Міністерство освіти і науки України

Київський національний університет імені Тараса Шевченка
Навчально-науковий інститут філології
кафедра української мови та прикладної лінгвістики

ТЕМА
«Автоматичне реферування тексту з використанням абстрактивних та екстрактивних методів»

Кваліфікаційна робота бакалавра
студентки 4 курсу
освітньої програми
«Прикладна (комп'ютерна) лінгвістика та англійська мова»
спеціальності – 035.10 Філологія (прикладна лінгвістика)
галузі знань – 03 гуманітарні науки
Єлизавети ТЕРНОВСЬКОЇ
Науковий керівник:
д.філол.н., проф. Наталія ДАРЧУК

«Допущено до захисту»
Протокол засідання
кафедри української мови та прикладної лінгвістики
протокол № 15 від «06» 06 2024 року

завідувач кафедри _____(підпис)
к.філол.н., доц. Сергій ВІЗНИК

КИЇВ – 2024

Анотація

Актуальність дослідження автоматичного реферування текстів зумовлена зростаючим обсягом інформації, який необхідно обробляти у сучасному світі. Це дослідження присвячено розробці та оцінці ефективності програмного забезпечення для автоматичного реферування текстів, яке використовує екстрактивні та абстрактивні методи. Об'єктом дослідження є процес автоматичного реферування текстів, а предметом — ефективність екстрактивних та абстрактивних методів у контексті їх точності, повноти, зрозумілості та інших важливих характеристик.

Метою дослідження є створення програми для автоматичного реферування текстів та оцінка її ефективності. Основні завдання включають розробку алгоритмів для екстрактивного та абстрактивного реферування, оцінку їх роботи за допомогою метрик ROUGE, порівняння результатів та визначення сильних і слабких сторін кожного підходу, а також пропозиції щодо можливих шляхів покращення алгоритмів.

Методологічна основа дослідження включає сучасні теорії та підходи до обробки природної мови (NLP) та машинного навчання. Було використано емпіричні дослідження для оцінки роботи алгоритмів за допомогою метрик ROUGE, а також спостереження та аналіз результатів.

За підсумками проведеного дослідження було встановлено, що екстрактивний метод демонструє вищу точність та лексичне багатство, проте страждає від низької повноти та когезії. Абстрактивний метод, хоча і є менш точним, забезпечує кращу зрозумілість та адекватність викладу. Запропоновано комбінований підхід для поєднання переваг обох методів.

Ключові слова: автоматичне реферування, екстрактивне реферування, абстрактивне реферування, NLP, машинне навчання, метрики ROUGE, алгоритми реферування.

Abstract

The relevance of text summarization research is driven by the increasing volume of information that needs to be processed in the modern world. This study is dedicated to the development and evaluation of software for automatic text summarization using extractive and abstractive methods. The object of the study is the process of automatic text summarization, and the subject is the effectiveness of extractive and abstractive methods in terms of their accuracy, completeness, readability, and other important characteristics.

The aim of the study is to create a program for automatic text summarization and to evaluate its effectiveness. The main tasks include the development of algorithms for extractive and abstractive summarization, evaluation of their performance using ROUGE metrics, comparison of results, identification of the strengths and weaknesses of each approach, and suggestions for possible improvements.

The methodological basis of the research includes modern theories and approaches to natural language processing (NLP) and machine learning. Empirical studies were conducted to evaluate the performance of the algorithms using ROUGE metrics, and observations and analyses of the results were performed.

The study found that the extractive method demonstrates higher accuracy and lexical richness but suffers from low completeness and cohesion. The abstractive method, although less accurate, provides better readability and adequacy of expression. A combined approach is proposed to leverage the advantages of both methods.

Keywords: automatic summarization, extractive summarization, abstractive summarization, NLP, machine learning, ROUGE metrics, summarization algorithms

Зміст

Вступ

Обробка природної мови (NLP) намагається подолати комунікативний розрив між людьми та комп'ютерами. Ключовою сферою в NLP є реферування тексту, яке має на меті стиснути великі тексти у коротші форми, зберігаючи основний зміст. Ця технологія допомагає користувачам, дозволяючи ефективно обробляти величезні обсяги інформації [1].

Автоматичне реферування тексту надає безліч переваг у різних сферах. У медіасфері реферат дає читачам швидкий огляд поточних подій [2]. Наукові дослідники можуть використовувати реферат, щоб визначити необхідні статті в обширних базах даних досліджень [3]. Крім того, інструменти реферування мають значну цінність для правознавців, оскільки узагальнюють довгі контракти та документи, що сприяє швидкому всебічному розумінню важливих моментів [4].

Актуальність теми та практичне значення

Значення реферування тексту особливо виразно для таких мов, як українська. Враховуючи зростаючий обсяг цифрової інформації українською мовою, автоматизовані засоби реферування можуть значно покращити здатність користувачів ефективніше орієнтуватися у великому інформаційному просторі.

Мета й завдання дослідження

Метою даного дослідження є розробка програми для автоматичного реферування текстів з використанням як екстрактивного, так і абстрактивного методів, а також оцінка та порівняння результатів роботи цих методів. Для досягнення цієї мети були поставлені такі завдання:

- Вивчити наявні матеріали за темою.

- Розробити систему для екстрактивного та абстрактивного реферування текстів.
- Оцінити результати її роботи за допомогою метрик.
- Порівняти ефективність екстрактивного та абстрактивного методів.
- Визначити сильні та слабкі сторони кожного з підходів.
- Запропонувати можливі шляхи покращення алгоритмів реферування.

Об'єкт і предмет дослідження

Об'єктом дослідження є процес автоматичного реферування текстів. Предметом дослідження є ефективність різних методів автоматичного реферування у контексті їх точності, повноти, зрозумілості та інших важливих характеристик.

Методологічна основа дослідження

Методологічну основу дослідження складають сучасні теорії та підходи до автоматичної обробки текстів. Дослідження спирається на роботи провідних фахівців у галузі обробки природної мови (NLP) та машинного навчання.

Методи вирішення поставлених завдань

Для вирішення поставлених завдань було створено код на мові Python, у якому використано бібліотеки nltk, torch, deep_translator, sumy, transformers, os, а також засоби моделі для абстрактивного реферування Pegasus[41]. Також було запропоновано критерії для оцінки автоматичних рефератів та використано метрики ROUGE.

Для реферування було обрано наукову статтю Камілли Владиславівни Вороніної “ОКАЗІОНАЛІЗМИ ДЖ. ДЖОЙСА В УКРАЇНСЬКОМУ ПЕРЕКЛАДІ” [42], а для референсного реферату її анотацію.

Результати дослідження можуть дозволити наблизитись до визначення оптимальні методи для автоматичного реферування текстів та запропонувати

напрямки для подальшого вдосконалення алгоритмів, що має значне практичне значення для підвищення ефективності роботи з великими обсягами інформації.

2. Загальний огляд літератури

Реферування тексту — створення скороченої версію абзацу, статті чи книги, зі збереженням більшої частини сенсу оригінального тексту.

Реферат — вихідний продукт реферування.

2.1 Методи реферування тексту

Автоматичне реферування тексту охоплює ряд методологій для автоматичного генерування стислого представлення великого текстового вмісту. У цій галузі домінують два основні підходи: екстрактивний та абстрактивний.

