Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*
(повна назва)

Кафедра *Системотехніки*
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ГЮИК.501310.003 ПЗ
(позначення документу)

Дослідження та розробка методів автоматичного реферування текстового контенту
(тема)

Виконав:
студент 2 курсу, групи *ІТПм-19-1*
Спеціальність *122 – Комп'ютерні науки*
(код і повна назва спеціальності)
Тип програми *освітньо-професійна*
Освітня програма *Інформаційні технології проектування*
(повна назва освітньої програми)
Дорошенко І.К.
(прізвище, ініціали)
Керівник *проф. Іванов В. Г.*
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри СТ _____ *проф. Гребеннік І.В.*
(підпис) (прізвище, ініціали)

2020 р

Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*
(повна назва)
Кафедра *Системотехніки*
(повна назва)
Рівень вищої освіти *другий (магістерський)*
Спеціальність *122 – Комп'ютерні науки*
(код і повна назва)
Тип програми *освітньо-професійна*
Освітня програма *Інформаційні технологої проектування*
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20____ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

Студентові Дорошенко Ірині Костянтинівні
(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження та розробка методів автоматичного реферування текстового контенту
затверджена наказом по університету від « 2 » листопада 2020 р. № 1517 Ст
2. Термін подання студентом роботи (проекту) 18 грудня 2020 р.
3. Вихідні дані до роботи (проекту): *Науково-технічні публікації, дані статей, результати експериментальних досліджень, дані проектів щодо розробки систем автоматичного реферування текстыв.*
4. Зміст пояснювальної записки (перелік питань, що потрібно розробити)
Вступ, 1 Огляд і аналіз сучасного стану проблеми, що розглядається, 1.1 Обробка природної мови, 1.2 Поняття реферування тексту, 1.3 Основні підходи до автоматичного реферування тексту, 1.4 «Добуваючий» підхід автоматичного реферування текстів, 1.4.1 Підходи з використанням тематичного представлення, 1.4.2 Підходи з використанням індикаторного представлення, 1.5 Оцінка якості систем реферування, 1.5.1 KL розбіжність, 1.5.2 JS розбіжність, 1.5.3 Косинусна відстань, 2 Постановка задачі, 3 Модель автоматичного реферування текстів новинних статей, 3.1 Формальний опис методу, 3.2 Вхідна та вихідна інформація до моделі, 4 Програмна реалізація та експериментальний аналіз, 4.1 Вибір інструментів та середовища розробки, 4.2 Система автоматичного реферування текстів новинних статей, 4.2.1 Опис алгоритму, 4.2.2 Опис текстових корпусів, 4.2.3 Наочні результати роботи системи, 4.3 Оцінка роботи системи, Висновки
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів)

5.1 Загальна класифікація типів реферування тексту (1 арк. А4), 5.2 Графічне подання тексту у вигляді зваженого графу (1 арк. А4), 5.3 Схема роботи системи автоматичного реферування текстів новинних статей (1 арк. А4), 5.4 Розподіл частин мови у текстовому корпусі, що представляє модель російської мови (1 арк. А4), 5.5 Розподіл частин мови у текстовому корпусі, що представляє модель англійської мови (1 арк. А4), 5.3 Наочні приклади роботи системи реферування (3 арк. А4)

6. Консультанти розділів роботи (проекту)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спеціальна частина	проф. Іванов В.Г.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на дипломну роботу	*02.11.2020*	*Виконано*
2	Аналіз предметної галузі	*03.11.2020-05.11.2020*	*Виконано*
3	Дослідження і аналіз існуючих підходів автоматичного реферування текстів	*06.11.2020-14.11.2020*	*Виконано*
4	Постановка завдання дослідження	*15.11.2020*	*Виконано*
5	Розробка моделі автоматичного реферування текстів новинних статей	*16.11.2020-25.11.2020*	*Виконано*
6	Розробка підсистеми автоматичного реферування	*26.11.2020-05.12.2020*	*Виконано*
7	Експериментальний аналіз та тестування розробленої моделі (підсистеми)	*06.12.2020-12.12.2020*	*Виконано*
8	Оформлення пояснювальної записки та текстових матеріалів	*13.12.2020-16.12.2020*	*Виконано*
9	Попередній захист	*17.12.2020*	*Виконано*
10	Преставлення роботи в ДЕК	*18.12.2020*	

Дата видачі завдання 2 листопада 2020р.

Студент _____ Дорошенко І.К.
(підпис)

Керівник роботи_____ проф. Іванов В. Г.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської атестаційної роботи: 92 с., 6 табл., 16 рис., 3 додатки, 32 джерела. Графічна частина атестаційної роботи містить: 8 плакатів.

АВТОМАТИЧНЕ РЕФЕРУВАННЯ, КЛЮЧОВІ СЛОВА, РЕФЕРАТ, СИМЕТРИЧНЕ РЕФЕРУВАННЯ, ТЕМАТИЧНА МОДЕЛЬ, EXTRACTIVE SUMMARIZATION, TF-IDF

Об’єктом дослідження є контент веб-сервісів (новинні та тематичні статті), що спеціалізуються на наданні користувачеві доступ до кінцевої інформації.

Предметом дослідження є алгоритми та методи які здатні автоматично скорочувати та підсумовувати тексти на природній мові в рамках «добуваючого» підходу до реферування.

Мета роботи – дослідження існуючих методів в рамках вирішення завдання автоматичного реферування текстів, та розробка на їх основі моделі автоматичного реферування текстів, яка не накладає обмеження на тематику текстів, не потребує складних ресурсомістких обчислень, додаткових наборів документів для навчання, здатна працювати з текстами на різних мовах.

Досягнення мети атестаційної роботи базується на комплексному використанні методів теорії інформаційного пошуку, обробки природної мови та системного підходу.

Результат атестаційної роботи – модель автоматичного реферування новинних статей та підсистема автореферування, що використовує цю модель.

ABSTRACT

Explanatory note: 92 pages, 6 tables, 16 figures, 3 applications, 32 sources. Graphic material 8 p.

ABSTRACT, AUTOMATIC TEXT SUMMARIZATION, EXTRACTIVE SUMMARIZATION, KEY WORDS, SYMMETRIC SUMMARIZATION, TF-IDF, TOPIC REPRESENTATION

The object of the work is the content of web services (news and thematic articles) that specialize in providing the user with access to the information.

The subject of the work is algorithms and methods of automatic text summarization in natural language within the "extractive" approach to abstracting.

The purpose of the work is to investigate existing methods of automatic text summarization and to create on their basis a model of automatic text summarization, which does not impose restrictions on the subject of texts, does not require complex resource-intensive calculations, additional sets of documents for learning, is able to work with texts in different languages.

Methods of development – methods of information research theory, natural language processing and systems approach.

The results of the work – model of automatic news article summarization and the autosummarization system that use this model.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

CRFs – Conditional random fields – випадкові умовні поля;

LCS – Longest Common Subsequence – найбільша загальна підпослідовність;

LSA – Latent Semantic Analysis – латентний семантичний аналіз;

NLP – Natural Language Possesing – обробка природної мови;

ROUGE – Recall-Oriented Oristudy;

TFIDF – Term Frequency Inverse Document Frequency.

ВСТУП

Сучасна людина живе в умовах постійного надлишку інформації, це називають проблемою так званого «інформаційного перевантаження». В зв’язку з настанням нової ери глобалізації та розвитком інформаційних технологій все більше людей змушені підключатися до мережі інтернет, яка в свою чергу дає їм необмежений доступ до поширення та споживання інформації. Міжнародна корпорація даних вважає, що загальний об’єм цифрової інформації, що кожен рік циркулює по усьому світу, зросте до 180 зетабайт до 2025 року. При такій великій кількості інформації, все гостріше стоїть необхідність у використанні алгоритмів, які здатні автоматично скорочувати та сумувати великі за об’ємом дані, щоб вільно передавати їх основний зміст.

Наразі існує необмежена кількість категорій інформації та способів (видів) її вираження, найчастіше людям доводиться маті справу з текстовою інформацією. Для цього існує велика кількість різноманітних джерел: блоги, сайти з новинами, статтями, науковими публікаціями, електронні бібліотеки та довідники, інформаційні канали у різноманітних системах миттєвого обміну повідомленнями. Переробити усю кількість доступної інформації людина не в змозі, так само, як і встановити релевантність вхідного документу без його безпосереднього вивчення. В таких випадках програма автоматичного реферування тексту могла би стати корисним помічником, допомогла би подолати інформаційне перевантаження і швидко прийняти рішення про те, яка інформація вартує подальшого розгляду.

Наприклад, займаючись дослідженнями, фахівець повинен прочитати значну кількість публікацій, як правило, вони забезпечені анотацією і списком ключових слів, але зазвичай цього не достатньо, щоб зрозуміти, чи потрібно вивчати той чи інший документ, або пересічний громадянин, що хоче ознайомитися с останніми новинами, мусить переробити багато несуттєвої

інформації (різноманітні коментарі, доповнення, уточнення, припущення тощо), замість того, щоб прочитати «сухі факти». Ці проблеми покликаний вирішити реферат - вторинний документ, що викладає основний зміст вихідного документа. При цьому ручне анотування — складна, рутинна робота, яка потребує додаткових людських ресурсів, тому створення та використання систем автоматичного реферування текстів є доцільною та актуальною задачею.

На сьогоднішній день існує два основні підходи до автоматичного реферування. Перший підхід орієнтований на знаходженні важливих фрагментів, зазвичай речень, так званий sentence extraction. Другий підхід використовує складні методи семантичного та лінгвістичного аналізу, - summary generation, він генерує реферат на основі семантичного представлення вхідного тексту. На сьогоднішній день науковців більше цікавить другий підхід, але зважаючи на його складність, як в плані реалізації, так і в плані обчислень, а також тому, що він накладає суттєві обмеження на тексти, практичного застосування більше набули методи першого підходу, саме вони використовуються у більшості відкритих сучасних систем автоматичного реферування.

Метою даної роботи є дослідження існуючих методів реферування в рамках «добуваючого» підходу та створення на їх основі системи автоматичного реферування текстів, яка не накладає обмеження на тематику текстів, не потребує складних ресурсомістких обчислень та додаткових наборів документів для навчання, здатна працювати з текстами на різних мовах.

Поставлена мета роботи обумовила наступні завдання дослідження:

– огляд і аналіз існуючих стратегій автоматичного реферування тестів;

– розробка моделі автоматичного реферування текстів, що відповідає вимогам, поставленим у меті роботи;

– розробка підсистеми автоматичного реферування текстів на основі створеної моделі;

– оцінка якості роботи розробленої моделі;

– порівняння результатів роботи розробленої підсистеми з існуючими аналогами.

Об’єктом дослідження є новинні та тематичні статті, які розміщуються у системах, що спеціалізуються на наданні користувачеві доступу до кінцевої інформації.

Предметом дослідження є алгоритми та методи які здатні автоматично скорочувати та підсумовувати тексти на природній мові, надаючи користувачеві у якості результату вторинний документ, що викладає основний зміст вихідного документа.

