забезпечив єдиний стиль для таких елементів, як форми введення тексту, кнопки, вкладки та метрики. Його гнучкість дозволила швидко змінювати зовнішній вигляд елементів, спрощуючи підтримку та розширення інтерфейсу. Tailwind CSS значно скоротив час розробки, зробивши інтерфейс привабливим та зручним для користувача.

1.6.5 Набір React-компонентів Radix UI

Щоб забезпечити сучасний і доступний користувацький досвід, у проєкті було використано Radix UI – набір React-компонентів для створення інтерактивних елементів [23]. Radix UI відзначається високою доступністю та підтримкою стандартів, що робить його ідеальним вибором для додатків, які потребують інтерактивних інтерфейсів. Radix UI було застосовано для створення вкладок, які дозволяють перемикатися між режимами введення тексту та завантаження файлів.

1.6.6 Бібліотека для завантаження файлів Mammoth.js

Mammoth.js – це спеціалізована бібліотека, яка забезпечує ефективну екстракцію тексту з документів формату .doc та .docx [24]. Її головна перевага полягає у збереженні чистоти тексту: бібліотека мінімізує вплив форматування документа, вилучаючи лише текстовий вміст без зайвих елементів, таких як стилі, таблиці чи зображення. Це робить Mammoth.js оптимальним вибором для систем, що працюють з аналізом тексту, де важливо отримати максимально “сирий” контент.

Mammoth.js використовується для завантаження текстових документів, які вводить користувач. Після завантаження файлу бібліотека екстрагує текстовий вміст і інтегрує його у форму введення, готуючи дані до подальшого аналізу. Це рішення значно підвищує зручність для користувача, оскільки він може завантажувати текст прямо з документів, не конвертуючи їх у інші формати.

Бібліотека працює швидко і забезпечує надійний результат навіть для великих документів. Вона також є легковаговим інструментом, який легко інтегрується з іншими технологіями клієнтської частини, такими як React.js і Formik. У цьому проєкті використання Mammoth.js дозволило розширити функціональність системи, забезпечивши підтримку завантаження файлів та зручність для кінцевого користувача.

1.7 Серверна частина

Серверна частина виступає центральним елементом розроблювальної системи, виконуючи ключові завдання аналізу текстів та генерації підсумків. Вона обробляє запити, отримані від клієнтської частини, використовуючи сучасні алгоритми обробки природної мови, та забезпечує збереження даних у базі MongoDB. Завдяки використанню Flask серверна частина побудована як RESTful API, що забезпечує ефективну взаємодію з клієнтом і можливість масштабування. Основні функції включають генерацію підсумків із застосуванням алгоритмів TextRank, ланцюги Маркова і BERT, а також обчислення метрик якості текстів. Інтеграція з бібліотеками для роботи з текстами та високопродуктивна архітектура роблять серверну частину гнучкою, надійною і здатною задовольнити сучасні вимоги до систем обробки текстів.

1.7.1 Фреймворк Flask

Flask – це мікрофреймворк для розробки веб-додатків, створений на мові програмування Python [25]. Завдяки своїй легковазі, модульності та гнучкості, Flask широко використовується для створення RESTful API, веб-сервісів та повноцінних серверних додатків. Основною ідеєю Flask є мінімалізм і можливість адаптації до потреб конкретного проєкту: розробники отримують базовий набір інструментів, який можна розширити за допомогою численних додаткових модулів і бібліотек.

Ключовою особливостю Flask є його принцип “без примушення” (convention over configuration), що означає, що розробники мають свободу організації структури проєкту відповідно до своїх потреб. Flask не вимагає використання жорстко визначених правил, як це часто буває у великих фреймворках, таких як Django, але при цьому надає всі необхідні засоби для створення продуктивного і масштабованого серверного додатка.

У цьому проєкті Flask використовується як основа серверної частини для реалізації API, що забезпечує обробку текстів, генерацію підсумків та взаємодію з базою даних MongoDB. Flask забезпечує ефективну маршрутизацію запитів, обробку даних та інтеграцію з алгоритмами обробки тексту.

Flask використовує просту і зрозумілу систему маршрутизації, яка дозволяє визначати, які функції повинні виконуватись при зверненні до певних URL-адрес. У цьому проєкті маршрут /summarize обробляє запити для аналізу текстів. За допомогою методу @app.route визначаються HTTP-методи (наприклад, GET, POST), які підтримує маршрут, і логіка обробки запитів.

У цьому проєкті запити містять текст для аналізу, ліміт слів та інші параметри, які обробляються сервером. Відповідь, сформована у форматі JSON, містить результати аналізу, такі як підсумки тексту та метрики якості.

Flask підтримує велику кількість розширень для розширення його функціональності. У цьому проєкті було використано Flask-CORS для налаштування політик кросдоменної взаємодії, що дозволяє клієнтській частині, розміщеній на іншому домені, коректно взаємодіяти із сервером. Крім того, інтеграція з базою даних MongoDB реалізована через бібліотеку PyMongo.

Хоча мікрофреймворк підтримує рендеринг HTML-шаблонів через Jinja2, у цьому проєкті використовується лише JSON-формат відповідей, оскільки серверна частина виконує роль API для обробки даних.

Серверна частина побудована на основі Flask для обробки текстів, отриманих від клієнтської частини. Запити на маршрут /summarize приймають текст і параметри аналізу, передають їх до алгоритмів, таких як TextRank, ланцюги Маркова і BERT, а результати повертаються у вигляді JSON-відповіді. Flask також забезпечує інтеграцію з MongoDB через спеціальний клас MongoConnector, який відповідає за зберігання текстів і підсумків у базі даних. Окрім цього, Flask-CORS дозволяє клієнтському додатку, розміщеному на іншому домені, безпечно взаємодіяти з сервером.

1.7.2 NoSQL база даних MongoDB

MongoDB – це нереляційна база даних типу NoSQL, яка спеціалізується на зберіганні та управлінні великими обсягами даних у вигляді документів. Основною перевагою MongoDB є її гнучка структура даних [16]: замість традиційних таблиць вона використовує документи у форматі BSON (Binary JSON), які можуть містити складні вкладені структури. Це дозволяє легко адаптувати базу даних до змін у структурі даних без необхідності вносити значні зміни в існуючі записи. У цьому проєкті MongoDB використовується для зберігання текстів, підсумків, а також метаданих, пов’язаних із процесами аналізу текстів.

MongoDB забезпечує високу продуктивність і горизонтальне масштабування, що є критично важливим для систем, які працюють з великими обсягами даних або потребують швидкої обробки запитів. Завдяки можливості зберігати складні структури даних у вигляді одного документа, MongoDB забезпечує швидкий доступ до інформації, оскільки немає потреби у складних операціях об’єднання, як це відбувається у реляційних базах даних.

База даних використовується через бібліотеку PyMongo, яка забезпечує простий інтерфейс для роботи з базою даних у Python. Для організації взаємодії з MongoDB було створено спеціальний клас MongoConnector, який відповідає за всі основні операції, такі як створення записів, зчитування даних, оновлення і видалення. Це дозволяє абстрагувати взаємодію з базою даних, зберігаючи основну логіку програми чистою та зрозумілою.

У цьому проєкті MongoDB використовується через бібліотеку PyMongo, яка забезпечує простий інтерфейс для роботи з базою даних у Python. Для організації взаємодії з MongoDB було створено спеціальний клас MongoConnector, який відповідає за всі основні операції, такі як створення записів, зчитування даних, оновлення і видалення. Це дозволяє абстрагувати взаємодію з базою даних, зберігаючи основну логіку програми чистою та зрозумілою.