- **Екстрактивне реферування:** Цей підхід визначає та відбирає ключові речення з оброблюваного тексту для створення реферату. Методи часто покладаються на такі фактори, як позиція речення, частота слів і статистичні метрики для оцінки важливості речення [5]. До популярних методик відносяться:
    - **Виділення ключових слів:** визначення ключових слів і їх пріоритетність допомагає визначити центральні теми в тексті [5].
    - **Оцінка речень:** процес відбору складається з присвоєння балів реченням на основі різних критеріїв, таких як довжина речення, розташування та наявність ключових слів [5].
- **Абстрактивне реферування:** цей метод виходить за рамки простого відбору речень. Він спрямований на розуміння основного значення тексту та створення нової, стислої версії, яка охоплює суть, потенційно використовуючи інші слова та фрази [6]. Цей підхід використовує такі

досягнення в обробці природної мови, як:

- **Natural Language Understanding (NLU):** Техніки, що дозволяють комп'ютеру обробляти та розуміти значення людської мови, такі як використання в навчанні загальних синтаксичних та граматичних правил, відіграють вирішальну роль у абстрактивному реферуванні [7].
- **Машинний переклад:** Абстрактивне реферування можна розглядати як форму «перекладу» з оригінального тексту на коротшу версію, що запозичує концепції з досліджень машинного перекладу [8].

Основні інструменти:

- **LexRank:** цей алгоритм використовує методи на основі графів для визначення найважливіших речень [9].
- **Gensim:** Python бібліотека з відкритим вихідним кодом надає багатий інструментарій для різноманітних завдань NLP, включаючи функції реферування тексту (не доступні у версіях після 3.8.0) [10].
- **BART (Bidirectional and Autoregressive Transformers for Pre-training):** модель використовує досягнення в архітектурах трансформерів для абстрактивного узагальнення, маючи найвищу продуктивність на різних тестах [11].
- **sumy:** бібліотека та утиліта командного рядка для отримання резюме зі сторінок HTML або звичайних текстів; також містить просту структуру оцінки для текстових резюме [12].

2.2 Труднощі в обробці українських текстів

Труднощі в обробці українських текстів, незважаючи на те, що вищезазначені техніки та інструменти дають вражаючі результати, застосування їх для

реферування українського тексту наштовхується на деякі унікальні проблеми. Перш за все, українська мова має багату флективну морфологію, тобто слова можуть зазнавати значних змін залежно від своєї граматичної ролі. Це створює додаткові труднощі для методів, які зосереджені на аналізі окремих слів, таких як вилучення ключових слів, оскільки морфологічні варіації ускладнюють ідентифікацію базових форм слів і, відповідно, визначення їх важливості [15].

Крім того, обмеженість ресурсів є ще однією значною проблемою. Порівняно з англійською мовою, для української існує значно менший обсяг навчальних даних, доступних для моделей обробки природної мови (NLP). Це обмежує продуктивність підходів, які керуються даними, таких як абстрактне реферування, де великі обсяги даних є критично важливими для навчання ефективних моделей [16]. Наприклад, відсутність великих корпусів текстів і спеціалізованих словників для української мови ускладнює створення високоякісних моделей машинного навчання.

Синтаксична варіативність української мови також ускладнює автоматичне реферування. Український синтаксис має ширший діапазон структур речень порівняно з англійською, що робить складним для систем реферування точне визначення важливості речення на основі частоти або позиції слова. Ця синтаксична різноманітність означає, що системи мають бути досить гнучкими, щоб розпізнавати різні способи вираження однієї і тієї ж думки, що може бути викликом для багатьох сучасних алгоритмів [16].

Наявні дослідження з українського NLP

Незважаючи на ці виклики, дослідники активно розробляють інструменти та ресурси для обробки української мови. Одним з таких проектів є UD Ukrainian IU, який надає структуру аналізу залежностей, спеціально розроблену для українського тексту. Синтаксичний аналіз залежностей допомагає визначити зв’язки між словами, що є вирішальним для таких завдань, як реферування [13].

Іншим корисним інструментом є Multilingual Text Processing Toolkit (MTP), який містить ресурси та інструменти для роботи з різними мовами, включаючи українську. Цей інструментарій надає можливості для обробки тексту, що значно спрощує роботу з українськими текстами в багатомовному контексті [14].

Таким чином, використовуючи наявні ресурси та вирішуючи специфічні лінгвістичні проблеми української мови, ми маємо можливість розробити надійніші та точніші системи узагальнення тексту для цієї мови. Це дозволить підвищити якість автоматичного реферування та забезпечити більш ефективну обробку великих обсягів інформації українською мовою.

3. Екстрактивне реферування

3.1 Принципи екстрактивного реферування

Екстрактивне реферування ґрунтується на припущенні, що найважливіша інформація в тексті міститься в конкретних реченнях. Основний принцип цього підходу полягає у визначенні та виділенні ключових речень для створення стислого резюме. Цей підхід використовує різні методи для оцінки важливості речень. Наприклад, положення речення у тексті відіграє значну роль: речення на початку або в кінці абзаців часто передають важливу інформацію, тому вони мають пріоритет [5]. Довжина речення також є важливим фактором, оскільки довші речення можуть містити більше деталей і, відповідно, мати більшу вагу [5]. Виділення ключових слів включає визначення частоти ключових слів у кожному реченні, що допомагає точно визначити центральні теми. Речення, багаті ключовими словами, вважаються більш важливими [2]. Оцінка речення здійснюється шляхом присвоєння балів кожному реченню на основі комбінації таких факторів, як позиція, довжина, наявність ключових слів та синтаксичні

особливості. Речення з найвищими балами відбираються для створення реферату [18].

3.2 Загальні алгоритми та методи

Існують декілька основних алгоритмів і методів, що використовують для систем екстрактивного реферування.

Як вказано вище, алгоритми виділення ключових слів ідентифікують і ранжують ключові слова на основі їх частоти та релевантності в документі. Загальні підходи включають TF-IDF (Term Frequency-Inverse Document Frequency), який враховує як частоту слова в документі, так і його рідкість у ширшому корпусі [19]. Цей метод допомагає відрізнити найбільш інформативні слова від просто часто вживаних, але не обов’язково важливих.

Feature-based ranking (ранжування на основі ознак) передбачає комбінування таких показників, як довжина речення, позиція та наявність іменованих сутностей, щоб отримати бал для кожного речення. Розширені моделі можуть також включати семантичні особливості, такі як векторне представлення слів, щоб краще відобразити контекстуальну важливість кожного речення. Речення, які перевищують певний пороговий бал, потім відбираються для реферату, забезпечуючи включення найбільш релевантних та інформативних речень [20].

Graph-based ranking algorithms (алгоритми ранжирування на основі графів) такі як LexRank, використовують структури графів для моделювання зв’язків між реченнями. Ці методи часто використовують косинусну подібність для вимірювання близькості речень і побудови графіка, де вузли представляють речення, а ребра — їх подібність. Речення з більшою кількістю зв’язків з іншими важливими реченнями вважаються більш центральними та мають пріоритет для включення, забезпечуючи послідовний та вичерпний реферат [21].

3.3 Труднощі в екстрактивному реферуванні українських текстів

У той час як екстрактивне реферування вважається надійним підходом, унікальні характеристики української мови можуть створити значні проблеми. Із згаданих вище тут фігуруватимуть морфологічна складність мови та варіативність структури речень.

В українській мові слова змінюють форму залежно від своєї граматичної ролі. Ця складність може зробити методи виділення ключових слів менш ефективними, оскільки те саме поняття може бути виражене різними словоформами, що ускладнює ідентифікацію ключових термінів [22]. Крім того, варіативність структури речень, яка забезпечує більшу гнучкість порівняно з англійською, може перешкодити методам оцінювання речень, які значною мірою покладаються на порядок слів. Таким методам може бути важко точно оцінити важливість виключно на основі позиції, потенційно пропускаючи критичну інформацію [23].

Для вирішення цих проблем можна застосувати кілька потенційних стратегій. Визначення основи та лематизація — це прийоми скорочення слів до їх основи або словникової форми (лема). Ці методи можуть покращити вилучення ключових слів, фіксуючи різні варіації однієї концепції, тим самим підвищуючи ефективність інструментів реферування [24]. Крім того, інтеграція синтаксичних функцій, таких як інформація про частину мови, у моделі оцінки речень може значно покращити їхню здатність ідентифікувати ключову інформацію, незалежно від варіацій порядку слів. Такий підхід може допомогти краще вловити нюанси структури українського речення [26]. Використання специфічних для української мови ресурсів, таких як списки стоп-слів і тегери частин мови, може ще більше підвищити точність методів екстрактивного реферування, адаптованих до українського тексту. Ці ресурси можуть

допомогти вдосконалити процес відбору, відфільтрувавши звичайні, але неінформативні слова та точно позначивши граматичні ролі слів.