Методи дослідження. Досягнення мети атестаційної роботи базується на комплексному використанні методів теорії інформаційного пошуку та обробки природної мови, що стосуються створення систем автоматичного реферування текстів.

Основні результати за темою магістерської роботи у вигляді тез доповіді, опубліковано у матеріалах міжнародної студентської наукової конференцій [1].

1 ОГЛЯД І АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ, ЩО РОЗГЛЯДАЄТЬСЯ

1.1 Обробка природної мови

Комп'ютери прекрасно працюють із стандартизованими і структурованими даними, такими як таблиці, бази даних, фінансові звіти. Вони здатні обробляти ці дані набагато швидше, ніж люди. Але люди не спілкуються за допомогою структурованих даних, ми спілкуємося, використовуючи слова – форму неструктурованих даних. Мова і писемність з’явилися тисячі років тому. За цей час людський мозок набув величезного досвіду та створив велику кількість асоціативних правил для розуміння природної мови.

На жаль, комп'ютери погано працюють з неструктурованими даними, тому, що немає стандартних методів їх обробки. В процесі програмування комп'ютеру надається набір правил, за якими він повинен працювати. Для неструктурованих даних ці правила досить абстрактні і їх складно визначити конкретно. Саме тому комп’ютери ще не мають такого ж інтуїтивного розуміння природної мови, як люди.

Підрозділ штучного інтелекту, який націлений на те, щоб дозволити комп’ютерам розуміти і обробляти людську мову називається - Natural Language Possesing [2].

Обробка природної мови (NLP) – це загальний напрямок штучного інтелекту і математичної лінгвістики. Він вивчає проблеми комп'ютерного аналізу і синтезу природних мов. Стосовно до штучного інтелекту аналіз означає розуміння мови, а синтез - генерацію грамотного тексту [3].

Основні, найбільш актуальні питання NLP [3]:

– синтез та розпізнавання мови;

– синтез та аналіз тексту;

– категоризація тексту;
– машинний автоматичний переклад;
– розробка систем запитання-відповідь;
– інформаційний пошук;
– вилучення інформації;
– аналіз тональності тексту;
– реферування тексту.

Розпізнавання мови – процес, що полягає у перетворенні мовного сигналу людського голосу в цифрову інформацію. Зворотна задача до розпізнавання – синтез мови. Це створення на основі друкованого тексту мовних сигналів, тобто штучне виробництво людського мовлення.

Аналіз тексту (Text mining) – процес відшукування та вилучення високоякісної, змістовної інформації з тексту на природній мові для автоматизації процесу аналізу даних.

Під машинним автоматичним перекладом, розуміють процес перекладу текстів, написаних природною мовою, на іншу, також природну, мову.

Системи запитання-відповідь – інформаційні систем, які здатні приймати, розпізнавати, класифікувати питання і давати відповіді на них на природній мові.

Інформаційний пошук – процес виявлення інформації в документах, що містяться в доступних системі пошуку базах даних, які відповідають заданому запиту по тематиці.

Аналіз тональності тексту – оцінка емоційного забарвлення лексем тексту та розподіл їх за належністю до позитивного, негативного або нейтрального лексичного шару мови.

Реферування – скорочення обсягу тексту за рахунок виділення основних тез шляхом пошуку відповідностей заданим в пошуку ключовим словам і його короткий виклад.

У всіх цих випадках головна мета полягає в тому, щоб взяти необроблені дані на природній мові і, використовуючи лінгвістику і алгоритми для

перетворення або збагачення тексту, обробити їх таким чином, щоб вони приносили більшу цінність.

1.2 Поняття реферування тексту

На сьогоднішній день використовують два синонімічні поняття – реферат та анотація, проте на практиці відсутня єдина думка про те, які функції виконує кожне із них. У загальному випадку реферат можна визначити як вторинний документ, що передає основний фактичний зміст первинного документу – на відміну від анотації, яка описує структуру і тематичний зміст вхідного документу. Надалі у роботі при використанні синонімів буде матися на увазі саме реферат.

З лінгвістичної точки зору процес реферування містить принаймні три компоненти: референта – людину, яка виконує реферування; первинний документ – текст, який реферують; та вторинний документ тобто реферат – результат процесу реферування. Структура реферування може бути наглядно представлена у вигляді наступного семантичного трикутника (рисунок 1.1):

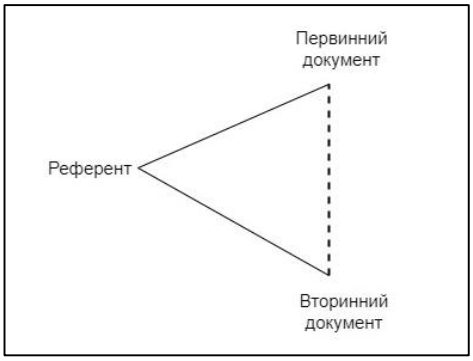


Рисунок 1.1 – Традиційний семантичний трикутник, що представляє структуру реферування

В процесі реферування референту необхідно вивчити структуру першоджерела та використати деякі методи трансформації тексту оригіналу

таким чином, щоб отримати вихідний документ, що відображає зміст первинно документу. Таким чином, референту необхідно знати:

– лінгвістичну структуру первинного документу;

– лінгвістичну структуру вторинного документу;

– методи, за допомогою яких лінгвістична структура первинного документу перетворюється в процесі створення вихідного документу.

У загальному випадку реферування текстів [4] – це процес виділення найбільш важливої інформації з тексту для створення нової скороченою версії документа, виходячи з конкретної мети.

В процесі ручного реферування виявлення основного змісту первинного документу відбувається фактично інтуїтивно, в той час, як в рамках автоматичного реферування основний зміст виявляється на основі встановлених правил, наявності визначених індикаторів чи зафіксованих у словниках лексико-граматичних ознак тощо.

Реферати бувають декількох типів: інформативні, індикативні і критичні. Індикативні реферати повинні надавати достатньо інформації для прийняття рішення, чи варто звертатися до оригіналу. Інформативні реферати повинні скорочувати вихідний текст. Критичні реферати не тільки скорочують, а й дають оцінку тексту.

Приблизну класифікацію типів реферування продемонстровано на рисунку 1.2. Якщо розглядати реферування як функцію, входом якої є документ, а виходом – анотація, в залежності від виду входу та виходу, можна виділии наступні задачі в рамках анотування:

– основні положення будь-якого документу;

– сніпети – невеликі фрагменти вихідного тексту, що містять слова запиту користувача і використовувані пошуковими системами для опису посилань;

– короткий виклад email листування;

– виділення ключових слів. На виході отримуємо не текст, а перелік слів чи фраз із вихідного документу;

– генерація заголовоку документу, замість анотації отримуємо лише один рядок, що максимально чітко описує сенс документу;

– генерація відповідей на складні питання за допомогою короткого змісту декількох документів.

Рисунок 1.2 – Загальна класифікація типів реферування тексту

Для кожного з цих типів реферування існують два важливі параметри:

– кількість документів, що подаються на вхід: один або кілька. У разі систем, які отримують на вхід один документ, як правило, не потрібно упорядкування інформації, відповідно результатом роботи є реферат, що містить речення вхідного документа зі збереженим порядком. В системах, що отримують на вхід кілька документів доводиться вирішувати проблеми упорядкування інформації та вирішення конфліктів;

– тип реферування: загальне реферування або виходячи з запиту користувача. У разі загального реферування не робиться перевага будь-якій інформації з наданого документа. У разі, коли є запит на визначення

важливості деякої частини інформації документа використовуються також слова із запиту користувача. Так можна, наприклад, додати до запиту користувача синоніми всіх слів, і залишати тільки ті речення, що містять слова з цього набору.

Отже саме скорочення обсягу вхідного документу та викладення його основних положень найкраще підійде для подолання проблеми інформаційного перевантаження та вирішення питання, яка інформація вартує подальшого розгляду тому, що саме такий тип реферування дозволяє отримати оцінку важливості інформації на основі всього документу, без надання переваги деяким частинам документа (наприклад, згідно із запитом користувача). Такий підхід чудово лягає в концепцію пересічного користувача який намагається визначити релевантність результатів пошукового запиту (наприклад намагається вирішити, які саме статті за обраною тематикою, будуть найбільш корисні та зможуть відповісти на поставлені запитання виходячи з їх короткого викладу). До того ж, саме загальний тип реферування є найбільш універсальним, адже він не має генерувати відповіді на додаткові питання, чи давати оцінку тексту, отже не потребує додаткових даних для виконання цих завдань.

1.3 Основні підходи до автоматичного реферування тексту

Існує два основних підходи до автоматичного анотування тексту:

– Extractive summarization (добуваючий підхід);
– Abstractive summarization (абстрактний підхід).

Добуваючий підхід – це автоматичне анотування, засноване на виділенні з первинних документів ключових фраз (фрагментів, речень), які додаються в реферат без змін в порядку їх появи в тексті. Цей підхід є досить надійним, оскільки використовує існуючі фрази, які взяті прямо з першоджерела. Але йому не вистачає гнучкості, оскільки метод не здатен використовувати нові слова або словосполучення.

Абстрактний підхід – це автоанотування, засноване на виділенні найбільш суттєвої інформації і генерації нових текстів, змістовно узагальнюючих оригінальні тексти. Він дає можливість використовувати слова, яких не було у вхідному наборі даних. Анотації, згенеровані в рамках такого підходу, мають бути дуже близькими до тих, що люди пишуть власноруч. Але це дуже складна задача, адже модель має вирішувати такі проблеми, як семантичне представлення тексту та генерація природної мови. При цьому така генерація осмислених, зв'язних фраз та речень потребує великих наборів даних для тренування. Також підхід втрачає гнучкість стосовно тематики вхідних даних, адже різні категорії інформації потребують різні набори тренувальних даних, також такий підхід потребує тренування корпусу, щоб бути в змозі працювати з текстами, написаними на різних мовах. Зважаючи на складність даного підходу, як у рамках реалізації так і у рамках обчислень та практичного застосування, а також значні обмеження, що накладаються на тексти, надалі буде проведено дослідження існуючих методів реферування в рамках «добуваючого» підходу.

1.4 «Добуваючий» підхід автоматичного реферування текстів

Зараз ядро всіх «добуваючих» методів складається з трьох незалежних завдань [5]:

– побудова проміжного представлення вхідного тексту. Існує два типи підходів до репрезентації початкового тексту: topic representation (тематичне представлення) та indicator representation (індикаторне, показове представлення). Тематичне представлення трансформує текст в проміжне представлення та інтерпретує теми, що представлені у тексті. Індикаторне представлення описує кожне речення як перелік формальних ознак (індикаторів) важливості, що відрізняються у кожному конкретному алгоритмі і використовує їх для безпосереднього ранжування тексту. Такими ознаками

можуть бути: довжина речення, позиція в документі, наявність певних фраз тощо;

– оцінка речень на основі проміжного представлення. При формуванні проміжного представлення кожному реченню присвоюється оцінка важливості. У підходах до подання теми оцінка речення відображає, наскільки добре речення пояснює деякі найважливіші теми тексту. У поданні індикаторів оцінка розраховується шляхом агрегування даних з різних зважених показників;

– формування підсумку. Система підсумовування вибирає k найважливіших речень для створення анотації. Деякі підходи використовують жадібні алгоритми для вибору важливих речень, а деякі підходи можуть перетворити вибір речень у задачу оптимізації, де вибирається набір речень, враховуючи обмеження, яке повинно максимізувати загальну важливість та мінімізувати надмірність.