Для кожного тексту, завантаженого в систему, створюється документ, що містить наступні ключові поля:

– title: заголовок тексту;

– text: текстовий вміст, очищений та підготовлений до аналізу;

– limit: максимальна кількість слів у підсумку;

– author: ім’я користувача або ідентифікатор, який завантажив текст;

– created_at: дата і час створення запису.

Крім текстів, у базі даних зберігаються підсумки, які генеруються різними алгоритмами. Кожен підсумок містить інформацію про використовуваний алгоритм, зв’язок із текстом, для якого його було створено, та час створення. Це дозволяє організувати історію аналізу та забезпечити можливість повторного використання результатів.

MongoDB також використовується для зберігання користувацьких даних. У проєкті реалізовано структуру, яка дозволяє зберігати інформацію про користувачів, наприклад, їхні облікові записи, історію роботи з текстами та пов’язані метадані. Завдяки підтримці гнучких структур MongoDB може зберігати дані різного типу, не вимагаючи суворого дотримання схеми.

Ще однією важливою перевагою MongoDB є її масштабованість. У системах із зростаючим обсягом даних база може бути розподілена між кількома серверами, забезпечуючи високу доступність і зниження навантаження на кожен окремий вузол. Це дозволяє БД обробляти значні обсяги даних навіть у великих проєктах.

У контексті проєкту базаданих також забезпечує швидкий доступ до результатів аналізу завдяки індексації даних. Наприклад, кожен документ у колекції текстів і підсумків має унікальний ідентифікатор, який використовується для швидкого пошуку відповідної інформації. Це значно скорочує час відповіді системи навіть при роботі з великою кількістю записів.

Висновки до розділу

У цьому розділі було проведено аналіз ключових компонентів клієнтської та серверної частин системи, а також описано їхню архітектуру, функціональні можливості та вибрані технологічні рішення. У результаті виконаної роботи я дійшов висновку, що використання React.js як основної технології для розробки інтерфейсу забезпечило модульність, масштабованість і простоту реалізації. Компонентна архітектура React дала змогу розділити інтерфейс на логічні частини, такі як форми введення, вкладки та блоки відображення результатів, що значно спростило підтримку коду та його подальший розвиток. Інтеграція з бібліотеками Tailwind CSS, Formik і Radix UI дозволила створити сучасний, адаптивний і зручний інтерфейс, який відповідає сучасним вимогам користувачів. Крім того, використання Mammoth.js для екстракції тексту з документів спростило процес роботи із системою, надавши користувачам можливість завантажувати файли у форматах .doc і .docx без додаткових конвертацій.

Розробка клієнтської частини на основі Remix забезпечила ефективну маршрутизацію, що дозволило чітко розподілити функціонал системи на окремі логічні блоки, такі як стартова сторінка, введення тексту, завантаження файлів і перегляд результатів. Інтеграція Remix із серверною частиною забезпечила швидке завантаження сторінок, оптимізацію використання ресурсів та підвищення продуктивності за рахунок серверного рендерингу.

Серверна частина системи, побудована на Flask, виявилася ефективним рішенням для реалізації RESTful API, що дозволило забезпечити злагоджену взаємодію з клієнтською частиною. Використання MongoDB як бази даних гарантує надійне зберігання текстів, підсумків і метаданих завдяки її гнучкій структурі. MongoDB також забезпечує можливість масштабування при зростанні обсягів даних, що критично важливо для подібних систем. Алгоритми TextRank, ланцюги Маркова і BERT, інтегровані в серверну частину, продемонстрували

високу ефективність у генерації підсумків текстів, що дозволяє користувачам отримувати якісні результати для подальшого аналізу.

Розглянуті технології забезпечили високу продуктивність і зручність системи. Завдяки технології Virtual DOM у React.js і серверному рендерингу в Remix додаток демонструє швидкість роботи навіть при великих обсягах даних. Інтеграція бібліотек Formik і Tailwind CSS оптимізувала процес роботи з формами та створення адаптивного дизайну, що поліпшило користувацький досвід. Крім того, побудована система має високу інтеграцію між клієнтською та серверною частинами, що забезпечує її стабільну роботу. Реалізована архітектура є гнучкою, що дозволяє легко додавати новий функціонал у майбутньому, включаючи підтримку інших форматів файлів, нових алгоритмів аналізу текстів та удосконалення інтерфейсу.

СТРУКТУРНА СХЕМА

Адаптація системи автоматичного реферування текстів для української мови та наукових текстів була реалізована з урахуванням багатих лінгвістичних, граматичних і стилістичних особливостей української мови, а також структурних характеристик, притаманних науковим текстам. Основною метою цієї адаптації було створення ефективної системи, яка забезпечує точне узагальнення змісту текстів із врахуванням їхньої термінологічної та стилістичної специфіки. Робота детально представленій у схемі Ж.

На початковому етапі функціонує модуль введення тексту, який забезпечує користувачеві можливість завантажити вхідний текст для подальшої обробки. Це може бути наукова стаття, реферат, чи будь-який інший текст українською мовою. Далі текст передається в модуль підготовки тексту, який виконує його попередню обробку. Цей модуль токенізує текст, тобто розбиває його на окремі слова чи фрази, а також проводить очищення тексту від зайвих символів і нормалізацію для приведення його до уніфікованого формату. На малюнку 5.1 можна побачити візуалізацію цих модулей. Завдяки цьому текст стає готовим для подальшої обробки алгоритмами.

Рис 5.1 – Модулі вводу тексту та підготовки тексту

Для реферування тексту система включає декілька спеціалізованих модулів, кожен із яких реалізує певний підхід. Модуль реферування за допомогою алгоритму TextRank використовує графову модель для визначення ключових речень і зв’язків між ними. Цей алгоритм був модифікований для роботи з українською мовою, враховуючи її синтаксичні та морфологічні особливості, такі як відмінки та варіативний порядок слів. Інший модуль, що базується на моделі BERT, застосовує потужний трансформерний підхід для аналізу контексту тексту. Для цього було адаптовано спеціальні мовні моделі, треновані на українських текстах, включаючи наукові корпуси, що дозволило врахувати як локальні, так і глобальні залежності в тексті. Моделі на основі ланцюгів Маркова були налаштовані для аналізу ймовірностей переходу між словами та фразами, що дозволяє ефективно узагальнювати текст із багатою граматичною структурою, дивитися на малюнку 5.2. Також були розроблені моделі на основі асоціативних правил, які виявляють часті шаблони та залежності, характерні для української наукової мови.

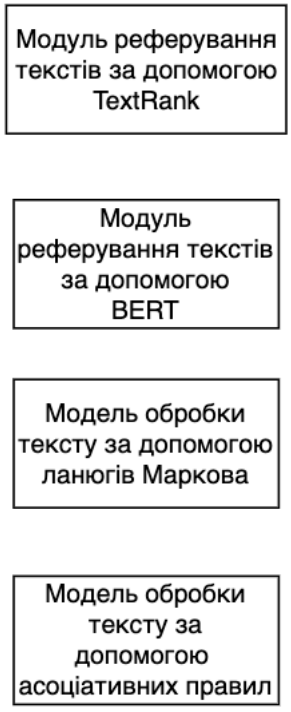

Рис 5.3 – Модулі реферування тексту

Ключовою особливістю є інтеграція моделей на основі ланцюгів Маркова та асоціативних правил (рисунок 5.4). Ця інтеграція забезпечує врахування як статистичних, так і семантичних залежностей між словами, що дозволяє створювати реферати з високим рівнем точності.