Визнаючи ці лінгвістичні нюанси та впроваджуючи відповідні рішення, можна побачити потенціал розробки ефективніших екстрактивних системи реферування, спеціально адаптованих для української мови, що підвищать загальну якість і корисність результатів цього процесу.

4. Абстрактивне реферування

Екстрактивне реферування, хоча й поширене, має обмеження. Абстрактивне реферування використовує більш сміливий підхід, спрямований на те, щоб охопити сутність оригінального тексту та створити стислу нову версію з використанням інших слів і фраз.

4.1 Принципи абстрактивного реферування

Абстрактивне реферування виходить за межі простого відбору речень, прагнучи зрозуміти глибинний зміст тексту. Цей підхід використовує досягнення із сфери обробки природної мови (NLP). Одним із ключових компонентів є розуміння природної мови (NLU), яке охоплює методи, що дозволяють комп’ютерам обробляти та наближатись до розуміння людської мови. Системи абстрактивного реферування повинні вловлювати основні ідеї, зв’язки між поняттями та загальні настрої, щоб створити точний та послідовний реферат [29]. Крім того, результат їх роботи можна розглядати як форму «перекладу» з оригінального тексту на коротшу версію. Ця перспектива запозичена з досліджень машинного перекладу, де моделі навчаються перекладати текст з однієї мови на іншу, зберігаючи оригінальне значення.

Системи абстрактивного реферування базуються на кількох підходах. Моделі кодера-декодера (encoder-decoder models) є фундаментальними і складаються з двох частин: кодера, який обробляє вихідний текст і фіксує його значення, і декодера, який генерує підсумок на основі цього закодованого представлення. Ці моделі полегшують перетворення складних текстів у стислі реферати, зберігаючи основну інформацію та перефразуючи її належним чином [29]. Механізми уваги (attention mechanisms) ще більше розширюють можливості моделей кодера-декодера. Дозволяючи декодеру зосередитися на певних частинах закодованого вихідного тексту під час процесу генерації, механізми уваги гарантують, що виділений реферат є більш релевантним та інформативним, точніше охоплюючи сутність оригінального тексту [30].

4.2 Загальні методи абстрактивного реферування

Методи абстрактивного реферування загалом поділяються на дві категорії: структурні підходи (Structured based approach) та семантичні підходи (Semantic based approach). Структурні підходи включають різні методи, такі як метод на основі дерева, метод на основі шаблонів, метод на основі онтології, метод на основі головної та основної фрази та метод на основі правил. З іншого боку, семантичні підходи охоплюють такі методи, як мультимодальна семантична модель, метод на основі інформаційних елементів і метод на основі семантичного графа.

Метод на основі дерева використовує дерево залежностей для представлення тексту. Різні алгоритми використовуються для вибору змісту реферату, напр. алгоритм перетину теми або алгоритм, який використовує локальне вирівнювання по парі розібраних речень. Методика використовує або генератор тексту (NLU), або алгоритм для генерації реферату

Метод на основі шаблонів використовує шаблон для представлення всього оброблюваного документа. Лінгвістичні шаблони або правила вилучення

зіставляються, щоб ідентифікувати фрагменти тексту, які будуть найкраще підходити в слоти шаблону. Ці текстові фрагменти вважаються індикаторами змісту реферату.

Метод головної та основної фрази базується на операціях із фразами (вставці та заміні), які мають синтаксично однакове ядро в головному та основному реченнях, щоб переписати головне речення [17].

У методі на основі правил тексти для реферування представлені у вигляді категорій та списку аспектів. Модуль вибору вмісту вибирає найкращі варіанти серед створених правилами вилучення інформації, щоб відповісти одному або більше аспектам категорії. Нарешті, моделі генерації використовуються для генерації речень реферату.

Мультимодальна семантична модель фіксує поняття та зв’язок між ними. Вона будується для представлення вмісту (тексту та зображень) мультимодальних документів. Важливі поняття оцінюються на основі певної міри, і, нарешті, вибрані з них виражаються у вигляді речень, щоб сформувати реферат.

У методі на основі інформаційних елементів зміст резюме генерується з абстрактного представлення вхідних ресурсів, а не з їх речень. Абстрактним представленням є інформаційний елемент, який є найменшим елементом зв’язної інформації в тексті.

Метод на основі семантичного графа має на меті узагальнити документ шляхом створення семантичного графа під назвою Rich Semantic Graph (RSG) для оригінального тексту, скорочення згенерованого семантичного графа, а потім генерування остаточного абстрактного реферату зі скороченого семантичного графа [36].

4.3 Труднощі в абстрактивному реферуванні українських текстів

В цілому, пропонуючи більш складний підхід, абстрактивне реферуванне створює значні проблеми. Однією з головних проблем є вимоги до даних, оскільки для навчання ефективним моделям необхідна величезна кількість високоякісних навчальних даних із паралельними корпусами вхідних текстів та їх відповідними рефератами. Цей дефіцит даних може бути особливо проблематичним для таких мов, як українська, які мають обмежені ресурси [31]. Оцінка якості цього процесу залишається складним завданням, оскільки метрики, які вимірюють схожість з оригінальним текстом, можуть не вловлювати нюанси точної абстракції [32]. Крім того, підтримання зв’язності та плавності є серйозною проблемою, оскільки генерування граматично правильних і плавних рефератів, зберігаючи значення за межами речень, є вирішальним для ефективних систем даного типу [33].

Мовні особливості української мови можуть ще більше ускладнити абстрактивне реферування. Вільний порядок слів в українській мові забезпечує більшу гнучкість у структурі речень порівняно з англійською, що ускладнює моделям точне фіксування значення та зв’язків між поняттями, коли порядок слів стає менш індикативним [23]. Крім того, складна система відмінків в українській мові, яка передає граматичні ролі та відношення, вимагає систем для обробки цих нюансів, щоб забезпечити точне збереження значення під час процесу[22].

Щоб вирішити ці проблеми, можна застосувати кілька потенційних підходів. Використання багатомовних моделей за допомогою попередньо навчених моделей на багатомовних наборах даних може покращити продуктивність абстрактивного реферування для мов з обмеженими ресурсами, таких як українська. Включення предметно-специфічних баз даних може підвищити здатність моделі генерувати зведені та релевантні підсумки, особливо в таких спеціалізованих областях, як юридичні документи чи новинні статті [34]. Крім того, інтеграція граматичних обмежень у процес декодування може допомогти

переконатися, що згенеровані реферати є граматично правильними та відповідають правилам української мови [35].

5. Порівняльний аналіз

5.1 Сильні та слабкі сторони

Екстрактивне реферування має кілька сильних сторін. Однією з його головних є простота та ефективність; ці алгоритми, як правило, простіші у реалізації та менш дорогі в обчислювальному плані порівняно з абстрактивними методами. Крім того, витягнені реферати часто легше інтерпретувати, оскільки вони складаються з реальних речень із вихідного тексту, що дозволяє користувачам безпосередньо перевіряти інформацію. Також екстрактивні методи гарантують фактичність, знову ж, оскільки вони спираються на існуючі речення, що робить їх менш схильними до введення фактичних помилок порівняно з абстрактивними підходами, які створюють новий текст.