1.4.1 Підходи з використанням тематичного представлення

Метод тематичних слів.

Техніка тематичних слів є загальним підходом тематичного представлення, головною ідеєю якого є виявлення слів, що описують теми вхідного документу. Тематичні слова можуть бути визначені різними способами, наприклад робота [6] була однією з перших, в яких використовувався цей метод, пошук описових слів у документі відбувався за рахунок використання порогових значень частоти. У роботі [7] використовували логарифмічний тест відношення правдоподібності для виявлення описових слів. Також інформативні слова можуть бути визначені за допомогою частотно-орієнетованих методів, що будуть розглянуті далі. Визначення важливості речення базується на розподілі інформативних слів у реченнях. Існує два способи обчислити важливість речення: як функцію кількості тематичних слів, яку вона містить, або як частку тематичних слів у

реченні. Обидві функції оцінки речень стосуються одного і того ж подання теми, однак вони можуть призначати різні оцінки реченням. Перший метод може призначати вищі оцінки довшим реченням, оскільки вони мають більше слів. Другий підхід вимірює щільність тематичних слів.

Найперші методи реферування [8] також були засновані на інтуїтивному припущенні про те, що речення є більш важливим, якщо воно містить більш важливі і інформативні слова. Важливість слова визначається виходячи з його частоти в документі, чим частіше воно вживається, тим воно важливіше. Цей метод не використовується безпосередньо з огляду на те, що існують слова з високою частотою, які при цьому не несуть корисної інформації, наприклад частинки і сполучники. Тому використовується один з декількох підходів, які мають загальну назву частотно-орієнетовані. Найбільш поширеними методами в цій категорії є TF-IDF модель, λ-відношення правдоподібності та їх модифікації.

TF-IDF (term frequency — inverse document frequency, частота терму – зворотна частота документа) – статистична міра яка використовується щоб оцінити важливість терма у межах документу, який є частиною набору документів чи текстового корпусу. Оцінка розраховується як добуток функції від кількості появи терма у документі та функції від величини, зворотної до кількості документів у колекції, в яких цей терм зустрічається.

TF (term frequency – частота слова) – відношення кількості появи деякого слова до загальної кількості слів у документі (формула 1.1). Таким чином, оцінюється важливість терма t в рамках окремого документа d.

$$\mathrm{tf}(t, d) = \frac{n_t}{\sum_k n_k}, \tag{1.1}$$

де n_t – кількість входжень терма t в документ;

$\sum_k n_k$ – загальна кількість слів у документі.

IDF (inverse document frequency – зворотна частота документа) – інверсія частоти, з якою деяке слово зустрічається в документах колекції (формула 1.2). IDF дозволяє зменшити оцінку широковживаних слів, та враховувати унікальність слів. Для кожного унікального слова в межах конкретного набору документів існує лише одне значення IDF. Основа логарифму у формулі може бути будь-якою та не має значення, оскільки її зміна призводить до зміни оцінки кожного слова на постійний множник, що не впливає на співвідношення оцінок. Велику оцінку в TF-IDF отримують слова з високою частотою вживань в межах конкретного документа і з низькою частотою вживань в інших документах.

$$\mathrm{idf}(\mathrm{t}, \mathrm{D}) = \log \frac{|\mathrm{D}|}{|\{\mathrm{d_i} \in \mathrm{D} \mid \mathrm{t} \in \mathrm{d_i}\}|}, \tag{1.2}$$

де $|\mathrm{D}|$ – кількість документів у колекції;

$|\{\mathrm{d_i} \in \mathrm{D} \mid \mathrm{t} \in \mathrm{d_i}\}|$ – кількість документів із колекції D, в яких зустрічається t (коли $\mathrm{n_t} \neq 0$).

λ-відношення правдоподібності (log-likelihood ratio) є логарифмом відношення ймовірності спостереження слова з однаковою ймовірністю в корпусі вхідних документів і корпусі відповідних їм резюме, до ймовірності появи слова з різними ймовірностями в цих корпусах. В роботі [9] запропонували використовувати формулу 1.3 для розрахунку ваги слова, а для розрахунку ваги речення – формулу 1.4. В реферат потрапляють речення з більш високою оцінкою.

$$\begin{cases} 1 \; if \; -2 * \log\big(\lambda(w_i)\big) > 10 \\ else \; 0 \end{cases}, \tag{1.3}$$

$$weight(s_i) = \sum_{w \in s_i} \frac{weight(w)}{|\{w | w \in s_i\}|} \tag{1.4}$$

Метод заснований на центруванні речень.

Головна ідея цього підходу заснована на припущенні, що найбільш значуща інформація із першоджерела міститься в окремих реченнях. Він полягає в обчислені відстаней між реченнями і у виборі тих з них, які в середньому знаходяться «ближче» до інших. Для визначення близькості речень зазвичай використовуються алгоритми, засновані на наборах слів, що містяться в реченні (Bag-of-words). Наприклад найближчі в середньому речення можна визначити наступним чином:

– обчислити близькість між усіма парами речень, наприклад за змістовним перекриттям між реченнями, або косинусною близькістю, що використовується в теорії інформаційного пошуку, тобто знайти косинус кута між векторами, що представляють ці речення;

– для кожного речення визначити середню близькість до інших речень;

– впорядкувати і вибрати ті, у яких близькість максимальна.

Латентний семантичний аналіз (LSA).

Тематична модель, що представляється в роботі [10], це неконтрольований метод вилучення прихованої семантики (або ж прихованих тенденцій) тексту на основі спостережуваних слів (базується на гіпотезі розподілу, згідно з якою слова з подібними значеннями часто зустрічаються разом.). Простіше кажучи LSA бере змістовні текстові документи та відтворює їх у n різних частинах, де кожна частина виражає різний спосіб погляду на значення тексту. Якщо уявити текстові дані як ідею, існувало б n різних способів розгляду цієї ідеї або n різних способів концептуалізації всього тексту. LSA зводить весь набір текстових даних до списку прихованих концепцій. У роботі [11] запропонували метод з використанням LSA для вибору речень з високим рейтингом для реферування одного чи декількох документів у сфері новин.

Метод LSA спочатку будує матрицю термін-речення (матриця n на m), де кожен рядок відповідає слову із вхідного документу (n слів), а кожен стовпець відповідає реченню (m речень). Кожен запис a_{ij} матриці є вагою

слова i у реченні j. Вага слів обчислюється методом TF-IDF, якщо в реченні немає слова, вага цього слова в реченні дорівнює нулю. Зазвичай такі матриці є дуже розрідженими і великими за розміром, через велику кількість різних термінів у документах, яка значно перевищує їх щільність, обчислення на таких матрицях обходяться дорого. Для зменшення складності обчислень і отримання більш актуального і корисного результату використовується сингулярна декомпозиція (SVD). SVD розбиває матрицю A на три різні матриці: $A = U\Sigma V^T$ (рисунок 1.3). Матриця U (матриця n × m) представляє матрицю термін-тема з вагами термінів. Матриця Σ (матриця m × m) це діагональна матриця в якій кожен рядок i відповідає вазі теми i. Матриця V^T це матриця тема-речення. Матриця $D = \Sigma V^T$ описує наскільки речення представляє тему, тобто кожен запис матриці d_{ij} показує вагу теми i в реченні j.

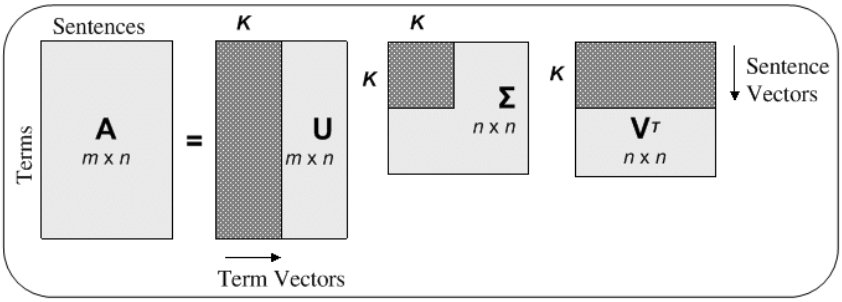


Рисунок 1.3 – Сингулярна декомпозиція вихідної матриці «документ-термін»

Метод у роботі [11] полягає у виборі одного речення для кожної з тем (рисунок 1.4), таким чином виходячи з довжини резюме з точки зору речень, воно зберігає кількість тем. Ця стратегія має недолік через те, що темі може знадобитися більше одного речення для передачі своєї інформації. Отже, були запропоновані альтернативні рішення для поліпшення ефективності методів узагальнення на основі LSA. Одним із покращень було використання ваги кожної теми для вирішення відносного розміру резюме, яке повинно охоплювати тему, що надає гнучкість у варіативності кількості речень.

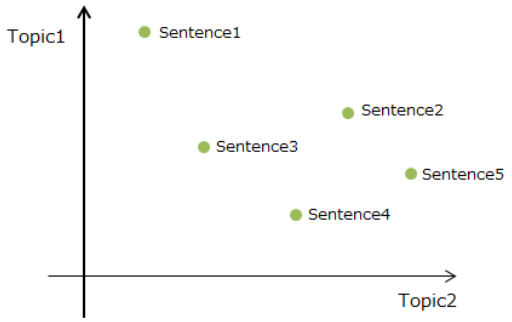
Інше поліпшення базується на тому, що його автори [12], зрозуміли, що речення, які обговорюють деякі важливі теми, є хорошими кандидатами для узагальнення, отже, для того, щоб знайти ці речення, вони визначили вагу речення таким чином (формула 1.5):

$$g(s_i) = \sqrt{\sum_{j=1}^{m} d_{ij}^2}, \tag{1.5}$$

le $g()$ – функція «зважування»;

m – кількість речень;

d_{ij} – вага теми i в реченні j.



	Topic1	Topic2
Sent1	0.9	0.25
Sent2	0.7	0.7
Sent3	0.45	0.4
Sent4	0.3	0.6
Sent5	0.4	0.9

Рисунок 1.4 – Простий вибір основних речень, що представляють теми документу (LSA)

1.4.2 Підходи з використанням індикаторного представлення

Підходи до подання показників (індикаторів) спрямовані на моделювання подання тексту у вигляді набору ознак і використання їх для безпосередньої класифікації речень, замість репрезентації тем вхідного тексту. До них відносяться методи на основі графів та техніки машинного навчання,

які часто використовуються для визначення важливих речень, що повинні бути включені в резюме.