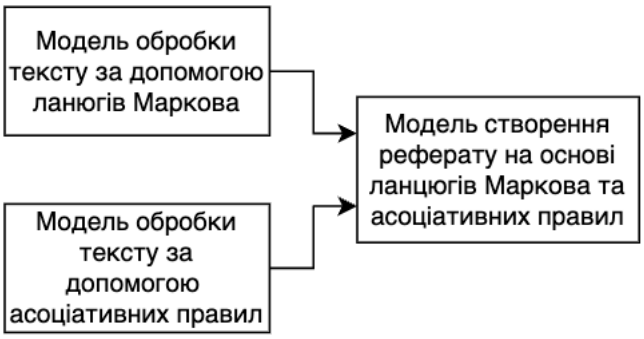

Рис 5.5 – Інтеграція алгоритмів реферування

Для вибору найкращого реферату серед створених різними модулями використовується модуль оцінювання тексту. Цей модуль порівнює результати за допомогою метрик точності, повноти та відповідності ключовим тезам оригінального тексту. Такий підхід дозволяє не тільки оцінити якість кожного реферату, а й обрати найбільш релевантний для конкретного тексту.

Після створення та оцінки реферату система забезпечує можливість його перекладу за допомогою модуля перекладу рефератів дивитися рисунок 5.6. Цей модуль використовує моделі машинного перекладу, такі як MarianMT та Helsinki-NLP, спеціально адаптовані для роботи з українською мовою. Переклад виконується зі збереженням контексту, термінології та стилістичних особливостей тексту, що особливо важливо для міжнародного обміну науковою інформацією.

Рис 5.6 – Пост обробка реферованого тексту

Усі результати роботи системи, включаючи вихідний текст, створені реферати та їх переклади, зберігаються у базі даних. Ця база даних дозволяє зберігати результати для подальшого використання, аналізу або перегляду. Користувач має змогу взаємодіяти із системою через модуль відображення рефератів, який забезпечує зручний інтерфейс для перегляду.

РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

1.8 Опис ідеї проєкту

Ідея проєкту полягає у розробці автоматизованої інформаційної системи для реферування україномовних наукових текстів. Основна мета системи — допомогти науковцям, викладачам, студентам та дослідникам швидко та якісно створювати реферати на основі великих текстових даних.

Таблиця 5.1 – Опис ідеї проєкту

Зміст ідеї	Напрямки застосування	Вигода для користувача
Розробка алгоритмів автоматичного реферування текстів з використанням ланцюгів Маркова, асоціативних правил, BERT та TextRank	Використання у наукових дослідженнях для автоматичного створення рефератів, зокрема в україномовному академічному середовищі, у медичній галузі для аналізу клінічних звітів та медичних публікацій, а також у бізнес-аналізі для автоматизації скорочення великих звітів і документів.	Забезпечення економії часу на аналіз великих текстових масивів, підвищення якості наукових рефератів, спрощення процесу створення аналітичних матеріалів, а також можливість швидко отримувати релевантну інформацію, зокрема для українських наукових текстів.

Зміст ідеї	Напрямки застосування	Вигода для користувача
Інтеграція алгоритмів для аналізу контексту та ключових фраз з урахуванням мовних особливостей української мови	Використання у освітніх платформах для автоматичного створення коротких анотацій навчальних матеріалів, в бібліотечних системах для автоматизації створення каталогів та у медіа-ресурсах для обробки і узагальнення новин.	Підвищення точності створення рефератів та анотацій, зменшення часу на виконання рутинних завдань, можливість працювати з текстами, що мають специфічні мовні характеристики, та забезпечення контекстної релевантності результатів для користувачів.
Розробка структури бази даних для зберігання текстів та рефератів	Використання у наукових установах для централізованого збереження текстових документів, у компаніях для управління текстовими базами даних, у редакційних системах для створення архівів статей та матеріалів.	Забезпечення зручного доступу до рефератів, можливість швидкого пошуку ключових матеріалів, структуроване зберігання інформації для ефективної обробки великих обсягів даних, що особливо актуально для роботи з науковими текстами.

Зміст ідеї	Напрямки застосування	Вигода для користувача
Створення інтерфейсу користувача для зручної взаємодії з системою	Використання у додатках для наукових співробітників, які працюють із великими текстовими базами, інтеграція у навчальні платформи для студентів і викладачів, а також у системи автоматизованої підготовки документів для бізнесу.	Забезпечення інтуїтивно зрозумілого доступу до функцій системи, спрощення роботи навіть для користувачів без технічної підготовки, можливість адаптувати систему до індивідуальних потреб користувачів і підвищення ефективності обробки текстових даних.
Проведення тестування та аналізу ефективності алгоритмів	Застосування у процесах порівняння автоматичних рефератів з існуючими аналогами, аналіз точності реферування для наукових текстів, використання у розробці нових підходів для узагальнення текстів.	Виявлення сильних і слабких сторін алгоритмів, підвищення якості кінцевого продукту за рахунок аналізу помилок, можливість створення більш адаптивних та точних інструментів для роботи з текстами.

Зміст ідеї	Напрямки застосування	Вигода для користувача
Розробка користувацьких інструментів для налаштування параметрів реферування	Використання у системах із потребою динамічного налаштування, таких як освітні платформи, наукові інститути та бізнес-системи для адаптації обробки текстів.	Забезпечення гнучкості у використанні, можливість підбору оптимальних параметрів залежно від завдань користувача, зокрема регулювання обсягу рефератів та вибору акцентів на ключові аспекти тексту.

Для оцінки конкурентоспроможності розробленої системи було проведено детальний порівняльний аналіз її техніко-економічних характеристик із найбільш популярними аналогами.

Результати аналізу відображені у таблиці 5.2, де визначено переваги, недоліки та характеристики, які не мають суттєвого впливу на загальну ефективність системи.

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W	N	S
		Мій проєкт	QuillBot	Scholarcy	Resoomer			
1	Адаптація до специфіки української мови	+	-	-	-			+

	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W	N	S
		Мій проєкт	QuillBot	Scholarcy	Resoomer			
2	Інтеграція алгоритмів BERT для аналізу контексту	+	+	-	-		+	
3	Автоматичне створення структурованих рефератів	+	+	+	-			+
4	Можливість налаштування параметрів реферування	+	-	-	-			+
5	Зручність інтеграції в освітні платформи	+	-	+	-		+	
6	Підтримка багатомовності для інтернаціональних текстів	+	+	-	-			+
7	Висока швидкість обробки великих текстових масивів	+	+	-	+		+	
8	Інтеграція асоціативних правил для покращення точності	+	-	-	-			+
9	Візуалізація результатів реферування	+	+	+	+			+

	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W	N	S
		Мій проєкт	QuillBot	Scholarcy	Resoomer			
10	Підтримка аналізу текстів із науковим контекстом	+	-	+	-			+

Аналіз показав, що запропонований проєкт має суттєві переваги у таких аспектах, як адаптація до української мови, інтеграція сучасних алгоритмів машинного навчання (BERT, TextRank), а також забезпечення високої точності обробки текстів з урахуванням контексту. У порівнянні з конкурентами, система демонструє кращу відповідність для роботи з україномовними текстами завдяки адаптації алгоритмів та інноваційним підходам до аналізу тексту.

Слабкі сторони проєкту зосереджені переважно у сфері швидкості обробки великих обсягів даних, де окремі конкуренти мають оптимізовані рішення для англомовних текстів. Водночас, це компенсується високою точністю та релевантністю результатів, які є критичними для цільової аудиторії.

Таким чином, проведений аналіз підтверджує доцільність впровадження розробленого проєкту, оскільки він не лише відповідає сучасним вимогам до автоматизації обробки текстів, але й забезпечує значні переваги у сфері роботи з українськими науковими та академічними текстами, що робить його конкурентоспроможним на ринку.