Однак екстрактивне реферування також має помітні недоліки. Воно, як правило, обмежено у творчості, що часто призводить до повторюваних і менш плавних результатів, оскільки програма просто обирає існуючі речення без змін. Таким системам може бути важко вловити нюанси та зв’язки між ідеями, особливо в довших або більш комплексних текстах, що ускладнює ефективну передачу складних ідей . Крім того, витягнуте резюме сприйнятливе до шуму, оскільки нерелевантні або погано сформульовані речення у вхідному тексті можуть негативно вплинути на якість реферату.

З іншого боку, абстрактивне реферування має власний набір сильних сторін. Однією з ключових сильних сторін цього методу є його здатність створювати більш точні реферати. Вони є більш стислими та інформативними, охоплюють суть оригінального тексту, залишаючись плавними та послідовними. Абстрактивні методи також можуть ефективніше обробляти складні тексти,

згущуючи інформацію та зберігаючи істотні зв’язки між ідеями. Крім того, таке реферування може покращити читабельність шляхом перефразування складних речень і генерації реферату з більш природним потоком.

Незважаючи на ці переваги, абстрактивний підхід стикається зі значними проблемами. Навчання ефективних моделей потребує великої кількості високоякісних навчальних даних, які можуть бути обмежені. Існують також проблеми з фактичністю, оскільки абстрактні моделі можуть вводити фактичні помилки під час процесу узагальнення, оскільки вони створюють новий текст, який може не повністю відображати вихідний матеріал. Оцінка якості результаті також залишається постійною проблемою.

5.2 Вибір методу в залежності від сценарію

Вибір більш ефективного методу реферування залежить від конкретного контексту та бажаного результату. Для фактичних рефератів коротких текстів, де збереження точності є першим приорітетом, екстративний метод є більш вдалим вибором завдяки його гарантованій фактичності. Цей метод особливо ефективний у сценаріях, коли текст малий, а інформацію потрібно подати точно так, як вона представлена в джерелі.

Навпаки, для генерації стислого та інформативного реферату довших або складніших текстів кращим варіантом є абстрактивний метод. Цей підхід показує себе краще у вловленні складних ідей і покращенні читабельності шляхом перефразування змісту. Він особливо корисний, коли мета полягає в тому, щоб стиснути велику інформацію, зберігаючи зв’язки між поняттями та гарантуючи, що резюме залишається привабливим і легким для читання.

У ситуаціях з обмеженими навчальними даними або коли можливість інтерпретації має вирішальне значення, можна віддати перевагу екстрактивному підходу. Його простота та залежність від існуючих речень роблять його більш

здійсненним у середовищах з обмеженими ресурсами та полегшують користувачам перевірку реферату. Цей підхід є ідеальним, коли наявних ресурсів недостатньо для навчання складних моделей або коли необхідні ясність і пряма перевірка.

Є також випадки, коли поєднання обох підходів може бути корисним. Змішаний метод може передбачати використання екстрактивних методів для початкового вибору ключових речень з подальшим абстрактивним реферуванням для уточнення та перефразування цих речень для кращої плавності та стислості. Цей комбінований підхід використовує сильні сторони обох методів, забезпечуючи фактичну точність і водночас покращуючи читабельність і узгодженість.

Розуміючи сильні та слабкі сторони кожного підходу та враховуючи конкретні потреби завдання узагальнення ми маємо можливість можуть обрати найбільш підходящий метод або вивчити гібридні підходи для досягнення оптимальних результатів.

6. Майбутні перспективи реферування українського тексту

6.1 Потенційні рішення проблем та сфери для вдосконалення

Щоб підвищити ефективність згаданих методів реферування для української мови, можна розглянути кілька потенційних рішень і областей для вдосконалення.

По-перше, використання специфічних для української мовних ресурсів, таких як списки стоп-слів, тегери частин мови та попередньо натренованих мовних моделей, може значно підвищити ефективність систем. Ці базові інструменти, адаптовані до нюансів української мови, можуть підвищити точність і релевантність створених рефератів [13, 14].

По-друге, використання методів морфологічного аналізу, таких як стемінг і лемматизація, може вирішити проблеми, пов’язані з морфологічною складністю української мови. Зводячи слова до їх базових форм, ці методи можуть покращити вилучення ключових слів і загальну точність вихідного продукту. Такий підхід може допомогти вловити основне значення слів незалежно від їх відмінюваних форм, що є вирішальним для ефективного резюмування [25].

Крім того, використання попередньо навчених моделей на багатомовних наборах даних або включення предметно-специфічних знань може значно покращити продуктивність абстрактивного підсумовування, особливо в сценаріях, де навчальні дані обмежені. Багатомовні моделі можуть перенести навчання з мов із великим ресурсом на українську, тоді як предметно-орієнтовані моделі можуть підвищити релевантність і узгодженість реферату у спеціалізованих галузях [34, 37].

Іншим важливим напрямком для вдосконалення є розробка метрик оцінювання спеціально для української. Поточні метрики оцінювання можуть не повністю охоплювати унікальні лінгвістичні особливості мови, особливо за використання абстрактивного підходу. Розробка критеріїв, які враховують ці нюанси, може забезпечити більш точні та значущі оцінки якості, тим самим допомагаючи вдосконаленню систем реферування [38].

6.3 Потенційні напрямки майбутнього розвитку

Описувана галузь має величезний потенціал для майбутніх досліджень і розробок.

Системи «Людина в циклі» (Human-in-the-Loop): вивчення гібридних систем, які поєднують автоматизовані методи реферування з людським наглядом і вдосконаленням, може ефективно вирішити проблему достовірності фактів та нюансів у абстрактивному реферуванні українською. Такі системи можуть

використовувати сильні сторони автоматизованих процесів, одночасно враховуючи людське судження для забезпечення точності та повноти викладення [39]. Цей об’єднаний підхід може бути особливо корисним у сценаріях, коли тонкість і складність вмісту вимагають рівня розуміння, якого існуючі автоматизовані системи ще не повністю досягають.

Реферування для конкретних галузей: розробка моделей реферування для певних галузей, таких як юридичні документи, наукові публікації чи новинні статті, може задовольнити унікальні потреби різних груп користувачів. Предметно-орієнтовані моделі можуть використовувати спеціалізовані словники та контекстні знання для створення більш релевантних та інформативних рефератів. Цей підхід може підвищити застосовність інструментів реферування в різних професійних і академічних сферах, зробивши їх більш корисними.

Українські інструменти NLP з відкритим кодом: сприяння розвитку та доступності інструментів NLP з відкритим кодом, спеціально розроблених для української мови, може сприяти подальшим дослідженням та інноваціям у цьому напрямку. Інструменти з відкритим вихідним кодом можуть забезпечити фундаментальну структуру, на яку можуть спиратися дослідники та розробники, полегшуючи співпрацю та прискорюючи прогрес у реферуванні українських текстів [40]. Ці інструменти також можуть допомогти подолати обмеженість ресурсів, зробивши складні можливості NLP широко доступними.

Зосередившись на конкретних напрямках, галузь реферування українських текстів може побудувати більш стійку базу, застосовуючи яку можна рухатись до розробки більш складних, точних і зручних систем реферування.

7. Підсумок

Теоретичне дослідження автоматичного реферування українських текстів за допомогою як екстрактивних, так і абстрактивних методів виявляє складний

ландшафт викликів і можливостей. Екстрактивне реферування, завдяки своїй простоті, ефективності та гарантованій точності, слугує надійною основою для створення рефератів, особливо в умовах обмежених ресурсів. Однак воно не завжди в змозі передати складні ідеї та зберегти плавність викладу, що підкреслює необхідність подальших вдосконалень у цій галузі.

З іншого боку, абстрактивне реферування пропонує можливість створення більш стислих, інформативних і читабельних рефератів, переказуючи зміст і захоплюючи глибші семантичні значення. Проте цей підхід ускладнюється високими вимогами до даних, проблемами з фактичною точністю та необхідністю розробки складних оціночних метрик, адаптованих для української мови.