Часто використовуються наступні ознаки:

– розташування в тексті: речення на початку або в кінці тексту є більш інформативними;

– довжина речення: занадто короткі або занадто довгі речення є швидше за все мало інформативними;

– наявність визначених сигнальних фраз;

– наявність слів з заголовка: наявність у реченні слів із заголовка означає, що воно відноситься до даної теми;

– наявність ключових слів;

– наявність емоційно забарвлених розділових знаків;

– інші статистичні класифікатори, які не потребують навчання.

Методи на основі графів, утворені під впливом алгоритму ранжування гіперпосилань PageRank [13], представляють документ у вигляді зв’язаного графу. Речення утворюють вершини графу, а ребра між реченнями вказують, на зв'язок подібності двох речень. Приклад подання тексту у вигляді графа можна побачити на рисунках 1.5 – 1.6. Поширена техніка, що застосовується для з'єднання вершин полягає у вимірюванні подібності двох речень та, якщо вона виявляється більше деякого заданого порогу, встановленні ребра між вершинами, таке ребро має вагу, що відображає міцність зв’язку між вершинами. Подібність речень може бути виміряна як змістовне перекриття між реченнями, або з використанням косинусної подібності з вагою TFIDF для слів.

Графічна інтерпретація тексту призводить до двох результатів. По-перше, розділи (підграфи), включені у граф, представляють дискретні теми, висвітлені в документі. Другий результат – ідентифікація важливих речень у документі, заснована на припущенні, що речення, які пов'язані з багатьма іншими реченнями, є центрами підграфів і, швидше за все, будуть включені в резюме. Методи, засновані на графах, можуть бути використані як для

одиничного, так і для багатодокументального реферування. Оскільки цей метод не вимагає спеціальної мовної обробки, його можна застосовувати до різних мов.

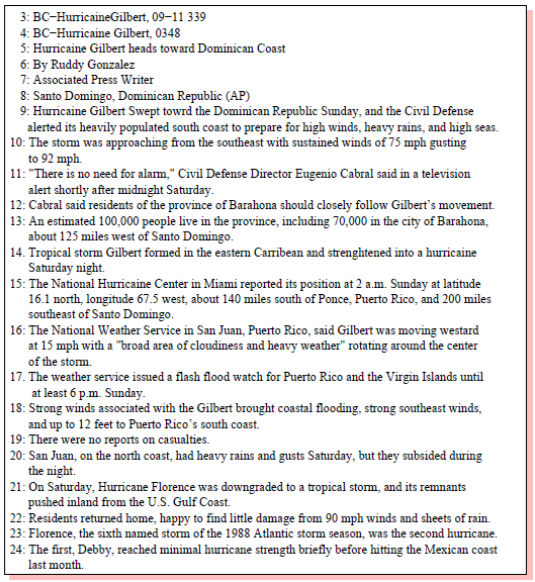

3: BC−HurricaineGilbert, 09−11 339
4: BC−Hurricaine Gilbert, 0348
5: Hurricaine Gilbert heads toward Dominican Coast
6: By Ruddy Gonzalez
7: Associated Press Writer
8: Santo Domingo, Dominican Republic (AP)
9: Hurricaine Gilbert Swept towrd the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains, and high seas.
10: The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.
11: "There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly after midnight Saturday.
12: Cabral said residents of the province of Barahona should closely follow Gilbert's movement.
13: An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.
14. Tropical storm Gilbert formed in the eastern Carribean and strenghtened into a hurricaine Saturday night.
15: The National Hurricaine Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.
16: The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westard at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.
17. The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.
18: Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds, and up to 12 feet to Puerto Rico's south coast.
19: There were no reports on casualties.
20: San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.
21: On Saturday, Hurricane Florence was downgraded to a tropical storm, and its remnants pushed inland from the U.S. Gulf Coast.
22: Residents returned home, happy to find little damage from 90 mph winds and sheets of rain.
23: Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.
24: The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.

Рисунок 1.5 – Приклад тексту для підсумовування

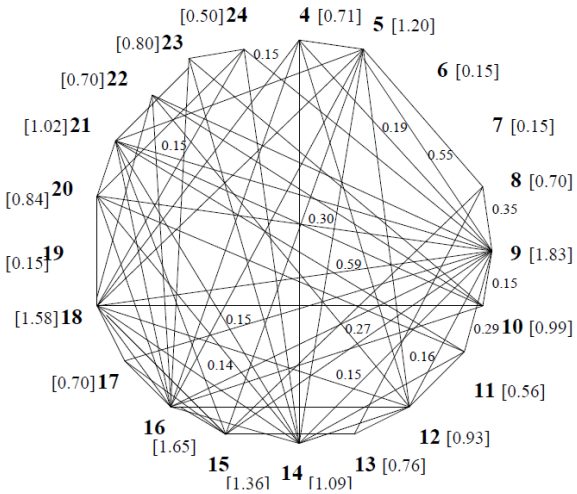


Рисунок 1.6 – Графічне подання тексту у вигляді зваженого графу (речення представляють вершини з встановленою результуючою оцінкою)

Підходи машинного навчання представляють реферування як проблему класифікації. У роботі [14] – представлена рання дослідницька спроба

застосування технік машинного навчання для реферування. Автори розробили функцію класифікації, наївний баєсовський класифікатор, для класифікації речень на результуючі (ті, що потраплять до фінального резюме) та не результуючі на основі ознак (властивостей), які вони мають, застосовуючи при цьому навчальний набір даних з резюме у «добуваючому» стилі. Ймовірності для класифікації вивчаються статистично на основі тренувальних даних з використанням правила Байеса (формула 1.6):

$$\mathrm{P}(s \in S|F_1, F_2, \dots, F_k) = \frac{\mathrm{P}(F_1, F_2, \dots, F_k|s \in S)P(s \in S)}{P(F_1, F_2, \dots, F_k)}, \quad (1.6)$$

де s – це речення із колекції документів;

$F_1, F_2, \dots, F_k$ – ознаки, що використовуються для класифікації;

S – резюме, що має бути створено.

Припускаючи умовну незалежність між ознаками (формула 1.7):

$$\mathrm{P}(s \in S|F_1, F_2, \dots, F_k) = \frac{\mathrm{P}(F_1, F_2, \dots, F_k|s \in S)P(s \in S)}{P(F_1, F_2, \dots, F_k)}, \quad (1.7)$$

Ймовірність того, що речення належить результуючому реферату це його оцінка. Обраний класифікатор гра роль функції оцінки речень.

Інші підходи машинного навчання також широко вживались для автоматичного підсумовування [15 16 17]. Деякі з них, Naive Bayes, decision trees (дерева рішень), support vector machines (машина опорних векторів), Hidden Markov models (приховані марківські ланцюги) and Conditional Random Fields, (умовні випадкові поля) є одними з найбільш загальновживаних методів машинного навчання, що використовуються для підсумовування.

Одна принципова різниця між класифікаторами полягає у тому, що деякі з них вважають, що речення, які обираються для реферату, є незалежними.

Проте виявилося, що методи, які передбачають явну залежність між реченнями такі, як Hidden Markov model [18] і Conditional Random Fields [19] часто перевершують інші техніки. Загалом, методи машинного навчання виявилися дуже ефективними та успішними при узагальненні як окремих так і кількох документів ,зокрема в підсумовуванні по класам, де класифікатори навчені знаходити певний тип інформації.

Отже, незважаючи на різноманітність методів та обгрунтовану дієвість кожного із них, все ж вони мають певні обмеження та недоліки. Наприклад, однією з основних проблем машинного навчання, окрім складності деяких підходів, є те, що для даних алгоритмів необхідна наявність спеціальної навчальної вибірки, тобто набору документів з розміченими реченнями для класифікації (тобто ті які мають потрапити в резюме і ті, які не мають). Задача створення такої вибірки є більш складною, ніж створення рефератів власноруч, які містять об'єднані речення, переформульовані фрази чи нові речення (які використовуються для навчання у методах «абстрактого» підходу), до того ж такі навчальні дані також мають створюватися окремо в залежності від типу інформації чи задачі класифікації, і робота алгоритму повністю залежить від якості таких навчальних даних. Якби вдалося прибрати це обмеження, то можна було б скористатися даними алгоритмами до широкого спектру областей, але на даний момент вони накладають значні обмеження на тексти, проте можуть успішно застосовуватися в підсумовуванні по класам, де класифікатори навчені знаходити певний вузькоспеціалізований тип інформації.

Статистичні підходи до виявлення тематичних слів дуже поширені у використанні завдяки їх простій ідеї виділення найбільш частотних лексем. Наприклад оцінки TF-IDF дозволяють враховувати параметри частотності та унікальності, їх легко і швидко обчислювати, вони є надійною оцінкою щодо визначення важливості речень, тому багато існуючих систем [20 21 22] використовували цю техніку (або його якусь її форму). Проте вони потребують наявності колекції документів або корпусу для підвищення коректності

вилучення тематичних слів, однак відсутність таких корпусів для кожної конкретної предметної області в реальному житті робить застосування таких корпусних моделей і методів вельми проблематичним. При використанні частоти слова в документі в якості головного параметра підрахунок загальної частоти словоформ з парадигми однієї лексеми найчастіше здійснюється наступним чином: загальна частота тематичних слів підраховується шляхом порівняння словоформ, нормалізованих до однієї форми, як правило, до основи або лемми . Автоматична нормалізація словоформи представляє собою завдання лінгвістичного аналізу і досить проблематична. Вона має бути проведена на основі морфологічного, синтаксичного і семантичного аналізів із використанням лінгвістичних баз знань різної глибини (словників, онтологій, граматик, лінгвістичних правилами тощо). Тому такі підходи використовують прості евристичні алгоритми, які нормалізують словоформу до її квазі-основі, найчастіше відсікаючи від словоформи певну кількість букв. Отже можна зробити висновок, що безперечною перевагою статистичних підходів є універсальність алгоритмів виявлення тематичних слів і відсутність необхідності в трудомістких процедурах побудови лінгвістичних баз знань, проте, з іншого боку, без використання таких баз знань такі підходи не знатні враховувати усі особливості мови, що буде відображатися на результатах їх роботи.

Підходи з виявленням тематичних слів представляють оцінку речення як функцію від кількості у ньому знайдених тематичних слів і не враховують інший важливий параметр, що характеризує зв’язність речень у тексті між собою. Припущення, що найбільш значуща інформація із першоджерела міститься в окремих реченнях, використовують графові підходи (найвідоміший представник алгоритм TextRank [13]), або підходи засновані на центруванні речень, вони доводять, що речення, які найбільше зв’язані (найбільш схожі, мають більше посилань на інші речення) з іншими реченнями в тексті виражають його основні думки, вони наче є центроїдами псевдокластерів (або підграфів, якщо мова йде про графові підходи), які

репрезентують вхідний текст. Алгоритми в рамках такого підходу є універсальними, адже не потребують спеціальної мовної обробки чи додаткових даних для зважування термінів чи навчання, але з іншого боку вони враховують лише формальну структуру речень, без урахування семантики, бо речення порівнюються у їх векторному представленні, або за рахунок змістовного (за допомогою n-грам) перекриття. Вирішити цей недолік можна було б, наприклад, за допомогою наявності спеціальних словників тематичної лексики, створених для висвітлення необхідної теми, тоді наявність елементів з такого словника становила б основу зв'язків між реченнями, але в такому разі підхід втратить свою універсальність, адже зможе застосовуватися лише до текстів певної тематики.