1.9 Технологічний аудит ідеї проєкту

У таблиці 5.3 можна побачити аудит технологій які будуть використані у проєкті.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

№	Ідея проєкту	Технології і реалізації	Наявність технологій	Доступність технологій
1	Back-end	Remix	Наявна	Open Source, безкоштовна ліцензія, активно підтримується спільнотою.
		Python	Наявна	Безкоштовна, доступна для використання на більшості платформ, має велику кількість бібліотек.
2	Front-end	Vite	Наявна	Безкоштовна, доступна для будь-яких проєктів, підтримує сучасні JavaScript-фреймворки.

№	Ідея проєкту	Технології і реалізації	Наявність технологій	Доступність технологій
		Tailwind CSS	Наявна	Відкритий код, безкоштовний для використання, підтримує інтеграцію з багатьма бібліотеками.
3	Сховище даних	MongoDB	Наявна	Доступна безкоштовна Community версія, є платні функції для корпоративного використання.
4	Обробка текстів	BERT	Наявна	Відкрита для використання, потребує адаптації під специфіку україномовних текстів.

№	Ідея проєкту	Технології і реалізації	Наявність технологій	Доступність технологій
		TextRank	Наявна	Вільний доступ, легко інтегрується у Python-проєкти, потребує адаптації під специфіку україномовних текстів.
5	Інтеграція фронтенду та бекенду	REST API	Наявна	Безкоштовна, широко використовується у веб-розробці.

Для реалізації проєкту обрано сучасні технології, які забезпечують високу ефективність, масштабованість і відповідність технічним вимогам. Використання таких інструментів, як Remix і Python для серверної логіки, Vite і Tailwind CSS для фронтенду, а також MongoDB для роботи з базами даних, дозволяє створити гнучку та продуктивну архітектуру системи. Крім того, інтеграція моделей машинного навчання, таких як BERT і TextRank, гарантує точність обробки текстів та адаптацію до україномовних публікацій.

Проведений аналіз підтвердив наявність усіх необхідних технологій для реалізації проєкту. Усі обрані технології є доступними для використання: переважна більшість з них має відкритий код і підтримується активними

спільнотами розробників. Інструменти, такі як Remix, Vite та Tailwind CSS, забезпечують високу продуктивність і швидкість розробки інтерфейсу. MongoDB як NoSQL база даних пропонує широкі можливості для масштабування та зручного зберігання текстових масивів. Моделі BERT і TextRank, що доступні у відкритих репозиторіях, дозволяють адаптувати систему до специфічних потреб користувачів, зокрема україномовних наукових текстів.

1.10 Аналіз ринкових можливостей запуску стартап-проекту

Проаналізувавши поточний ринок існуючих рішень було створено таблицю 5.4, це дає змогу зрозуміти можливість запуску стартап-проєкту.

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	15
2	Загальний обсяг продажів, $	320000 $ на рік.
3	Динаміка ринку (якісна оцінка)	Високі, зростання на 15% щорічно
4	Наявність обмежень для входу	Потреба в отриманні міжнародних сертифікатів, значна конкуренція у сфері, високі витрати на маркетинг

	Показники стану ринку (найменування)	Характеристика
5	Вимоги до якості та сертифікації	ISO 9001:2015 (Системи управління якістю), ISO/IEC 27001:2013 (Системи управління інформаційною безпекою)
6	Середній рівень рентабельності в галузі, %	28%

Проведений аналіз стану ринку продемонстрував перспективність реалізації проєкту у вибраній галузі. Ринок характеризується середньою кількістю основних конкурентів (15), що створює умови для потенційного успіху нового продукту за умови його конкурентоспроможності. Загальний обсяг ринку становить 320 000 доларів на рік, із темпами зростання 15% щорічно, що свідчить про динамічний розвиток і збільшення попиту на подібні рішення.

Наявність бар'єрів для входу, таких як необхідність отримання міжнародних сертифікатів, висока конкуренція та значні витрати на маркетинг, визначає необхідність ретельного планування стратегії виходу на ринок. Водночас дотримання стандартів якості, таких як ISO 9001:2015 та ISO/IEC 27001:2013, є важливою умовою для забезпечення відповідності продукту вимогам клієнтів і підвищення його конкурентоспроможності.

Середній рівень рентабельності у галузі становить 28%, що дозволяє очікувати стабільного фінансового результату за умови ефективної реалізації проєкту. Таким чином, отримані дані підтверджують перспективність вибраної галузі та доцільність впровадження розробки в умовах поточного ринку

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Автоматизація обробки текстів	Наукові установи, видавництва, освітні платформи	Установи орієнтовані на високу точність, освітні платформи – на зручність використання.	Простота інтеграції з іншими системами, підтримка різних мов, зручний інтерфейс.
2	Скорочення часу на аналіз документів	Корпорації, державні установи, юридичні фірми	Корпорації цінують продуктивність, державні установи – відповідність регуляторним вимогам.	Висока швидкість обробки, відповідність стандартам, деталізовані звіти.

№	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
3	Зниження витрат на підготовку рефератів	Бізнес-аналітики, фінансові аналітичні центри	Бізнесу потрібна адаптивність, аналітикам – точність і структурованість даних.	Оптимізація витрат, можливість кастомізації звітів, інтеграція з хмарними сервісами.
4	Використання сучасних методів обробки даних	ІТ-компанії, розробники програмного забезпечення	ІТ-сектор очікує адаптивності алгоритмів, розробники – простоти інтеграції з власними системами.	Широкий набір API, відкриті стандарти, підтримка інноваційних технологій.

Важливо відзначити, що різні цільові аудиторії мають свої специфічні вимоги: наукові установи акцентують увагу на високій точності, бізнес орієнтований на зниження витрат і підвищення продуктивності, а державні установи вимагають відповідності регуляторним стандартам. У зв’язку з цим, розроблений проєкт має значну конкурентну перевагу завдяки підтримці сучасних методів обробки текстів, таких як інтеграція моделей машинного навчання та можливість адаптації до різних сценаріїв використання, а також роботи саме з україномовними текстами.

Таблиця 5.6 – Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Зростання витрат на розробку	Збільшення витрат на розробку продукту через подорожчання ресурсів та інфраструктури	Оптимізація ресурсів, пошук додаткового фінансування або впровадження менш ресурсомістких технологій
2	Нестабільність регуляторних норм	Часті зміни у законодавстві або регуляторних вимогах для роботи в галузі	Створення юридичного відділу для моніторингу змін, адаптація продукту до нових стандартів
3	Недостатня кваліфікація команди	Відсутність у команді спеціалістів із необхідними компетенціями	Підвищення кваліфікації команди через тренінги або залучення зовнішніх експертів
4	Зниження споживчого інтересу	Втрата зацікавленості потенційних клієнтів через появу нових альтернатив	Розробка маркетингової кампанії, покращення функціоналу продукту та зниження ціни для конкурентоспроможності
5	Технологічні обмеження	Відсутність доступу до сучасних технологій або інструментів для реалізації проекту	Впровадження власних розробок, співпраця з технологічними партнерами, пошук альтернативних рішень

Можна виділити головні загрози для реалізації проєкту, такі як збільшення витрат, зміна регуляторних норм, потреба в кваліфікованих кадрах та можливе коливання попиту. Розроблені заходи для подолання цих викликів забезпечують проєкту здатність гнучко реагувати на зовнішні впливи та залишатися конкурентоспроможним у динамічних ринкових умовах.