Інтеграція україномовних ресурсів, технік морфологічного аналізу та розробка багатомовних і орієнтованих на конкретну галузь моделей можуть значно покращити ефективність обох підходів до реферування. Крім того, створення оціночних метрик, які точно відображають нюанси української мови, є необхідним для оцінки та покращення якості реферування.

Майбутні дослідження могли б зосередитися на поєднанні сильних сторін екстрактивних і абстрактивних методів за допомогою гібридних підходів, враховуючи граматичні та стилістичні особливості, а також використовуючи системи з участю людини для підвищення точності. Сприяння розвитку відкритих інструментів NLP, спеціально розроблених для української мови, також сприятиме дослідженням та інноваціям, спрямовуючи цю галузь до більш досконалих і зручних для користувачів рішень для реферування.

Практичне порівняння методів

Створення програми

Для написання програми було обрано мову програмування Python версії 3.9 [27].

Використані бібліотеки

NLTK 3.8.1 [42]

NLTK (Natural Language Toolkit) - це популярний набір бібліотек і програм для символьної та статистичної обробки природної мови (NLP) на мові програмування Python. Він розроблений для підтримки досліджень і викладання навчальних курсів, пов'язаних з NLP та суміжними областями.

Встановити бібліотеку можна за допомогою команди:

```
pip install nltk
```

NLTK пропонує широкий спектр можливостей для роботи з текстовими даними, включаючи: токенізацію (розбиття тексту на окремі одиниці, такі як слова або речення), стеммінг та лемматизацію (зведення слів до їх базової форми), частотний аналіз (підрахунок кількості поява слів або фраз), визначення частин мови, синтаксичний аналіз (аналіз структури речення), семантичний аналіз (визначення значення слів і фраз), аналіз настроїв (визначення емоційного тону тексту), класифікацію тексту (автоматичне присвоєння категорій текстовим документам).

З цієї бібліотеки використано команду `nltk.download("punkt")`, що завантажує модель токенізатора Punkt.

Токенізатор Punkt - це інструмент, який використовується для розбиття тексту на окремі одиниці, такі як слова або речення.

Punkt використовує некерований метод навчання, щоб побудувати модель для скорочень, словосполучень та слів на початку речення. Він також включає розділові знаки після речень (з NLTK 3.0 і надалі).

Хоча nltk.download("punkt") безпосередньо не сегментує речення у створеній нами програмі, вона налаштовує основні функції, необхідні для інших частин коду, які можуть вимагати цього. А саме, для бібліотеки sumy (див. нижче), яка використовується для екстрактивного реферування, і внутрішньо покладається на сегментацію речень під час процесу. Завантаживши Punkt, ми гарантуємо, що sumy має необхідні інструменти для розбиття тексту на речення перед початком роботи.

PyTorch 2.3 [43]

PyTorch - це бібліотека з відкритим кодом для глибокого навчання. Її гнучкість, простота використання та потужні можливості роблять її одним із найпопулярніших інструментів для досліджень і розробки в галузі штучного інтелекту.

Встановити бібліотеку можна за допомогою команди:

pip install torch

Вона використовує динамічні обчислення графів, має інтуїтивно зрозумілий інтерфейс та обширну документацію, може бути оптимізована для роботи на GPU, пропонує широкий спектр модулів та інструментів для різних завдань глибокого навчання.

У процесі дослідження було проведено порівняння між PyTorch та TensorFlow, ще однією популярною бібліотекою глибокого навчання. PyTorch було обрано з

орієнтації на гнучкість, простоту використання та швидкість. TensorFlow, на нашу думку може бути кращим вибором для масштабних проектів, яким потрібна зріла бібліотека з широкою промисловою підтримкою або для розгортання своїх моделей, наприклад, на мобільних платформах.

У фінальній версії коду бібліотека PyTorch не отримала масштабного застосування, її використано для перевірки доступності на комп'ютерному пристрої CUDA (Compute Unified Device Architecture) [46]. Це програмно-апаратна архітектура паралельних обчислень, розроблена компанією NVIDIA[47]. Вона дозволяє значно збільшити обчислювальну продуктивність завдяки використанню графічних процесорів (GPU) NVIDIA. Такої перевірки вимагає функція абстрактивного реферування. Також бібліотека забезпечує коректну роботу моделі Pegasus.

sumy 0.11.0 [44]

Sumy - це бібліотека з відкритим кодом для автоматичного реферування тексту, написана мовою Python. Вона надає набір інструментів та алгоритмів для створення рефератів.

Встановити бібліотеку можна за допомогою команди:

```
pip install sumy
```

Sumy пропонує кілька методів, таких як TextRank, Luhn та LexRank, які ґрунтуються на різних принципах. Вона дозволяє налаштовувати багато параметрів методів узагальнення, таких як довжина резюме, важливість речень та ключові слова. Також її можна легко інтегрувати з іншими бібліотеками NLP, такими як NLTK.

Для екстрактивного реферування у створеній програмі з бібліотеки sumy було використано Tokenizer з sumy.nlp.tokenizers, PlaintextParser з sumy.parsers.plaintext та TextRankSummarizer з sumy.summarizers.text_rank.

transformers 4.41.2 [45]

Бібліотека Transformers, розроблена компанією Hugging Face, є потужним інструментом для роботи з трансформерними моделями, які стали основою сучасних систем обробки природної мови. Ці моделі, включаючи BERT, GPT, T5 та інші, демонструють чудові результати в різноманітних завданнях NLP, таких як машинний переклад, системи питання-відповіді, аналіз тональності та генерація тексту. Бібліотека Transformers надає зручний інтерфейс для використання, навчання та донавчання трансформерних моделей.

Встановити бібліотеку можна за допомогою команди:

```
pip install transformers
```

У коді використовується PegasusTokenizer та PegasusForConditionalGeneration для абстрактивного реферування. PegasusForConditionalGeneration, як клас, наслідує аналогічний з моделі Bart та функціонує як модуль PyTorch.

Програмний код

Програма на вхід потребує файл sample_ua.txt[49] у папці проекту, файл має містити текст для реферування.

```
import nltk

import torch

from deep_translator import GoogleTranslator

from sumy.parsers.plaintext import PlaintextParser
```

```
from sumy.summarizers.text_rank import TextRankSummarizer
from sumy.nlp.tokenizers import Tokenizer
from transformers import PegasusForConditionalGeneration, PegasusTokenizer
import os
```

Імпортуємо необхідні бібліотеки.

```
TRANSLATE_LIMIT = 4500
TRANSTATE_USE_CACHE = True
```

Задаємо константи. TRANSLATE_LIMIT: Максимальна кількість символів для одного блоку тексту, який можна передати в сервіс перекладу. TRANSTATE_USE_CACHE: Вказує, чи використовувати кеш для перекладу тексту. Якщо True, програма зчитує перекладений текст з файлу замість повторного перекладу. Необхідність наявності кешу зумовлена лімітом на кількість перекладеного матеріалу у безкоштовній версії API Google Translate.

```
def clear() -> None:
    os.system("cls" if os.name == "nt" else "clear")
```

Ця функція виконує системну команду для очищення екрану консолі. В залежності від операційної системи, виконується або команда cls (для Windows), або clear (для Unix-подібних систем).

```
def read_file_text() -> str:
    file = open("sample_ua.txt", "r", encoding="utf-8")
    return file.read()
```

Відкриває файл sample_ua.txt у режимі читання з кодуванням UTF-8 та повертає його вміст як рядок.

```
def main() -> None:
    nltk.download("punkt")
```

Тут розпочинається основна функція. Спочатку вона завантажує необхідні ресурси для токенізації тексту.

```
    text = read_file_text()
```

Викликає функцію read_file_text, щоб отримати вміст файлу sample_ua.txt.

```
    text_summary = perform_extractive_summarization(text)
```

Виконує екстрактивне реферування тексту за допомогою perform_extractive_summarization.

```
    text_en = perform_translate(
        text, "uk", "en", operation_cache_file="text_translate_ua_en.txt"
    )
```