Оскільки латентний семантичний аналіз базується на пошуку прихованих концепцій у текстах, він знаходить своє застосування для класифікації або кластеризації документів, але також застосовується для реферування. Проте через те, що він загострює свою увагу на виділенні тем, та оцінює речення щодо відображення тієї чи іншої теми, він буде більш придатний до незагального реферування, коли відомо яким темам потрібно віддати перевагу (важливими темами в такому випадку можна назвати, наприклад, слова із запиту користувача) і з урахуванням цього знання вибрати речення, що найкраще репрезентують обрані теми. У разі висвітлення кожної прихованої теми, необхідно додатково розраховувати їх важливість у рамках всього тексту, а також вирішувати, яка кількість речень необхідна для висвітлення кожної теми, у такому разі підхід втрачає легкість у регулюванні довжини реферату. До того ж, якщо проводиться реферування у межах одного документу, який має незначне розсіювання підтем, такий підхід можна замінити на звичайні статистичні підходи, які є більш простими в пані реалізації та з розрахункової точки зору.

1.5 Оцінка якості систем реферування

Традиційний підхід для оцінки якості реферату – порівняння з еталонним людським рефератом Таке тестування може бути проведене в ручну, за допомогою експертів, які виставляють рефератам бали в залежності від їх якості, або автоматично. Тестування систем вручну вимагає багато часу, зусиль і додаткових людських ресурсів, отже, дані підходи вкрай неефективні. Найрозповсюдженішим інструментом автоматичної оцінки якості реферування є система тестування ROUGE, що була представлена у роботі [23], вона також заснований на ідеї порівняння із людським рефератом. Дана система містить багато метрик: "Rouge-N", "Rouge-L", "Rouge-W", "Rouge-S", "Rouge-SU". Всі метрики засновані на ідеї максимального покриття тестованими резюме модельних, при цьому у всіх методах використовуються N-грами.

Проте, як зазначалося раніше, наявність еталонних рефератів не завжди можлива, адже їх створення довгий ресурсозатратний процес, а знайти зразки рефератів в у вільному доступі не так просто. Тому в роботі [24] були представлені, обґрунтовані та протестовані, повністю автоматичні системи оцінювання, що спираються лише на першоджерело і реферат. Розглянемо детальніше ці системи оцінювання.

У роботі[24] застосовуються міри подібностей між двома розподілами ймовірностей для оцінки якості систем реферування. Використовуються три загальновідомі такі міри: KL (Kullback Leibler) розходження, JS (Jensen Shannon) розходження та косинусна подібність. Очікується, що хороші реферати характеризуються низькою розбіжністю між розподілами ймовірностей слів у вхідному документі та резюме (за KL та JS відстанями) , а також великою схожістю з першоджерелом (за косинусною відстанню).

1.5.1 KL розбіжність

Kullback Leibler (KL divergence) розбіжність між двома розподілами ймовірностей P і Q дорівнює (формула 1.8):

$$D(P||Q) = \sum_{w} pP\ (w) \log_2 \frac{pP\ (w)}{pQ\ (w)} \tag{1.8}$$

Перший аргумент функціонала (розподіл P) зазвичай інтерпретується як істинний або постулюючий апріорі розподіл, другий (розподіл Q) – як ймовірний (той, що підлягає перевірці). Розподіл Q часто служить наближенням розподілу P. Значення функціоналу можна розуміти як кількість неврахованої інформації розподілу P, якщо Q було використано для наближення P. Дана міра відстані в теорії інформації також інтерпретується як величина втрат інформації при заміні справжнього розподілу P на розподіл Q. У даному випадку випадку два розподіли це значення для слів у вхідному документі та підсумку відповідно. Однак розбіжність KL не симетрична, тобто $D(P||Q) \neq D(Q||P)$. Крім того, розходження невизначене коли $pP\ (w) > 0$, але $pQ\ (w) = 0$. Для подолання проблеми пропонують виконати просте згладжування (формула 1.9):

$$p(w) = \frac{C\ +\ \delta}{N\ +\ \delta * B}, \tag{1.9}$$

де C – кількість слів;

N – кількість токенів.

Значення, що в півтора рази перевищує вхідний словниковий запас, використовувалось у якості оцінки результатів (B) розподілу імовірностей, а

для параметра δ було задано невелике значення 0,0005, щоб уникнути зміщення занадто великої імовірності на невидимі події.

1.5.2 JS розбіжність

Розбіжність Дженсена-Шенона також відома як інформаційний радіус або повне відхилення від середнього. Вона заснована на розходженні Кульбака-Лейблера, з деякими помітними (і корисними) відмінностями, в тому числі тим, що вона симетрична і завжди має кінцеве значення, та формально визначається як (формула 1.10):

$$J(P||Q) = \frac{1}{2}[D(P||A) + D(Q||A)], \quad (1.10)$$

де $A = \frac{P+Q}{2}$ – Є середнім розподілом P і Q.

Таким чином обидві відстані показують, скільки інформації буде втрачено, якщо замінити оригінал рефератом, відповідно, чим ближче показник до нуля, тим вища якість реферування.

1.5.3 Косинусна відстань

Косинусна відстань – це міра схожості між двома ненульовими векторами , яка вимірює косинус кута між ними. Таким чином, це оцінка орієнтації, а не величини: два вектори з однаковою орієнтацією мають схожість косинусів 1 два вектори, орієнтовані під кутом 90 ° відносно один одного, мають схожість 0, а два діаметрально протилежних вектори мають схожість -1, незалежно від їх величини. Косинуса відстань використовується в позитивному просторі, де результат обмежений в інтервалі [0,1]. Вона розраховується наступним чином (формула 1.11):

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1.11)$$

Для того, щоб використовувати косинусну відстань для порівняння документів їх необхідно представити у вигляді векторів. Для такого представлення в теорії інформаційного пошуку існує векторна модель. Документ у векторній моделі розглядається як невпорядкована множина термів. Кількісне значення кожного терма визначається як його «важливість» для ідентифікації даного тексту. Якщо терм не зустрічається в документі, то його вага в цьому документі дорівнює нулю. Всі терми, які зустрічаються в документах колекції, можна впорядкувати. Якщо для деякого документа виписати по порядку ваги всіх термів, включаючи ті, яких немає в цьому документі, отримаємо вектор, який і буде представленням даного документа у векторному просторі. У якості ваги терма можна використовувати TF-IDF оцінку. Таким чином ця оцінка є косинусним перекриттям між tf-idf векторними поданнями вхідного та вихідного документів, і чим ближче показник до 1, тим більше відповідає реферат оригіналу по щільності термів.

2 ПОСТАНОВКА ЗАДАЧІ

У даній роботі буде розглядатися модель інформаційного реферату на основі одного документу, що передбачає загальний тип реферування, тобто виділення основних положень документу, без надання переваги тій чи іншій інформації та без додаткових вимог до реферату (наприклад запит користувача). Передбачається, що реферати будуть створюватися для текстів різних тематик, наприклад новинних чи наукових статей, що написані на різних мовах.

Для побудови інформаційного реферату можуть буди використані як підходи з використанням тематичного так і з використанням індикаторного представлень. З інтуїтивної точки зору краще надати перевагу методам індикаторного представлення, адже вони оцінюють кожне речення на основі деякого набору ознак чи функцій, що дає можливість отримати оцінку важливості інформації представленої у реченнях у рамках усього документу, що і є вирішенням задачі виділення основних положень документу, методи ж тематичного представлення загострюють свою увагу на виділенні тем, висвітлених у тексті, та у оцінці речень, щодо відображення тієї чи іншої теми, теоретично такий підхід більше придатний у випадках, коли справа стосується незагального реферування, проте його також застосовують для вирішення такої задачі, хоча деяким його представникам (наприклад алгоритм LSA) може знадобитися додатковий розрахунок оцінки важливості тем та їх ранжування у рамках документу. Хоча, насправді, деякі представники обох підходів, наприклад, статистичні підходи до виявлення тематичних слів, центрування речень (представники тематичного представлення) чи графові підходи (представники індикаторного представлення), можуть бути використані для побудови інформаційного реферату. Всі представлені методи є досить універсальними, та можуть бути застосовані до різних текстів, тому їх можна використати для вирішення такої задачі, але проблема в тому, що кожен із них

використовує лише на одну з двох характеристик речень (важливих у рамках «добуваючого» реферування), розподіл тематичних слів у реченні (статистичні підходи до виявлення тематичних слів) або наявність зав'язків між реченнями (центрування речень, графові підходи), для розрахунку їх оцінок, що породжує деякі з недоліків цих методів, та ставить під сумнів їх застосування в такому варіанті. До того ж наразі можна знайти велику кількість реалізацій графових підходів чи підходів до виявлення тематичних слів у своїх «класичних» (та їх модифікацій, які не стосуються заявленої проблеми) реалізаціях. Тому задачею даної роботи є створення такої моделі, яка б оцінювала речення спираючись на обидві характеристики.

2.1 Формалізована постановка задачі

В рамках роботи створити «добуваючу» модель реферування, яка відповідає наступним вимогам:

– базується на припущенні, що найбільш значуща інформація із першоджерела міститься в окремих реченнях;

– базується на універсальних (не залежних від конкретної предметної області) алгоритмах зважування речень і термінів;

– враховує і розподіл тематичної лексики, і зв'язки (схожість, близкість) між реченнями для їх оцінки;

– не залежить від конкретної мови, може бути адаптована для різних мов без втручання в глобальну роботу моделі, а тільки за рахунок налаштування її параметрів;

– не потребує лінгвістичної бази знань мови;

– є універсальною відносно тематики вхідних текстів;

– не вимагає заздалегідь побудованого корпусу чи словника тематичної лексики кожної конкретної предметної області;

– дає можливість регулювати розміри реферату.

Оцінку речення у рамках моделі представити у вигляді агрегатора двох різних компонент (формула 2.1):

$$K_1 + K_2, \tag{2.1}$$

де K_1 – компонента, що представляє оцінку речення як функцію від розподілу у ньому тематичної лексики;

K_2 – компонента, що представляє оцінку речення як функцію від сили його зв'язків з іншими реченнями тексту.

Вхідними даними до моделі є текстові дані t_{in} – змістовна частина документу d, що підлягає підсумовуванню, на вихід подаються текстові дані t_{out} – набір речень з t_{in}, що отримали найбільші оцінки за формулою 2.1, розмір тексту t_{out} залежить від значення параметру $size_{t_{out}}$, що встановлюється у відсотковому значенні від $size_{t_{in}}$.