Таблиця 5.7 – Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Впровадження інновацій	Розробка нових технологій для покращення продуктивності продукту	Залучення інвестицій для досліджень і розробок, пошук нових партнерів
2	Розширення ринку	Зростання попиту на продукт у суміжних галузях	Проведення маркетингових кампаній, адаптація продукту під нові ринки
3	Технологічна співпраця	Можливість інтеграції з іншими інноваційними рішеннями	Налагодження партнерських відносин із розробниками технологій та стартапами.
4	Державна підтримка	Можливість отримання грантів або пільг від державних програм	Залучення ресурсів для участі у державних тендерах або програмах підтримки

№	Фактор	Зміст можливості	Можлива реакція компанії
5	Зростання цифровізації	Загальна тенденція до автоматизації та використання цифрових інструментів	Розширення функціоналу продукту, впровадження нових IT-рішень

Необхідно акцентувати увагу на тому, що аналіз можливостей відкриває перспективи для посилення позицій проєкту на ринку. Виявлені фактори, такі як зростання попиту в нових сегментах, впровадження інноваційних рішень та підтримка цифровізації, формують платформу для стратегічного розвитку. Реалізація запропонованих реакцій на ці можливості забезпечить не лише стійкість у конкурентному середовищі, а й значне підвищення конкурентоспроможності продукту. Таким чином, ефективне використання цих можливостей стане ключовим фактором успішності проєкту.

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Чистий тип конкуренції	Кожен учасник ринку пропонує продукт із подібними характеристиками, але з різною додатковою цінністю	Постійне вдосконалення функціональності продукту, додавання інноваційних можливостей

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Національний рівень конкурентної боротьби	Більшість конкурентів працює лише в межах країни, без виходу на зовнішні ринки	Створення стратегії для виходу на міжнародний ринок, зокрема через локалізацію продукту
Міжгалузева конкуренція	Продукт може бути корисним для кількох різних галузей, але конкуренти представлені в окремих секторах	Адаптація продукту для використання в різних галузях з урахуванням специфіки кожної
Товарно-родова конкуренція	Конкуренція між продуктами, які вирішують однакову проблему, але за різними технологічними підходами	Запропонувати користувачам гнучкі налаштування та багатофункціональність, що відповідає їх потребам
Нецінова конкуренція	Конкуренти акцентують увагу на унікальних характеристиках, а не на зниженні ціни	Розробка маркетингової стратегії, що підкреслює переваги продукту, такі як швидкість, надійність та інноваційність
Висока інтенсивність конкуренції	Ринок насичений великою кількістю схожих пропозицій, що створює сильний тиск на нових гравців	Активна рекламна кампанія, створення додаткових сервісів та програм лояльності для клієнтів

Важливо відзначити, що успіх у цьому середовищі залежить не лише від прямого протистояння з конкурентами, а й від здатності знайти нові ніші та розкрити прихований потенціал ринку. Визначені особливості конкурентного середовища вказують на необхідність постійного вдосконалення продукту, розширення його можливостей та впровадження інновацій.

Конкуренція на ринку переважно нецінова, що створює можливості для впровадження інноваційних рішень та адаптації продукту до сучасних викликів. Ринок розвинений, але залишається відкритим для нових учасників із унікальними перевагами. Постачальники не диктують умов, що спрощує управління ланцюгом постачання. Основна увага має бути зосереджена на клієнтах, які формують високі вимоги до функціоналу, точності та зручності використання. Таким чином, інвестиції в підвищення якості та впровадження інновацій є ключовими для досягнення успіху.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	Resoomer Scholarcy QuillBot	Потенційні гравці з новими інноваційними продуктами	Контроль ціни продуку	Клієнти очікують високої якості, точності та адаптивності продукту	Інші технології, що виконують схожі функції, але можуть мати відмінні принципи роботи
Висновки	Насиченість ринку існуючими гравцями	Високі витрати на розробку, потреба у масштабному маркетингу	Обмежений доступ до специфічних компонентів	Висока вимогливість до ціни та функціоналу	Інтеграція деяких функцій замінників для підвищення конкурентоспроможності

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Якість продукту	Забезпечення високої точності, стабільності та адаптивності продукту до потреб різних категорій клієнтів

№	Фактор конкурентоспроможності	Обґрунтування
2	Технологічна інноваційність	Використання сучасних алгоритмів та методів для забезпечення конкурентних переваг у порівнянні з аналогами
3	Позиціювання на ринку	Чітке визначення цільової аудиторії та адаптація маркетингових стратегій під її потреби
4	Доступність продукту	Гнучка цінова політика, яка враховує регіональні та фінансові особливості клієнтів
5	Швидкість впровадження	Мінімізація часу на встановлення та навчання користувачів для початку роботи з продуктом
6	Партнерство з клієнтами	Активна двостороння комунікація з клієнтами для врахування їхніх побажань у процесі розробки та оновлень
7	Лояльність клієнтів	Запровадження програм лояльності та додаткових сервісів для постійних клієнтів
8	Унікальність функціоналу	Розробка функцій, які недоступні у конкурентів, або забезпечення багатофункціональності продукту
9	Ефективність витрат	Оптимізація витрат на розробку та впровадження, що дозволяє зберігати конкурентоспроможну ціну продукту
10	Реклама та промоція	Використання сучасних каналів просування, таких як соціальні мережі, конференції та партнерські виставки

Аналіз факторів конкурентоспроможності показав, що основою успіху на ринку є не лише якість продукту, але й уміння адаптуватися до мінливих потреб

клієнтів і технологічного середовища. Високий рівень технологічної інноваційності, чітке позиціювання та гнучка комунікація з клієнтами стають ключовими елементами стратегії. Особлива увага до швидкості впровадження, доступності продукту та побудови довготривалих відносин із клієнтами забезпечує компанії конкурентну перевагу. Таким чином, розвиток унікальних функцій, ефективне управління витратами та активна промоція формують основу для успішного просування на ринку.

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін проєкту

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з проєктом						
			-3	-2	-1	0	+1	+2	+3
1	Інноваційність продукту	19							+
2	Якість обслуговування	17						+	
3	Цінова політика	15				+			

Результати порівняльного аналізу продемонстрували, що проєкт має значну перевагу в аспекті інноваційності, що визначає його як ключову силу серед конкурентів. Якість обслуговування також показала високий рівень, забезпечуючи проєкту додаткову конкурентну перевагу. Цінова політика виявилася нейтральною відносно конкурентів, що свідчить про необхідність подальшої оптимізації у цьому напрямку. Таким чином, основний акцент варто зробити на утриманні лідерських позицій у сфері інновацій, покращенні якості обслуговування та створенні гнучкішої цінової стратегії для збільшення конкурентоспроможності.

Таблиця 5.12 – SWOT- аналіз стартап-проєкту

Сильні сторони: – унікальний підхід до розробки продукту; – використання власних алгоритмів та технологій; – відповідність сучасним стандартам якості.	Слабкі сторони: – висока складність впровадження; – обмеженість ринку; – значна залежність від кваліфікації команди.
Можливості: – розширення використання продукту на суміжні ринки; – впровадження інновацій у процес розробки; – використання державної підтримки для масштабування.	Загрози: – поява нових конкурентів із більш доступними рішеннями; – зростання витрат через економічну нестабільність; – зниження попиту через зміну регуляторних норм.

Після аналізу SWOT можна скласти таблицю альтернатив ринкового впровадження стартап-проєкту.

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проєкту

№	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Проведення рекламної кампанії через соціальні мережі	Висока (власні ресурси та доступні платформи)	1 місяць
2	Розробка демонстраційної версії продукту для	Наявні (внутрішній ресурс розробки)	3 місяців

№	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
	безкоштовного тестування		
3	Пошук інвесторів через профільні заходи, наприклад, виставки стартапів	Помірна (залежить від інтересу інвесторів)	6 місяців

Сильні сторони проєкту, зокрема інноваційність рішень та відповідність стандартам, формують основу для успішного розвитку. Можливості, такі як розширення ринку та впровадження нових методів, відкривають перспективи для масштабування. Однак для мінімізації впливу слабких сторін, таких як складність процесів і вузька спеціалізація, необхідно вдосконалювати функціонал продукту та розширювати цільову аудиторію. Загрози, зокрема конкуренція та економічна нестабільність, потребують постійного моніторингу та адаптації стратегії. Успіх проєкту залежатиме від ефективного використання сильних сторін і можливостей, водночас зводячи до мінімуму слабкості та загрози.