Перекладає текст з української на англійську за допомогою perform_translate.

```
    abstract_summary_en = perform_abstractive_summarization_pegasus(text_en)
```

Виконує абстрактивне реферування перекладеного тексту за допомогою perform_abstractive_summarization_pegasus.

```
    abstract_summary_ua = perform_translate(
        abstract_summary_en,
        "en",
```

```
        "ua",
        operation_cache_file="abstract_summary_translate_en_ua.txt",
    )
```

Перекладає абстрактивний реферат з англійської мови на українську. Вказує, що результати перекладу повинні бути збережені або прочитані з файлу кешу.

```
    clear()
    print("================================")
    print("======== Extract Summary ========")
    print("================================")
    print(text_summary)
    print()


    print("================================")
    print("======== Abstract Summary ========")
    print("================================")
    print(abstract_summary_ua)
    print()
```

Очищує екран консолі та виводить результати екстрактивного та абстрактивного реферування.

```
def perform_extractive_summarization(text: str) -> str:
```

```
    parser = PlaintextParser.from_string(text, Tokenizer("ukrainian"))
```

Наступна функція - екстрактивне реферування. Використовує PlaintextParser та токенізатор для української мови, щоб розбити текст на речення.

```
    summarizer = TextRankSummarizer()
```

Створює об'єкт TextRankSummarizer для реферування тексту.

```
    summary = summarizer(parser.document, sentences_count=7)
```

Виконує реферування тексту, зберігаючи 7 найбільш значущих речень.

```
    return ". ".join(str(sentence) for sentence in summary)
```

Об'єднує речення в один рядок, розділяючи їх крапками.

```
def perform_translate(
    text: str, source: str, target: str, *, operation_cache_file: str
) -> str:
    if TRANSTATE_USE_CACHE:
        return open(operation_cache_file, "r", encoding="utf-8").read()
```

Наступна функція виконує переклад з української на англійську, це рішення обумовлено відсутністю моделі здатної створити адекватний реферат українського тексту серед загальнодоступних. Отже вона перевіряє чи існує вже файл з перекладом і, якщо так, зчитує переклад з файлу.

```
    sentences = nltk.tokenize.sent_tokenize(text)
    chunk = []
    for sentence in sentences:
```

```
            chunk1 = chunk.copy()
            chunk1.append(sentence)
            block = " ".join(chunk1)

            if len(block) <= TRANSLATE_LIMIT:
                chunk = chunk1
            else:
                blocks.append(" ".join(chunk))
                chunk = []
```

Розбиває текст на блоки, щоб кожен не перевищував ліміт символів для перекладу.

```
            output = []
            for block in blocks:
                translated = GoogleTranslator(source, target).translate(block)
                output.append(translated)
```

Перекладає кожен блок за допомогою GoogleTranslator та зберігає перекладені блоки в список.

```
            return " ".join(output)
```

Об'єднує перекладені блоки в один рядок.

```
    def perform_abstractive_summarization_pegasus(text: str) -> str:
```

```
is_cuda = torch.cuda.is_available()

device = "cuda" if is_cuda else "cpu"

print(f"Processing on {device.upper()}")
```

Далі описується функція для абстрактивного реферування. Її початок визначає, чи доступний на пристрої GPU, і використовує його, якщо доступний.

```
model_name = "google/pegasus-xsum"

tokenizer = PegasusTokenizer.from_pretrained(model_name)

model = PegasusForConditionalGeneration.from_pretrained(model_name).to(device)
```

Завантажує модель PEGASUS та відповідний токенізатор.

```
batch = tokenizer(text, truncation=True, padding="longest", return_tensors="pt").to(device)
```

Токенізує текст та створює пакет для моделі, враховуючи необхідність заповнення та усічення.

```
translated = model.generate(**batch)
```

Використовує модель для генерації абстрактивного реферату.

```
decoded = tokenizer.batch_decode(translated, skip_special_tokens=True)

return decoded[0]
```

Декодує згенерований текст у читабельний формат.

```
main()
```

Запускає основну функцію, яка виконує весь процес обробки тексту.

Результати роботи програми

Після запуску програми отримуємо наступний результат:

“==========

======== Extract Summary ========

==========

До причин, які роблять її привабливою для перекладознавців та перекладачів-практиків, можна віднести, насамперед, ускладнену інтерпретацію не тільки оказіональних одиниць, а й всього тексту, що їх містить, а також відсутність готових еквівалентів у цільовій мові. У відповідності до мети дослідження передбачає вирішення ряду завдань: - визначити особливості утворення та функціонування авторських оказіоналізмів у рамках художнього твору; - встановити особливості утворення україномовних відповідників авторським оказіональним одиницям з урахуванням структури їх утворення та особливостей функціонування у творі оригіналу; - встановити провідні способи відтворення україномовних перекладацьких відповідників; - визначити провідну стратегію, обрану перекладачами задля відтворення авторських оказіоналізмів українською (одомашнення чи очуження). Так, змальовуючи епізод спіритичного сеансу, автор дещо спотворює узуальні англомовні одиниці для пародіювання санскриту: «Interrogated as to whether life there resembled our experience in the flesh he stated that he had heard from more favoured beings now in the spirit that their abodes were equipped with every modern home comfort such as talafana, alavatar, hatakalda, wataklasat and that the highest adepts were steeped in waves of volupcy of the very purest nature. На відміну від словоскладання, телескопічні одиниці утворюються з морфем або довільних сегментів двох або більшої кількості лексичних одиниць, у результаті чого формується лексема з непрозорим морфемним членуванням [12, с. На другому етапі дослідження

нами здійснено спроби виокремити способи перекладацького відтворення авторських оказіоналізмів та встановити, як їх структурні та функціональні характеристики впливають на відтворення перекладацьких відповідників. Розтлумачити сенс утворення Ticktacktwo wouldyousetashoe можна лише з опорою на контекст, з якого стає зрозумілим, що Блум заколисує немовлят та показує їм всілякі фокуси. У результаті дослідження ми дійшли висновку, що не існує єдиного підходу до відтворення оказіональних одиниць у мові перекладу

==========

======== Abstract Summary ========

==========

Метою цих досліджень є визначення особливостей україномовного відтворення оказіоналізмів іноземних творів.”

Порівняння

Загальні критерії

Для оцінки отриманих рефератів вручну було обрано наступні критерії [50].

Точність змісту. Наскільки правильно реферат передає основні ідеї та факти оригінального тексту. Чи немає спотворень або неточностей у викладенні інформації.

Повнота. Чи включає реферат більшість ключових моментів та аспектів анотації (референсу).

Зрозумілість. Наскільки легко реферат читається і сприймається.

Граматичні та стилістичні помилки. Відсутність граматичних, орфографічних та стилістичних помилок. Відповідність стилю оригінального тексту.

Лексичне багатство. Різноманітність використаних слів і термінів. Відсутність повторень одних і тих самих слів без необхідності.

Синтаксична складність. Різноманітність синтаксичних конструкцій. Використання складних речень у відповідних контекстах.

Когезія. Наскільки логічно і зв'язно побудований текст реферату. Використання засобів зв’язності для об'єднання частин тексту (наприклад, займенники, сполучники, паралельні конструкції). Відсутність розривів у логічних зв'язках між реченнями, неясностей або суперечливих частин.

Когерентність. Логічна послідовність і зв'язність викладення думок. Наявність ясних переходів між ідеями.

Адекватність викладу. Наскільки точно передано смисл оригінального тексту. Відсутність двозначностей і неправильних інтерпретацій.

Семантична точність. Правильне використання слів з точки зору їх значення. Відсутність лексичних помилок, які можуть змінити сенс тексту.

Було обрано трибальну систему оцінки, для зменшення суперечливості результату, де: 1 - низька, 2 - середня, 3 - висока.