Для розрахунку компоненти K_1, підготувати додаткові дані, для виявлення тематичних слів, у вигляді текстового корпусу. Корпус повинен бути репрезентативним, тобто не обмежуватися однією предметною областю. Помістити до корпусу *C* фрагменти *n* текстів, без додаткової розмітки, притаманної лінгвістичним корпусам для дослідження мов, (під розміткою розуміється лінгвістична чи інша інформація, приписана тим чи іншим відрізкам тексту), лексеми у фрагментах текстів привести до їх квазі-основ. Підготувати такі текстові корпуси для кожної мови з якою має працювати модель. Першочергово реалізувати можливість працювати з російською та англійською мовами. Корпуси помісти до бази даних, з можливістю надавати доступ до читання їх вмісту.

Модель реалізувати програмно. Для цього розробити програмний додаток на одній з високорівневих мов програмування, що реалізує алгоритм роботи представленої моделі. Окрім реалізації самого алгоритму у програмному додатку передбачити інтерфейс для роботи з ним. Інтерфейс

користувача реалізувати на рівні CLI (Command line interface) інтерфейсу. Користувач має вводити до командного рядку нерозмічені (нешаблонізовані) текстові дані, що підлягають реферуванню, в початкову вигляді, тобто зі збереженням початкової синтаксичної та граматичної структури (послідовність викладення, знаки пунктуації, поділ на абзаци тощо), але при цьому без рисунків, таблиць та інших об’єктів окрім «чистого» тексту. В якості вихідного результату по черзі вивести до командної стрічки речення з вхідного документу у початковому вигляді, які отримали найбільші оцінки в процесі реферування.

Оцінити якість реферування розробленої моделі. Для цього підготувати (або знайти готові) набори текстових даних для тестування, що представляють фрагменти різноманітних текстів в вихідному вигляді (зі збереженням початкової структури тексту без додаткової розмітки, що містить лінгвістичну інформацію), дані не мають містити жодних об’єктів окрім текстових. Текстові посилки по черзі пропустити через алгоритм та зберегти результати реферування. Якість отриманих рефератів оцінити за метриками, представленими у розділі 1.5 (косинуна подібність, розбіжність Дженсена-Шенона) та порівняти їх з якістю рефератів, створених базовими (існуючими) алгоритмами на тому самому наборі даних.

Порівняти практичні результати реферування розробленої моделі з рефератами, що створюють існуючі системи. Для цього на вхід власної моделі та обраних систем подати однаковий текстовий фрагмент для реферування, отримані реферати порівняти евристично, шляхом надання експертних оцінок за зміст та якість вихідних рефератів.

У якості предметної області вхідних текстів, розглянути новинні статті та статті з тематичних блогів, бо такі тексти охоплюють найрізноманітніші тематики.

3 МОДЕЛЬ АВТОМАТИЧНОГО РЕФЕРУВАННЯ ТЕКСТІВ НОВИННИХ СТАТТЕЙ

3.1 Формальний опис методу

Згідно вимог поставлених до моделі, вона має базуватися на припущені, що найбільш важлива інформація тексту знаходиться в окремих реченнях, та оцінювати речення з урахуванням і взаємозв'язків між ними і ключової лексики, що міститься в них (формула 2.1), тому основна ідея закладається у тому, щоб об'єднати підходи тематичного представлення, головною ідеєю якого є виявлення слів, що описують теми вхідного документу, обчислення важливості речення при цьому становить функцію кількості тематичних слів, яку воно містить та підходи, які припускають, що найважливіші речення це ті, які в середньому знаходяться «ближче» до інших (є найбільш схожими на інші) і дають таким реченням більші оцінки та агрегувати оцінки цих підходів між собою. Для реалізації цього завдання введемо до моделі спеціальний словник тематичної лексики, яка репрезентує вхідний текст. Такий словник дозволить розрахувати обидві оцінки. Наявність елементів з нього буде становити основу зав'язків між реченнями, що дозволяє оцінити не лише формальну, але й якісну схожість між реченнями. Розподіл тематичної лексики у реченні також можна провести на основі термів зі словника. Оскільки модель має бути універсальною і не залежати від тематики вхідного документу неможливо підгодувати такий словник заздалегідь, тому він має створюватися динамічно для кожного вхідного тексту.

Отже, метод включає наступні основні етапи:

– попередня обробка тексту вхідного документу;
– створення словника ключової лексики;
– зважування речень на основі тематичних слів;
– зважування речень на основі зав'язків з іншими реченнями;

– ранжування речень відповідно до розрахованих оцінок та вибір тих, що мають найбільшу оцінку.

На етапі попередньої обробки вилученого тексту відбувається його стеммінг, видалення стоп-слів, поділ на речення, абзаци та слова.

Стоп-слова (інакше звані шумовими) – це слова, знаки, символи, які самостійно не несуть ніякого змістовного навантаження, але які, тим не менш, вкрай необхідні для нормального сприйняття тексту, його цілісності, читабельності. Ці слова просто ігноруються пошуковими системами при здійсненні ранжування або індексації сайтів, а в нашому випадку їх необхідно видалити для покращення роботи моделі, щоб часто використовувані слова були в основному словами, що відносяться до контексту, а не загальними словами, використовуваними в тексті. Переліки стоп-слів індивідуальні для кожної мови (вони регулярно оновлюються пошуковими системами та бібліотеками для обробки природної мови), тому надати їх повний перелік неможливо. Найчастіше стоп-слова поділяють на 2 групи: загальні та залежні. До загальних відносять частки, вигуки, сполучники, прислівники, займенники, числа від 0 до 9 (однозначні), інші часто вживані службові, самостійні частини мови, символи, знаки пунктуації. До другої групи потрапляють слова, які в ключовому запиті визначаються, як другорядні. Приклад: в запиті «Лев Миколайович Толстой» пошукові системи виділяють основний компонент запиту - «Толстой» і другорядні, тобто залежні стоп-слова, що мають значення тільки поруч з головним ключовим словом, - «Лев», «Миколайович».

Стеммінг – процес знаходження основи слова, тобто незмінної частини слова, яка виражає його лексичне значення. Основа не обов'язково збігається з морфологічним коренем слова. Для алгоритмів реферування, що використовують частоту слів в якості ознаки, ця ознака дає точніші результати, якщо враховувати всі словоформи слова як одне слово. Одним із найпростіших способів стеммінгу є відсікання від слова суфіксів і закінчень, для того, щоб решта слова (його квазі-основа) була однакова для всіх граматичних форм слова. Нормалізація слова такого виду не потребує

проведення морфологічного, синтаксичного і семантичного аналізів, використання лінгвістичних баз та може прекрасно працювати з мовами, які реалізують зміну слів через афікси, тобто такі, як наприклад російська, англійська, німецька, українська тощо.

Оскільки використання словників, створених вручну, накладає значні обмеження на область застосування моделей та вимагає додаткових ресурсів та трудомістких робіт, в даному випадку, такий спосіб необхідно замінити автоматичним, який буде працювати із будь-яким вхідним текстом та не залежати від його тематики. Тобто, необхідна функція, яка отримує на вхід текст для підсумовування, а на вихід подає список значущих для вхідного тексту елементів (слів, словосполучень), які і будуть утворювати словник.

Для виявлення значущих елементів, що репрезентують вхідний текст, застосуємо звичайну для статистичних підходів процедуру зважування термінів, засновану на простій ідеї виявлення найпоширеніших лексем. Зважування термінів це фундаментальний алгоритм, який застосовується у всіх предметних областях, що стосуються автоматичної обробки текстових документів. На вході алгоритму, що виконує зважування, – терміни вхідного документу (або вхідний документ), на виході – список термінів з числовими коефіцієнтами, які відображають значущість термінів для даного текстового документу. У якості алгоритму зважування була обрана TF-IDF модель (формули 1.1 – 1.2), оскільки це дуже поширений спосіб оцінки, він дозволяє врахувати не лише частотність, а й унікальність термів, ці оцінки є надійними, їх легко і швидко обчислювати. Ця модель передбачає інтертекстуальний підхід до зважування, тобто зіставлення розподілу термінів в конкретному документі з їх розподілом в інших документах, бо IDF-частина формули містить колекцію документів, на основі якої відбувається зважування кожного конкретного терміну. Тому для роботи моделі необхідно підготувати текстові корпуси. Корпус – це набір мовних фрагментів, зібраних з відповідністю до чітких мовних критеріїв для використання в якості моделі мови.

Після TF-IDF зважування кожен термін вхідного документу отримує числовий коефіцієнт, в результаті до списку ключових слів, які будуть відібрані у словник, будуть вибрані терміни з вагою вище середньої, але до термінів попередньо будуть додаватися додаткові ваги, якщо вони задовольнятимуть наступним індикаторам:

– наявність терміну у заголовку;
– якщо він відзначений як власне ім’я;
– якщо він знаходиться в першому і останньому реченнях абзаців;
– якщо він зустрічається в питальних і окличних реченнях.

Представлені додаткові індикатори обумовлені обраною предметною областю новинних та тематичних статей та можуть відрізнятися (або бути відсутніми) для інших предметних областей. Наприклад заголовок у таких документах є дуже інформативним, тому припускаємо, що слова з нього претендують на «додаткову важливість». Оскільки абзаци сприяють правильному і швидкому сприйняттю тексту та слугують для угруповання однорідних одиниць викладу, вичерпуючи один з його моментів тематичний, сюжетний, смисловий, перше та заключне речення абзаців у деякому сенсі є «заголовком» та «висновком» його основної думки. Питальні і окличні речення мають додаткове емоційне забарвлення, отже також виділяють деякі важливі думки для тексту.

Після створення словника необхідно виконати процедуру розрахунку важливості речень. Важливість речення буде складатися з його функціональної ваги, тобто такої, що відображає зв’язки між реченнями та ваги ключових слів, що містяться у реченні.

Розрахунок функціональної ваги, буде відбуватися на основі методу симетричного реферування, що представлений у роботах [25 26].

Симетричне зважування характеризується наступними особливостями:

– вагові коефіцієнти розраховуються для речень;
– ваговий коефіцієнт речення визначається за його функціональною вагою, яка дорівнює кількості зав’язків даного речення з іншими реченнями

тексту. Під зв'язком розуміють повторення основ слів (стем) зі словника в інших реченнях тексту. Враховуються зв'язки в даному реченні, як з попередніми, так і з наступними реченнями. Ваговий коефіцієнт речення S_j розраховується за наступною формулою (формула 3.1):

$$WF(S_j) = \sum_{r_j=1}^{m} r_j * r_i \,, \tag{3.1}$$

де r_j – кількість входжень даної стеми в речення S_j;

r_i – кількість входження даної стеми в інше речення даного тексту.

– вагові коефіцієнти нараховуються на основі принципу симетричності: якщо речення X містить n зав'язків з реченням Y, то речення Y має n зав'язків з реченням X.

Наглядно продемонструвати особливості симетричного реферування можна на прикладі аналізу речень на рисунку 2.1, слова, які містяться у словнику виділені жирним шрифтом та підкресленням.