1.11 Розроблення ринкової стратегії проєкту

Для успішності проєкту треба провести аналіз ринку та розбити ринок на цільові групи.

З аналізу видно, що основний інтерес до продукту виявляють наукові установи, державні організації та корпоративні аналітичні центри, які цінують точність і адаптивність інноваційних рішень. Простота входу в ці сегменти висока завдяки специфічності потреб і низькому рівню конкуренції. Освітні платформи та юридичні компанії також демонструють значний потенціал, хоча конкуренція в цих сегментах є більш інтенсивною. Таким чином, для успішного виходу на ринок

необхідно зосередитися на сегментах із високою готовністю споживачів і мінімізувати ризики в умовах підвищеної конкуренції.

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Наукові установи	Висока	85%	Низька інтенсивність	Висока
2	Освітні платформи	Середня	70%	Середня інтенсивність	Середня
3	Корпоративні аналітичні центри	Висока	80%	Висока інтенсивність	Середня
4	Юридичні компанії	Середня	65%	Середня інтенсивність	Середня
5	Фінансові установи	Висока	75%	Середня інтенсивність	Низька
6	Державні організації	Висока	90%	Низька інтенсивність	Середня

Таблиця 5.15 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспро-можні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Інтеграція інноваційних технологій у процеси автоматизації	Охоплення вузьких цільових сегментів	Висока точність, адаптивність до потреб клієнтів	Стратегія фокусування
Розширення функціональності продукту	Стратегія національного охоплення	Зручність використання, широкий набір функцій	Стратегія диференціації
Вихід на міжнародний ринок	Локалізація продукту під вимоги іноземних користувачів	Відповідність міжнародним стандартам, адаптація мовних та технічних аспектів	Стратегія диференціації

Розробка базової стратегії розвитку проєкту дозволила визначити ключові напрямки його подальшого просування та вдосконалення. Інтеграція інноваційних технологій та фокусування на вузьких цільових сегментах дозволяють максимально задовольнити потреби клієнтів, зберігаючи конкурентну перевагу. Розширення функціональності продукту забезпечує диференціацію, сприяючи охопленню національного ринку. Вихід на міжнародний ринок відкриває нові

можливості, формуючи основу для диверсифікації та масштабування проєкту. Таким чином, запропоновані стратегії охоплення ринку гармонійно поєднують технологічні інновації, орієнтацію на клієнта та можливості для глобального зростання.

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні, на ринку вже присутні подібні рішення, які відзначаються обмеженою адаптивністю та високою вартістю	Компанія буде залучати нових клієнтів, які не задоволені поточними рішеннями, а також орієнтуватиметься на частковий перехід клієнтів від конкурентів, демонструючи переваги свого продукту	Компанія не буде копіювати характеристики конкурентів, але використовуватиме їх досвід для визначення точок вдосконалення та побудови унікальних переваг продукту	Стратегія диференціації через інновації

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні, ринок насичений рішеннями, орієнтованими на специфічний функціонал без урахування ширших потреб клієнтів	Основна увага буде зосереджена на розширенні цільової аудиторії та виході на нові ринкові сегменти, де конкуренція є менш інтенсивною	Компанія буде досліджувати практики конкурентів для адаптації найкращих підходів, але не копіюватиме функціонал, фокусуючись на впровадженні інноваційних рішень	Стратегія фокусування на нішевому сегменті

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні, існують усталені рішення, але їхні характеристики не завжди відповідають сучасним технологічним стандартам	Компанія планує залучити нових клієнтів шляхом пропозиції більш ефективного та економічно вигідного продукту, який вирішує потреби, які залишаються поза увагою конкурентів	Компанія буде враховувати найбільш вдалі характеристики рішень конкурентів, зокрема інтуїтивність інтерфейсу, але розширюватиме функціонал для створення унікальних можливостей	Стратегія адаптивної конкуренції

Проєкт не є “першопрохідцем” на ринку, однак це створює сприятливі умови для впровадження інновацій, оскільки існуючі рішення не завжди повністю задовольняють сучасні потреби клієнтів. Компанія орієнтується на залучення нових клієнтів, які незадоволені поточними продуктами, а також частково на клієнтів конкурентів, пропонуючи їм ефективніші, економічно вигідніші та адаптивніші рішення. Водночас стратегія компанії не передбачає прямого копіювання основних характеристик товарів конкурентів. Натомість, використовуючи досвід інших гравців ринку, компанія вдосконалює свій продукт, впроваджуючи унікальні функції та інноваційні підходи.

Таблиця 5.17 – Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспро-можні позиції власного стартап-проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту
1	Висока точність та відповідність вимогам наукових стандартів	Стратегія фокусування	Унікальні алгоритми обробки тексту; висока швидкість генерації рефератів; точність даних	Технологічність, наукова надійність, швидкість
2	Простота використання для користувачів	Стратегія диференціації	Інтуїтивний інтерфейс; інтеграція з популярними форматами текстів	Зручність, доступність, інтегративність
3	Широкий набір функцій для різних цільових груп	Стратегія розширення функціональності	Можливість гнучкої адаптації до специфічних потреб; підтримка багатомовності	Мультифункціональність, універсальність, глобальність

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспро-можні позиції власного стартап-проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту
4	Адаптивність до специфічних потреб наукових установ	Стратегія індивідуалізації	Налаштування для різних наукових дисциплін; гнучкі алгоритми	Персоналізація, точність, адаптивність
5	Висока швидкість обробки великих обсягів текстів	Стратегія технологічного вдосконалення	Використання сучасних обчислювальних моделей для прискорення аналізу	Продуктивність, оптимізація, сучасність
6	Підтримка інтеграції з іншими науковими інструментами	Стратегія партнерства	Інтеграція з базами даних, бібліотечними платформами та науковими порталами	Синхронізація, ефективність, співпраця
7	Можливість генерації детальних звітів і аналітики	Стратегія розширення функціональності	Створення багаторівневих звітів з ключовими висновками для різних категорій користувачів	Аналітичність, деталізація, професійність

Розроблена стратегія позиціонування системи реферування наукових текстів базується на чіткому розумінні ключових вимог ринку та потреб користувачів. Висока точність, відповідність науковим стандартам, швидкість обробки текстів та адаптивність до специфічних потреб різних аудиторій формують конкурентоспроможні позиції продукту. Впровадження багатомовності, інтеграція з іншими науковими інструментами та забезпечення безпеки даних значно підсилюють його привабливість на глобальному ринку.

Особливу увагу приділено інтуїтивності інтерфейсу, що забезпечує зручність використання для різних категорій користувачів, та можливості персоналізації, яка дозволяє адаптувати продукт до конкретних дисциплін та запитів клієнтів. Така стратегія розвитку дозволяє не тільки зайняти провідну позицію у своєму сегменті, але й формувати нові стандарти якості та функціональності у сфері реферування текстів.