Тип реферування	Екстрактивне	Абстрактивне
Точність змісту	3	2
Повнота	2	1
Зрозумілість	1	3
Граматичні та стилістичні помилки	3	3

Лексичне багатство	3	2
Синтаксична складність	2	2
Когезія	1	2
Когерентність	2	1
Адекватність викладу	2	3
Семантична точність	3	3
Сума	22	22

Таблиця 1. Оцінка резюме лінгвістом-експертом

ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation)[48] - це набір метрик, що використовуються для оцінки якості автоматичних текстових резюме шляхом порівняння їх з референсними або написаними людиною зразками резюме. ROUGE розроблено для вимірювання подібності між резюме та вихідним текстом за допомогою різних методів порівняння, таких як уніграми, біграми та найдовша спільна підпослідовність (LCS).

Для автоматичної оцінки резюме було використано наступні метрики. ROUGE-1, що вимірює кількість співпадаючих одиночних слів (уніграм) між системою резюме та референсом. ROUGE-1 оцінює точність збігу слів у резюме зі словами у референсному резюме. Ця метрика важлива для оцінки повноти та правильності вибраних слів.

ROUGE-2 вимірює перекриття послідовних пар слів (біграм) між резюме та референсом. ROUGE-2 враховує більш тісний контекст слів, оцінюючи збіг

послідовностей. Ця метрика допомагає виявити наявність важливих фраз або виразів у резюме.

ROUGE-L вимірює довжину найдовшої спільної послідовності слів між резюме та референсом. Вона оцінює структурну схожість та логічний зв'язок. Ця метрика особливо корисна для оцінки цінності та зв'язності вибраних слів у резюме.

У результаті обчислення метрик ROUGE зазвичай отримуються три значення: R, P та F.

R (Recall) вимірює співпадіння між автоматично згенерованим резюме та референсним резюме відносно всіх важливих елементів, які повинні бути включені. Вище значення R вказує на те, що автоматично згенероване резюме успішно включає у себе більшу кількість важливих елементів з оригінального тексту.

P (Precision) вимірює точність або кількість правильно вибраних елементів у згенерованому резюме відносно всіх вибраних елементів у ньому. Вище значення P вказує на те, що автоматично згенероване резюме містить менше непотрібної інформації або шуму.

F (F1 Score) є гармонічним середнім між Recall та Precision і використовується для об'єднання обох цих метрик в одному значенні. Високе значення F1 вказує на те, що резюме має як високий Recall, так і високу Precision, тобто вміщає в себе багато важливої інформації та містить мало шуму.

Простий варіант коду для застосування ROUGE буде виглядати наступним чином:

```
from rouge import Rouge
```

```
generated_summary = "This is a generated summary of the text."

reference_summaries = [
    "This is a reference summary of the text.",
    "Another reference summary for evaluation."
]

rouge = Rouge()

scores = rouge.get_scores(generated_summary, reference_summaries)

print("ROUGE Scores:")
print(scores)
```

Отримавши результати такого порівняння з референсом ми можемо зіставити їх між собою.

Екстрактивне	
rouge-1	
r	0,078448
p	0,379102
f	0,127767

Абстрактивне	
rouge-1	
r	0,028902
p	0,333333
f	0,053191

rouge-2

rouge-2

r	0,014610
p	0,089851
f	0,024434

r	0,003788
p	0,058824
f	0,007117

rouge-L	
r	0,066887
p	0,323420
f	0,108768

rouge-L	
r	0,028902
p	0,333333
f	0,053191

Таблиця 2. Оцінка резюме ROUGE

Висновок

У процесі дослідження були оцінені два підходи до автоматичного реферування текстів: екстрактивний та абстрактивний. Кожен з цих підходів має свої переваги та недоліки, що відобразилось у результатах оцінювання за різними критеріями.

Екстрактивне реферування, яке полягає у вибірковому копіюванні сегментів тексту з оригіналу, виявилося достатньо точним і граматично правильним, проте мало проблеми з повнотою, зв'язністю і синтаксичною складністю. Незважаючи на високу семантичну точність і лексичне багатство, реферати за цим підходом іноді були важкими для сприйняття через недостатню когезію та зрозумілість. Воно також продемонструвало високу точність у метриках ROUGE, особливо у ROUGE-1 (0,379102) та ROUGE-L (0,323420). Однак значення recall залишаються низькими, що свідчить про те, що значна частина важливих деталей з оригінального тексту не потрапила у реферат. Це також відобразилося на помірних значеннях F-score.

Абстрактивне реферування, яке включає генерацію нових речень на основі змісту оригіналу, виявилось більш зрозумілим і адекватним з точки зору

викладу, проте мало низьку точність змісту і повноту. Хоча текст абстрактивного реферату був граматично правильним і зрозумілим, йому бракувало лексичного багатства та когерентності. Точність (precision) також була високою (0,333333 у ROUGE-1 та ROUGE-L), значення recall були ще нижчими, ніж у екстрактивного підходу, що особливо помітно у ROUGE-2 (0,003788). Це свідчить про ще більшу втрату важливих деталей з оригінального тексту. Відповідно, F-score для абстрактивного реферування також був нижчим, ніж для екстрактивного підходу.

З огляду на результати, можна зробити висновок, що вибір підходу до реферування залежить від конкретних вимог завдання. Якщо важлива точність передачі змісту і семантична точність, то доцільніше використовувати екстрактивне реферування. Якщо ж на перший план виходять зрозумілість і адекватність викладу, то абстрактивне реферування буде кращим вибором.

Для досягнення оптимальних результатів можна розглянути комбінований підхід, що об'єднує переваги обох методів, забезпечуючи як точність і повноту, так і зрозумілість і когезію тексту.

Список використаної літератури

1. Babar, S. A. (2013, October). Text Summarization: An Overview. [Режим доступу] - https://www.researchgate.net/publication/257947528_Text_SummarizationAn_Overview
2. Zhu, C., Liu, Y., Mei, J., & Zeng, M. (2021, March 12). MEDIASUM: A Large-scale Media Interview Dataset for Dialogue Summarization. Microsoft Cognitive Services Research Group. [Режим доступу] - https://arxiv.org/pdf/2103.06410
3. Mridha, M. F., Akter, A., Nur, K., Das, S. C., Hasan, M., & Kabir, M. M. (2021, November). A Survey of Automatic Text Summarization: Progress, Process and Challenges. [Режим доступу] - https://www.researchgate.net/publication/356450175_A_Survey_of_Automatic_Text_Summarization_Progress_Process_and_Challenges
4. Takale, S. A. (2023, August). A Survey of Legal Document Summarization Methods. [Режим доступу] - https://www.researchgate.net/publication/372862017_A_Survey_of_Legal_Document_Summarization_Methods
5. Asa, A. S., Akter, S., Uddin, M. P., Hossain, M. D., Roy, S. K., & Afjal, M. I. (2017). A Comprehensive Survey on Extractive Text Summarization Techniques. [Режим доступу] - https://www.academia.edu/33775488/A_Comprehensive_Survey_on_Extractive_Text_Summarization_Techniques?source=swp_share
6. Gao, S., Chen, X., Li, P., Ren, Z., Bing, L., Zhao, D., & Yan, R. (2018, December 13). Abstractive Text Summarization by Incorporating Reader Comments. [Режим доступу] - https://arxiv.org/pdf/1812.05407
7. Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing. [Режим доступу] - https://web.stanford.edu/~jurafsky/slp3/ed3bookfeb3_2024.pdf