(1) *A sting operation by the **environment**al group **Green-peace** suggests that some **research**ers who dispute main-stream **scient**ific conclusions on **climate** change are willing to conceal the sources of payment for their **research**, even if the money is purported to come from overseas corporations producing oil, gas and **coal**.*
(5) ***Disclos**ure of funding for **scient**ific **research** has been a flash point in the fight over **climate** change, especially in the case of published **scient**ific **research**.*

Рисунок 2.1 – Приклад речень для розрахунку їх функціональної ваги

Стеми environment, coal та Greenpeac, які присутні в (1), відсутні в (5), а disclos використовується в (5), але відсутня в (1), тобто вони не дають жодних зав'язків для розрахунку функціональної ваги. Стема climate використовується

один раз в обох реченнях, що створює один зв'язок. Стема research зустрічається два рази в обох реченнях, що дає чотири зв’язки. Стема scient використовується один раз в (1) та два рази в (5), що дає два зв’язки. Відповідно до формули 2.1: WF(S_1)=1*0+1*0+2*2+1*2+1*1+1*0=7. За принципом симетричності кількість зав’язків речення (5) з реченням (1) також буде дорівнювати 7.

Далі необхідно розрахувати базову вагу речення, тобто провести аналіз розподілу термінів зі словника у реченні, шляхом розрахунку суми ваг всіх ключових термінів, що містяться у реченні S_j (формула 3.2).

$$WB(S_j) = \sum_{n=1}^{m} s_n \,, \tag{3.2}$$

де S_j – поточне речення;

m – кількість ключових слів у реченні;

S_n – вага n-го ключового слова.

Результуюча оцінка речення розраховується шляхом сумування двох оцінок (формула 3.3):

$$W(S_j) = WB(S_j) + WF(S_j), \tag{3.3}$$

де $WF(S_j)$ – оцінка речення від сили його зв’язків з іншими реченнями тексту;

$WF(S_j)$ – оцінка речення від розподілу у ньому тематичної лексики.

Через додавання ваги ключових слів, що містяться у реченні до його функціональної ваги, існує ризик того, що довгі речення завжди будуть мати більш високий рейтинг. Щоб уникнути такої надмірної ваги, оцінку речення необхідно нормалізувати, шляхом множення ваги речення на середню

довжину речення у тексті і діленням на кількість слів у реченні. Результуючу оцінку речення можна розрахувати наступним чином (формула 3.4):

$$Final\ sentence\ weight\ = \frac{\frac{WC}{SC} * W(S)}{sentence\ length}, \tag{3.4}$$

де WC – кількість слів у тексті;

SC – кількість речень у тексті;

$sentence\ length$ – кількість слів у реченні.

На завершальному етапі модель обирає n перших речень зі списку, відсортованого у порядку спадання оцінки речень. Кількість речень, які будуть використовуватися в результуючому резюме, може залежати від потреб користувача. За замовчуванням ступінь стиснення складає 20% від усіх речень в списку тому, що прийнято вважати, що реферати розміром 17%-20% є найкращими та прискорюють процес прийняття рішень. Далі вибрані речення сортуються у порядку їх появи в початкову тексті для збереження логічної зв’язності реферату.

3.2 Вхідна та вихідна інформація до моделі

Вхідною інформацією до моделі є:

– документ, що підлягає реферуванню;
– розмір вихідного реферату;
– список стоп-слів;
– текстовий корпус для зважування термінів;
– індикатори для зважування термінів.

Перелік та опис вхідних повідомлень наведено в таблиці 3.1

Вихідною інформацією є:

– tf-idf вага термінів вхідного документу;

– фінальна вага термінів вхідного документу (з додаванням індикаторів);

– середня вага термінів вхідного документу;

– список термінів, що утворюють словник, з їх вагою;

– функціональна вага речень;

– базова вага речень;

– кількість речень у тексті;

– кількість слів у тексті;

– кількість слів у поточному реченні;

– фінальна вага речень;

– відсортований список зважених речень;

– список речень із вхідного документу, що входять до реферату.

Перелік та опис вихідних повідомлень наведено в таблиці 3.2

Таблиця 3.1 – Перелік та опис вхідних повідомлень;

Ідентифікатор	Опис	Форма представлення	Термін і частота видачі
t_{in}	Текстові дані які представляють собою змістовну частину документу, що підлягає підсумовуванню	Дані, що надає користувач у певному форматі	Кожного разу при використанні підсистеми автоматного реферування текстів
C	Текстовий корпус для зважування термінів	Дані, підготовані заздалегідь, що зберігаються у БД	Кожного разу при використанні підсистеми автоматного реферування текстів

Продовження таблиці 3.1

Ідентифікатор	Опис	Форма представлення	Термін і частота видачі
$l_{st.w}$	Список стоп-слів, що використовуються на етапі попередньої обробки, для очищення даних t_{in}	Дані, підготовані заздалегідь, що зберігаються на енергонезалежному носії	Кожного разу при використанні підсистеми автоматного реферування текстів
$size_{t_{out}}$	Розмір вихідного реферату; встановлюється у відсотковому значенні від t_{in}	Чисельні дані, що надає користувач	Кожного разу при використанні підсистеми автоматного реферування текстів
l_i	Список індикаторів для зважування термінів та значення ваги, на яку вони збільшують оцінку терміну	Дані вбудовані в програмний код, визначені на етапі проектування підсистеми реферування	Кожного разу при використанні підсистеми автоматного реферування текстів.

Таблиця 3.2 – Перелік та опис вихідних повідомлень

Повідомлення	Термін і частота видачі	Отримувач
Tf-idf вага термінів вхідного документу	Кожного разу при проведенні процедури зважування термінів, для формування словника	Програмний код

Продовження таблиці 3.2

Повідомлення	Термін і частота видачі	Отримувач
Фінальна вага термінів вхідного документу (з додаванням індикаторів)	Кожного разу при проведенні процедури зважування термінів, для формування словника	Програмний код
Середня вага термінів вхідного документу	Кожного разу при проведенні процедури зважування термінів, для формування словника	Програмний код
Список термінів, що утворюють словник, з їх вагою	Кожного разу при проведенні процедури зважування термінів, для формування словника	Програмний код
Функціональна вага речень;	Кожного разу при проведенні процедури зважування речень	Програмний код
Базова вага речень	Кожного разу при проведенні процедури зважування речень	Програмний код
Кількість речень у тексті;	Кожного разу при нормалізації оцінки речень	Програмний код
Кількість слів у тексті	Кожного разу при нормалізації оцінки речень	Програмний код

Продовження таблиці 3.2

Кількість слів у поточному реченні	Кожного разу при нормалізації оцінки речень	Програмний код
Фінальна вага речень	Кожного разу при проведенні процедури зважування речень	Програмний код
Відсортований список зважених речень	Кожного разу при формуванні результуючого реферату	Програмний код
Список речень із вхідного документу, що входять до реферату	Кожного разу при формуванні результуючого реферату	Користувач

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНИЙ АНАЛІЗ

4.1 Вибір інструментів та середовища розробки

Оскільки програмний засіб, який необхідно розробити передбачає виконання певних дій по обробці текстів на природній мові, таких як знаходження стоп слів для певних мов, приведення слів у тексті до їх квазі-основ, векторне представлення текстів, виконання специфічних математичних розрахунків, та багато іншого, мовою для написання підсистеми автоматичного реферування була обрана мова Python. Її вибір обумовлений тим, що вона відмінно адаптована для вирішення завдань в області машинного навчання загалом, і містить широкий інструментал для задач в рамках NLP в тому числі (пакети для оцінювання якості реферування, словники стоп-слів, попередньо завантажені набори текстів та інструменти для роботи з ним, готові реалізації деяких алгоритмів реферування, стемери), та для складних математичних розрахунків.

Python – високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читання коду.

Python підтримує кілька парадигм програмування, в тому числі структурну, об'єктно-орієнтовану, функціональну, імперативну і аспектно-орієнтовану. Основні архітектурні риси - динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень і зручні високорівневі структури даних [27].

Python портований і працює майже на всіх відомих платформах - від КПК до мейнфреймів. Існують порти під Microsoft Windows, практично всі варіанти UNIX (включаючи FreeBSD і Linux), Plan 9, Mac OS і Mac OS X, iPhone OS 2.0 і вище, Palm OS, OS / 2, Amiga, HaikuOS, AS / 400 і навіть OS / 390, Windows Mobile, Symbian і Android [27].

Основні можливості та особливості [28]:

– класи є одночасно об'єктами з усіма нижче наведеними можливостями;

– спадкування, в тому числі множинне;

– поліморфізм (всі функції віртуальні);

– інкапсуляція (два рівні - загальнодоступні та приховані методи і поля). Особливість - приховані члени доступні для використання і позначені як приховані лише особливими іменами;

– спеціальні методи, що керують життєвим циклом об'єкта: конструктори, деструктори, розподільники пам'яті;

– перевантаження операторів (всіх, окрім is, '.', '=' I символьних логічних);

– властивості (імітація поля за допомогою функцій);

– управління доступом до полів (емуляція полів і методів, частковий доступ, і т. п.);

– метапрограмування (управління створенням класів, тригери на створення класів, та ін.);

– повна інтроспекція;

– класові і статичні методи, класові поля;

– класи, вкладені в функції і класи.

У якості середовища розробки був обраний PyCharm. PyCharm – інтегроване середовище розробки для мови програмування Python (рисунок 3.1). Надає засоби для аналізу коду, графічний зневаджувач, інструмент для запуску юніт-тестів і підтримує веб-розробку на Django. PyCharm розроблена чеською компанією JetBrains на основі IntelliJ IDEA.

PyCharm має зручний редактор коду зі всіма корисними функціями: підсвічуванням синтаксису, автоматичним форматуванням, доповненням і відступами.

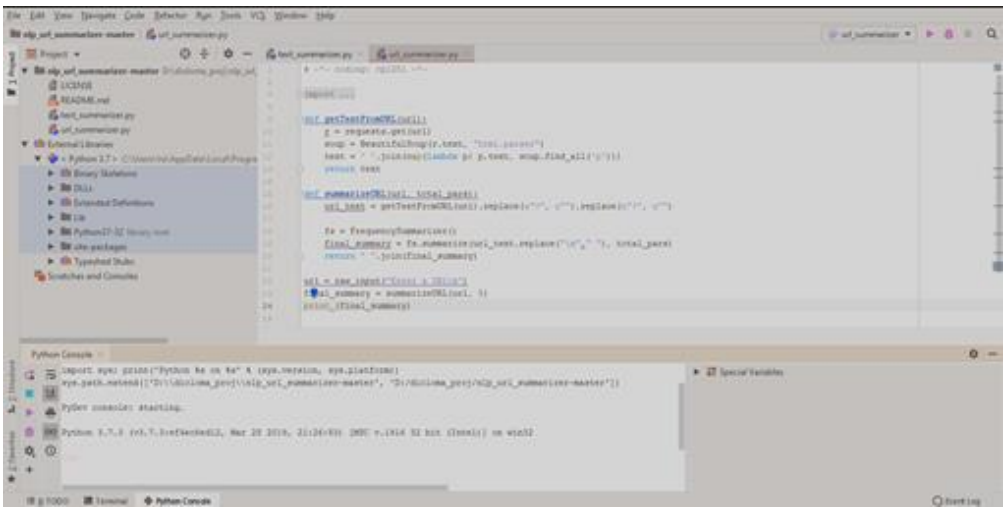


Рисунок 4.1 – Середовище розробки PyCharm

PyCharm дозволяє перевіряти версії інтерпретатора мови на сумісність, а також використовувати шаблонні коди. PyCharm дозволяє швидко робити рефакторинг коду, а також використовувати зручний графічний відладчик. У PyCharm можна проводити інтегроване тестування Unit, використовувати інтерактивні консолі для Python, Django, SSH, і баз даних .