1.12 Розроблення маркетингової програми стартап-проекту

Для того щоб побудувати гарну маркетингову програму треба розглянути загальну конкурентноспроможність.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Точність реферування та відповідність стандартам	Забезпечення високої точності обробки тексту	Унікальні алгоритми аналізу тексту; відповідність науковим стандартам; автоматична перевірка наукових цитат

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
2	Зручність використання	Інтуїтивний інтерфейс для всіх категорій користувачів	Простота роботи; інтеграція з популярними форматами текстів, мінімізація навчальної кривої
3	Швидкість обробки великих текстів	Обробка та реферування текстів у кілька разів швидше за аналоги	Використання сучасних обчислювальних технологій, паралельна обробка великих масивів даних
4	Гнучкість та адаптивність	Можливість налаштування для різних наукових дисциплін	Підтримка багатомовності, адаптація до специфіки галузей
5	Конфіденційність даних	Забезпечення безпеки користувацької інформації	Впровадження шифрування, відповідність міжнародним стандартам безпеки даних
6	Широкі функціональні можливості	Генерація детальних рефератів з аналізом основних ідей тексту	Автоматизація структуризації, генерація звітів, підтримка аналітики

Система забезпечує високу точність обробки та відповідність сучасним науковим стандартам, що робить її незамінним інструментом для науковців і

дослідників. Інтуїтивний інтерфейс і зручність використання сприяють залученню широкої аудиторії користувачів, мінімізуючи складність навчання.

Використання сучасних технологій забезпечує значну швидкість обробки великих обсягів україномовного тексту, що ставить продукт у виграшну позицію порівняно з аналогами. Гнучкість та адаптивність системи, зокрема підтримка багатомовності й можливість налаштування під специфіку окремих галузей, роблять її придатною для різних сфер використання. Крім того, забезпечення конфіденційності даних та відповідність міжнародним стандартам безпеки підвищують рівень довіри до продукту.

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Система для автоматичного створення рефератів наукових текстів українською мовою з урахуванням сучасних стандартів академічного письма та стилістики.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Інтуїтивний інтерфейс з підтримкою кількох мов, включаючи українську	Нм	Тх/Тл
	2. Швидка обробка текстів (до 100 сторінок за хвилину)	М	Тл
	3. Точність реферування до 95% відповідно до специфіки української наукової термінології	М	Ор
	4. Автоматична перевірка наукових цитувань і відповідність стандартам академічного письма	Нм	Тх

Рівні товару	Сутність та складові		
	5. Генерація рефератів у різних форматах (PDF, DOCX, TXT)	M	E
	Якість: відповідність українським і міжнародним стандартам академічного письма		
	Пакування: цифровий продукт із локалізованою документацією українською мовою; доступ через вебплатформу		
	Марка: Назва організації-розробника + “Інноваційна система реферування наукових текстів”.		
III. Товар із підкріпленням	До продажу: технічна підтримка користувачів, регулярні оновлення продукту, організація сертифікації та навчання користувачів		
	Після продажу: стабільна підтримка і оновлення застосунку, оформлення ліцензій та сертифікацій		
За рахунок чого потенційний товар буде захищено від копіювання: патентування унікальних алгоритмів обробки текстів, інтеграція з національними та міжнародними базами наукової інформації, шифрування даних і захист інтелектуальної власності			

На рівні задуму система спрямована на задоволення основної потреби користувачів у точному, швидкому та адаптивному створенні рефератів українською мовою, що відповідають сучасним науковим стандартам. Реалізація продукту включає унікальні характеристики, такі як багатомовний інтерфейс, швидкість обробки текстів до 100 сторінок за хвилину, точність реферування до 95% та інтеграція з сучасними форматами даних. Ці переваги підсилюються високими стандартами якості та доступністю для користувачів.

Таблиця 5.20 – Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
200–300 грн/місяць	250–400 грн/місяць	Середній дохід – 10 000–15 000 грн/місяць	Нижня межа: 250 грн/місяць; Верхня межа: 350 грн/місяць

Таблиця 5.21 – формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Потенційні клієнти купують ліцензію на програму або доступ до вебплатформи	Налаштування середовища, інтеграція з іншими системами, технічна підтримка	Нульова	Вебплатформа/хмарний сервіс
Наукові установи замовляють масштабні рішення для локальних серверів	Налаштування серверного середовища, навчання персоналу	Одна ланка	Ліцензія для локальних серверів
Університети чи бібліотеки використовують демоверсії для ознайомлення	Презентації продукту, підготовка демоверсій	Нульова	Демо-доступ для користувачів

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Індивідуальні користувачі набувають ліцензії через офіційний сайт	Продаж через сайт, автоматична активація продукту	Нульова	Продаж через офіційний вебресурс

Таблиця 5.22 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Потенційні клієнти шукають ефективні інструменти для скорочення часу на підготовку наукових рефератів	Інтернет-платформи, соціальні мережі, тематичні наукові форуми	Інноваційність, швидкість обробки, точність результатів	Ознайомити з функціональністю системи, акцентуючи на її інноваціях, точності та економії часу	Використання відеопрезентацій і текстових оголошень із результатами швидкої обробки та прикладами рефератів, які підкреслюють точність

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Університети та бібліотеки зацікавлені в інструментах для автоматизації процесу роботи з науковими текстами	Прямі переговори, тематичні конференції, презентації на освітніх платформах	Технологічність, відповідність науковим стандартам	Продемонструвати практичну користь продукту для освітніх установ, підкреслюючи його універсальність та зручність	Рекламні матеріали у вигляді PDF-презентацій, а також демонстрація роботи системи в реальному часі на конференціях чи зустрічах

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Індивідуальні користувачі прагнуть отримати зручний інструмент для роботи над науковими публікаціями	Таргетована реклама, емейл-маркетинг	Доступність, зручність, багатомовність	Показати простоту використання продукту навіть для новачків, акцентуючи увагу на функціоналі, адаптованому під українську наукову термінологію	Банери, які демонструють зрозумілість інтерфейсу, та емейл-розсилки з посиланням на демоверсію продукту, що дозволяє одразу протестувати систему

Розроблена концепція маркетингових комунікацій системи реферування наукових текстів враховує специфіку поведінки різних цільових груп, таких як науковці, університети, бібліотеки, індивідуальні користувачі та дослідницькі інститути. Для кожної категорії клієнтів визначено оптимальні канали комунікації, які дозволяють максимально ефективно донести переваги продукту.

Ключові позиції позиціонування, такі як інноваційність, швидкість, точність і багатомовність, підкреслюють конкурентні переваги системи. Завдяки інтерактивним демонстраціям, таргетованій рекламі, презентаціям на

конференціях та тематичних заходах забезпечується залучення нових користувачів і підвищується довіра до продукту.

Важливими завданнями рекламного повідомлення є демонстрація простоти використання, відповідність сучасним науковим стандартам, а також універсальність і адаптивність продукту до потреб різних клієнтів. Концепція рекламного звернення орієнтована на практичні приклади роботи системи, реальні результати та користь для споживачів, що сприяє формуванню позитивного іміджу продукту та закріпленню його позицій на ринку.

Висновки до розділу

Проведений аналіз і розробка системи реферування наукових текстів, представлені в таблицях, охоплюють ключові аспекти її створення, позиціонування, просування на ринку та забезпечення конкурентоспроможності. Завдяки комплексному підходу, який враховує всі рівні розробки продукту, визначення його особливостей, адаптацію до потреб клієнтів та оптимізацію маркетингових стратегій, система отримала чітку структуру та ефективну модель впровадження на ринок.

У таблицях, присвячених формуванню продукту, була побудована трирівнева модель товару, яка враховує не лише його задум, але й конкретні характеристики реалізації та підкріплення після продажу. Висвітлено унікальні особливості системи, такі як високоточні алгоритми, багатомовність, адаптація до української наукової термінології, швидкість обробки текстів, простота у використанні та відповідність міжнародним стандартам. Додатково визначено способи захисту від копіювання, зокрема патентування алгоритмів, забезпечення конфіденційності даних та інтеграція з національними і міжнародними науковими базами.

Аналіз цільової аудиторії та її потреб дозволив виявити сегменти ринку, що зацікавлені у продукті, включаючи університети, наукові інституції, бібліотеки,

дослідницькі центри та індивідуальних користувачів. У таблицях визначено специфіку поведінки цих груп, рівень їх доходів, готовність сприйняти продукт, а також побудовано маркетингові стратегії, спрямовані на ефективне залучення кожного сегмента.