8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). Attention Is All You Need. [Режим доступу] - https://arxiv.org/abs/1706.03762
9. LexRank. [Електронний ресурс] - https://github.com/crabcamp/lexrank
10. Gensim. [Електронний ресурс] - https://pypi.org/project/gensim/
11. BART. [Електронний ресурс] - https://huggingface.co/docs/transformers/model_doc/bart
12. Sumy. [Електронний ресурс] - https://pypi.org/project/sumy/
13. Universal Dependencies. [Електронний ресурс] - https://universaldependencies.org/treebanks/uk_iu/index.html
14. MTP. [Електронний ресурс] - https://github.com/Kyubyong/mtp
15. Bauzha, O., Kramov, A., & Yavorskyi, O. (2023). Estimation of the Factual Correctness of Summaries of a Ukrainian-language Silver Standard Corpus. [Режим доступу] - https://ceur-ws.org/Vol-3646/Paper_1.pdf
16. Galeshchuk, S. (2023). Abstractive Summarization for the Ukrainian Language: Multi-Task Learning with Hromadske.ua News Dataset. [Режим доступу] - https://aclanthology.org/2023.unlp-1.6.pdf
17. Gong, S., Zhu, Z., Qi, J., Tong, C., Lu, Q., & Wu, W. (2022). Improving Extractive Document Summarization with Sentence Centrality. [Режим доступу] - https://www.researchgate.net/publication/362205142_Improving_extractive_document_summarization_with_sentence_centrality
18. Gong, Y., & Liu, X. (2001). Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. [Режим доступу] - https://dl.acm.org/doi/abs/10.1145/383952.383955
19. Литвин, В. В., Шаховська, Н. Б., & Крайовський, В. Я. (2010). Реферування текстових документів на основі зважування міри TF-IDF онтологією предметної галузі. [Режим доступу] -

https://science.lpnu.ua/sites/default/files/journal-paper/2019/apr/16287/vis689ism-294-302.pdf

20. Yadav, D., Katna, R., Yadav, A. K., & Morato, J. (2022). Feature Based Automatic Text Summarization Methods: A Comprehensive State-of-the-Art Survey. [Режим доступу] - https://www.researchgate.net/publication/366479972_Feature_Based_Automatic_Text_Summarization_Methods_A_Comprehensive_State-of-the-Art_Survey
21. Mihalcea, R. (2004). Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization. [Режим доступу] - https://dl.acm.org/doi/pdf/10.3115/1219044.1219064#:~:text=Graph%2Dbased%20ranking%20algorithms%20are,a%20range%20of%20ranking%20problems
22. Andreichuk, N., & Babelyuk, O. (2019). Contrastive Lexicology of English and Ukrainian Languages: Theory and Practice. [Режим доступу] - https://lingua.lnu.edu.ua/wp-content/uploads/2015/03/contrastive-lexicology.pdf
23. Гладуш, Н. Ф., & Павлюк, Н. В. (2019). Contrastive Grammar: Theory and Practice. [Режим доступу] - https://elibrary.kubg.edu.ua/id/eprint/27556/1/N_Gladush_N_Pavliuk_Contrastive%20Grammar.pdf
24. Georgiev, G., Zhikov, V., Osenova, P., & Simov, K. (2012). Feature-Rich Part-of-speech Tagging for Morphologically Complex Languages: Application to Bulgarian. [Режим доступу] - https://aclanthology.org/E12-1050.pdf
25. Voutilainen, A. (2012). The Oxford Handbook of Computational Linguistics, Part-of-Speech Tagging. [Режим доступу] - https://academic.oup.com/edited-volume/34563/chapter-abstract/293282669?redirectedFrom=fulltext&login=false
26. Nadeau, D., & Sekine, S. (2007). A Survey of Named Entity Recognition and Classification. [Режим доступу] -

https://www.semanticscholar.org/paper/A-survey-of-named-entity-recognition-and-Nadeau-Sekine/4a554da55fd9ff76c99e25d2ce937b225dc1100c

27. Python Software Foundation. (2020). Python. [Електронний ресурс] - https://www.python.org/downloads/release/python-390/
28. Bird, S., Loper, E., & Klein, E. (2009). Natural Language Processing with Python. O'Reilly Media Inc.
29. See, A., Liu, P. J., & Manning, C. D. (2017). Get to the Point: Summarization with Pointer-Generator Networks. [Режим доступу] - https://aclanthology.org/P17-1099.pdf
30. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems, 5998–6008. [Режим доступу] - https://www.bibsonomy.org/bibtex/c9bf08cbcb15680c807e12a01dd8c929?lang=en
31. Dong, Y. (2018). A Survey on Neural Network-Based Summarization Methods. [Режим доступу] - https://www.researchgate.net/publication/324492700_A_Survey_on_Neural_Network-Based_Summarization_Methods
32. Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. [Режим доступу] - https://aclanthology.org/W04-1013/
33. Kryściński, W., Paulus, R., Xiong, C., & Socher, R. (2018). Improving Abstraction in Text Summarization. [Режим доступу] - https://www.researchgate.net/publication/334116483_Improving_Abstraction_in_Text_Summarization
34. Foroutan, N., Romanou, A., Massonnet, S., & Lebret, R. (2022). Multilingual Text Summarization on Financial Documents. [Режим доступу] - https://aclanthology.org/2022.fnp-1.7.pdf
35. Jing, H., & McKeown, K. (2000). Cut and Paste Based Text Summarization. [Режим доступу] -

https://www.semanticscholar.org/paper/Cut-and-Paste-Based-Text-Summarization-Jing-McKeown/ba0eac94ff6e5bc956852e21ba08df4827448f59

36. Khan, A. (2014). A Review on Abstractive Summarization Methods. Journal of Theoretical and Applied Information Technology, 59, 64-72. [Режим доступу] - https://www.researchgate.net/publication/287206659_A_Review_on_Abstractive_Summarization_Methods
37. Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., & Zettlemoyer, L. (2020). Multilingual Denoising Pre-training for Neural Machine Translation. Transactions of the Association for Computational Linguistics, 8, 726-742. [Режим доступу] - https://www.researchgate.net/publication/347164819_Multilingual_Denoising_Pre-training_for_Neural_Machine_Translation
38. Wu, Z., Helaoui, R., Recupero, D. R., & Riboni, D. (2023). Towards Effective Automatic Evaluation of Generated Reflections for Motivational Interviewing. In Companion Publication of the 25th International Conference on Multimodal Interaction (ICMI '23 Companion), 368-373. Association for Computing Machinery. [Режим доступу] - https://doi.org/10.1145/3610661.3616127
39. Liu, Q., Chen, Y., Cheng, G., Kharlamov, E., Li, J., & Qu, Y. (2020). Entity Summarization with User Feedback. The Semantic Web, 12123, 376-392. doi: 10.1007/978-3-030-49461-2_22. PMCID: PMC7250598.
40. NLP UK. LanguageTool API. [Електронний ресурс] - https://github.com/brown-uk/nlp_uk
41. Liu, P. J., & Zhao, Y. (2020). PEGASUS: A State-of-the-Art Model for Abstractive Text Summarization. [Електронний ресурс] - https://research.google/blog/pegasus-a-state-of-the-art-model-for-abstractive-text-summarization/
42. nltk. [Електронний ресурс] - https://pypi.org/project/nltk/
43. PyTorch. [Електронний ресурс] - https://pytorch.org/

44. sumy. [Електронний ресурс] - https://pypi.org/project/sumy/0.2.1/
45. transformers. [Електронний ресурс] - https://pypi.org/project/transformers/
46. CUDA. [Електронний ресурс] - https://developer.nvidia.com/cuda-downloads
47. NVIDIA. [Електронний ресурс] - https://www.nvidia.com/en-eu/
48. ROUGE. [Електронний ресурс] - https://hyperskill.org/learn/step/29669
49. Sample_ua.txt. [Електронний ресурс] - https://drive.google.com/file/d/1OrtYyZV14Ugsuy-D_BKuYRiMzjaTzkkH/view?usp=sharing
50. Bean, J. (2023). Holistic Scale for Grading Article Summaries. [Режим доступу] - https://undergradcollege.utexas.edu/sig/essentials/writing/rubrics/article-summary