Система збуту враховує глибину каналів продажу та особливості кожного клієнта. Розроблено оптимальні стратегії дистрибуції, які дозволяють ефективно реалізувати продукт через інтернет-платформи, ліцензії для локальних серверів, презентації на виставках і прямий продаж університетам. Такий підхід забезпечує як масштабне охоплення ринку, так і індивідуальну роботу з окремими клієнтами.

Значну увагу приділено концепції маркетингових комунікацій, яка включає використання сучасних каналів взаємодії з клієнтами, таких як соціальні мережі, тематичні форуми, освітні конференції та виставки. Рекламні кампанії розроблено з акцентом на ключові переваги системи — інноваційність, зручність, точність і швидкість. Важливим елементом є персоналізація повідомлень для кожного сегмента аудиторії, що підвищує довіру та ефективність комунікацій.

Також було проведено порівняльний аналіз цінової політики, що дозволило встановити верхні та нижні межі ціни на продукт залежно від рівня цін на аналоги, товарів-замінників і доходів цільових клієнтів. Цей аналіз забезпечує конкурентоспроможність продукту на ринку, роблячи його доступним для широкого кола користувачів.

ВИСНОВКИ

Результати проведеної роботи спрямовані на вирішення актуальної науково-практичної задачі створення інформаційної системи для автоматичного реферування україномовних наукових текстів, яка відповідає сучасним вимогам точності, адаптивності та ефективності. У ході виконання дисертації проведено всебічний аналіз сучасних рішень у сфері автоматизованого реферування текстів, що дозволило виявити ключові проблеми, зокрема низьку адаптованість алгоритмів до специфіки української мови та особливостей наукових текстів.

Для вирішення цих проблем розроблено інноваційну систему, яка інтегрує кілька підходів, кожен з яких виконує специфічну роль. Ланцюги Маркова дозволяють моделювати короткострокові ймовірнісні залежності між текстовими елементами, забезпечуючи базову структуру реферату. Алгоритм BERT, перенавчений на локальних корпусах української мови, забезпечує глибокий контекстуальний аналіз тексту, розпізнаючи семантичні зв’язки між словами та реченнями. TextRank, у свою чергу, використовується для виділення ключових слів і фраз, що додає релевантності згенерованим рефератам.

Одним із основних досягнень роботи є адаптація сучасних алгоритмів до специфіки української мови. Зокрема, система враховує морфологічні та синтаксичні особливості, що дозволяє підвищити точність і релевантність результатів. Для цього проведено налаштування токенізації, видалення стоп-слів та стемінгу, що забезпечило коректну роботу алгоритмів на україномовних текстах. Додатково впроваджено алгоритми оцінки якості рефератів, які враховують такі метрики, як повнота, точність та релевантність, дозволяючи оцінювати ефективність системи.

Система є зручною у використанні завдяки розробленому графічному інтерфейсу, який дозволяє користувачам налаштовувати параметри реферування, такі як бажана довжина реферату чи акцент на ключових аспектах тексту. Це забезпечує високу гнучкість і робить продукт універсальним для різних категорій

користувачів, включаючи науковців, студентів та працівників академічних установ.

Практична частина дослідження включає створення програмного продукту, що реалізує функціонал автоматичного реферування текстів. Програмне забезпечення протестовано на реальних даних, зокрема на корпусах українських наукових статей, що дало змогу оцінити його ефективність у порівнянні з іншими популярними системами, такими як Scholarcy, QuillBot та інші. Тестування показало, що розроблена система забезпечує точність реферування, яка перевищує показники існуючих рішень, адаптованих до англомовних текстів.

Розробка також включає створення структури бази даних для зберігання текстів і рефератів, що забезпечує можливість подальшого розширення системи та інтеграції з іншими платформами. Запропоновано стратегії масштабування продукту, включаючи розширення функціоналу для роботи з багатомовними текстами, що відкриває можливості для міжнародного застосування.

У рамках дослідження було розроблено план комерціалізації проекту, який включає аналіз ринкових можливостей, створення маркетингової програми та розроблення стратегії просування. Проведено оцінку конкурентоспроможності продукту, яка підтвердила його інноваційність і потенціал на ринку програмного забезпечення для обробки текстів.

Проведене дослідження не лише вирішує важливу науково-практичну задачу, але й створює передумови для подальшого вдосконалення продукту. Зокрема, передбачено можливість інтеграції нових алгоритмів, таких як GPT, та додавання функціоналу для аналізу емоційного забарвлення тексту. Загальний підхід до реферування може бути масштабований для використання в інших галузях, таких як автоматизація роботи із законодавчими актами або комерційними документами.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Article Summarizer—Read Faster and Improve Comprehension [Електронний ресурс] – Режим доступу до ресурсу: https://quillbot.com/summarize.
2. What is Quillbot and how does it work? [Електронний ресурс] – Режим доступу до ресурсу: https://www.quora.com/What-is-Quillbot-and-how-does-it-work.
3. SMMRY - Automatically creates a TL:DR for an article [Електронний ресурс] – Режим доступу до ресурсу: https://www.reddit.com/r/InternetIsBeautiful/comments/aq31dg/smmry_automatically_creates_a_tldr_for_an_article/.
4. SMMRY [Електронний ресурс] – Режим доступу до ресурсу: https://aitoptools.com/tool/smmry/.
5. What is Resoomer? [Електронний ресурс] – Режим доступу до ресурсу: https://www.futurepedia.io/tool/resoomer.
6. A Quick Introduction to Text Summarization in Machine Learning [Електронний ресурс] – Режим доступу до ресурсу: https://towardsdatascience.com/a-quick-introduction-to-text-summarization-in-machine-learning-3d27ccf18a9f.
7. Two minutes NLP — Four different approaches to Text Summarization [Електронний ресурс] – Режим доступу до ресурсу: https://medium.com/nlplanet/two-minutes-nlp-four-different-approaches-to-text-summarization-5a0ce9c06c74.
8. Understanding Automatic Text Summarization-1: Extractive Methods [Електронний ресурс] – Режим доступу до ресурсу: https://towardsdatascience.com/understanding-automatic-text-summarization-1-extractive-methods-8eb512b21ecc.
9. A study of extractive summarization of long documents incorporating local topic and hierarchical information [Електронний ресурс] – Режим доступу до ресурсу: https://www.nature.com/articles/s41598-024-60779-z.

10. Markov Chains [Електронний ресурс] – Режим доступу до ресурсу: https://brilliant.org/wiki/markov-chains/.
11. Association Rule Learning [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/association-rule-learning.
12. BERT Language Model [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/bert-language-model.
13. BERT Applications [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/bert-applications.
14. Konigsberg Bridge Problem in Discrete Mathematics [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/konigsberg-bridge-problem-in-discrete-mathematics.
15. K-Means Clustering Algorithm [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning.
16. MariaDB Tutorial [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/mariadb-tutorial.
17. React.js MCQ (Multiple Choice Questions) [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/reactjs-mcq.
18. What is DOM in React? [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/what-is-dom-in-react.
19. Next.js Alternatives [Електронний ресурс] – Режим доступу до ресурсу: https://remix.run/.
20. Angular Server-side Rendering (SSR) with Angular Universal [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/angular-server-side-rendering-with-angular-universal.
21. React Forms [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/react-forms.
22. Tailwind CSS in React [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/tailwind-css-in-react.

23. Radix UI [Електронний ресурс] – Режим доступу до ресурсу: https://www.radix-ui.com/.
24. Mammoth.js [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/mwilliamson/mammoth.js.
25. Flask Python Tutorial: What It is and How to Install [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/flask-tutorial.
26. MongoDB Tutorial: What It is and Features [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/mongodb-tutorial.