Були використані наступні специфічні бібліотеки:

– NLTK (Natural Language Toolkit) – це провідна платформа для створення програм на Python для роботи з даними на природній мові. Вона надає прості у використанні інтерфейси для більш ніж 50 корпоративних та лексичних ресурсів таких як WordNet, а також набір бібліотек обробки текстів на природній мові для класифікації, токенізації, стемінгу, лемітизації, маркування, аналізу, попередньої обробки та семантичного аналізу;

– Scikit-learn – це безкоштовна бібліотека машинного навчання мови програмування Python. Вона містить різні алгоритми класифікації, кластеризації, регресії, а також призначена для роботи з чисельними та науковими бібліотеками Python NumPy та SciPy;

– SciPy – бібліотека з відкритим вихідним кодом, призначена для виконання наукових та інженерних розрахунків;

– Chardet – універсальний детектор кодування символів;

– Guess-language – застосовується для визначення природної мови обраного тексту у кодуванні Unicode (utf-8). Виявляє більше 60 мов, використовує евристику на основі набору символів і триграм в зразку тексту для визначення мови. Вона краще працює з більш довгими зразками тексту та буде збита з пантелику, якщо зразок тексту буде включати розмітку, наприклад таку як HTML;

– NumPy (Numeric Python) – це модуль, який надає загальні математичні і числові операції у вигляді пре-скомпільованих, швидких функцій. Вони забезпечують функціонал, який можна порівняти з функціоналом MatLab. NumPy надає базові методи для маніпуляції з великими масивами і матрицями та має розширення SciPy (Scientific Python).

4.2 Система автоматичного реферування текстів новинних статтей

4.2.1 Опис алгоритму

Весь процес складається з трьох етапів:

а) попередня обробка;
   1) ідентифікація заголовку;
   2) поділ тексту на параграфи;
   3) поділ параграфів на речення;
   4) токенізація;
      – видалення стоп слів;
      – стеммінг;

б) підрахунок балів;
   1) зважування термів;
   2) відбір термів у словник ключової лексики;
   3) зважування речень;

в) генерація реферату.

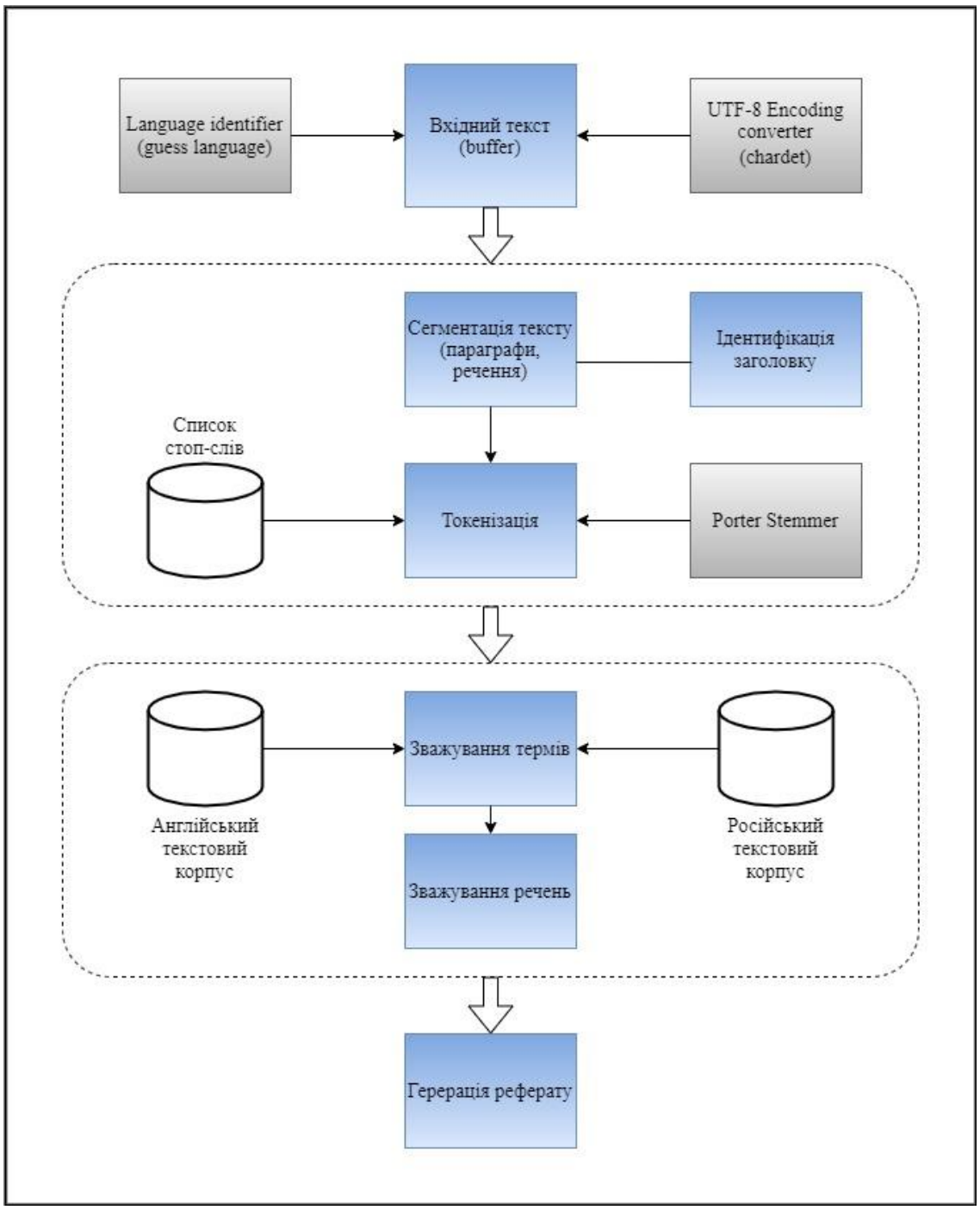


Рисунок 4.2 – Узагальнена схему роботи системи автоматичного реферування текстів новинних статей

На етапі попередньої обробки суматор проходить по вхідному тексту і виконує чотири основні процедури:

– Визначення заголовку статті. Заголовком вважається рядок до першого символу нового рядка без крапки в кінці. Проте рядок з крапкою

також може бути інтерпретований як заголовок, якщо він закінчується акронімом або абревіатурою (“США”, “і т. д.”). Рядок також має бути довжиною не більше сімнадцяти токенів. Заголовок використовується пізніше для присвоєння додаткових ваг термам при формуванні словника. Тому накладається додаткове обмеження: тексти для підсумовування мають надаватися із заголовками.

– Розбиття тексту на абзаци. Текст розділяється на абзаци за управляючими символами нового рядка. Межі абзаців необхідно знати, щоб знайти їх перші та останні речення та реалізувати додаткову оцінку термів на основі позиції.

– Розбиття абзаців на речення. Ця процедура виконується у два етапи: початкова декомпозиція речень, виправлення після розщеплення. На першому етапі всі можливі термінатори речення ('.', '!', '?', ':', ';', '...') перевіряються на відповідність регулярними виразами, що описують лівий і правий контексти цих термінаторів. Також передбачена обробка простих випадків, коли між двома реченнями опускається пробіл. На другому етапі неправильно розбиті речення об'єднуються. Після цього етапу система повертає введений текст у вигляді списку абзаців із вкладеними списками окремих речень.

– Токенізація речень. Модуль розділяє речення на слова, зіставляючи рядок з шаблоном регулярного виразу. Після цього кожне речення представляється у вигляді списку символів Python в нижньому регістрі (цифри зберігаються) без знаків пунктуації. Далі ті лексеми, що не являються стоп-словами оброблюються стемером Портера, утворюючи список кортежів (стема, токен). Така структура даних допомагає при вилученні ключових слів.

В результаті етапу попередньої обробки вхідний текст перетворюється на великий список Python-абзаців, кожен з яких містить вкладені списки розділених та токенізованих речень, очищених від стоп-слів, розділових значень та з трансформованими термами.

Далі йде етап підрахунку балів. На цьому етапі суматор розраховує ваги термів та динамічно створює словник ключових слів, на основі якого він

зважує речення вхідного тексту. Спочатку розраховується оцінка термів способом, представленим у попередньому розділі. Далі, терми з вагами, що перевищують середню, відсортовані за спаданням, відбираються до списку ключових слів. Результуюча структура даних представляє собою список кортежів, що містять основи та їх оцінку. Наступним кроком розраховується оцінка речень способом, представленим у попередньому розділі, створюється новий список, що містить кортеж з речень та їх оцінок.

На третьому етапі суматор вибирає n перших речень зі списку, генерованого на попередньому кроці. Кількість речень, які будуть використані в остаточному резюме, встановлюється в залежності від користувача. За замовчуванням ступінь стиснення становить 20% від усіх речень у списку.

4.2.2 Опис текстових корпусів

Оскільки модель передбачає інтертекстуальний підхід до зважування термів їй необхідні додаткові тексти (окрім вхідного документу) на основі яких будуть розраховані оцінки лексем вхідного документу. Функцію таких документів у даному випадку виконують текстові корпуси, а також представляють собою модель тієї чи іншої мови.

Для забезпечення необхідної якості реферування та виконання вимоги до універсальності моделі, використовувані корпуси відповідають наступним критеріями:

– одномовність;
– орієнтованість на вирішення багатьох лінгвістичних задач;
– охоплення безліч стилів і жанрів;
– максимальна репрезентативність щодо предметних областей та тематики текстів.

Модель англійської мови представлена британським національним корпусом (BNC – British National Corpus). BNC – це колекція зразків писемної та розмовної мови з широкого кола джерел, що становить близько ста

мільйонів слів, призначена для представлення широкого спектру британської англійської мови пізньої частини 20-того сторіччя. Останнє видання – BNC XML Edition, випущене у 2007 році.

Письмова частина BNC (90%) включає, наприклад, виписки з регіональних та національних газет, спеціалізованих періодичних видань та журналів для будь-якого віку та інтересів, академічні книги та популярну художню літературу, опубліковані та неопубліковані листи та меморандуми, шкільні та університетські есе, та багато інших видів тексту.

Основну інформацію про тексти британського національного корпусу можна побачити у таблиці 4.1 та на рисунку 4.3.

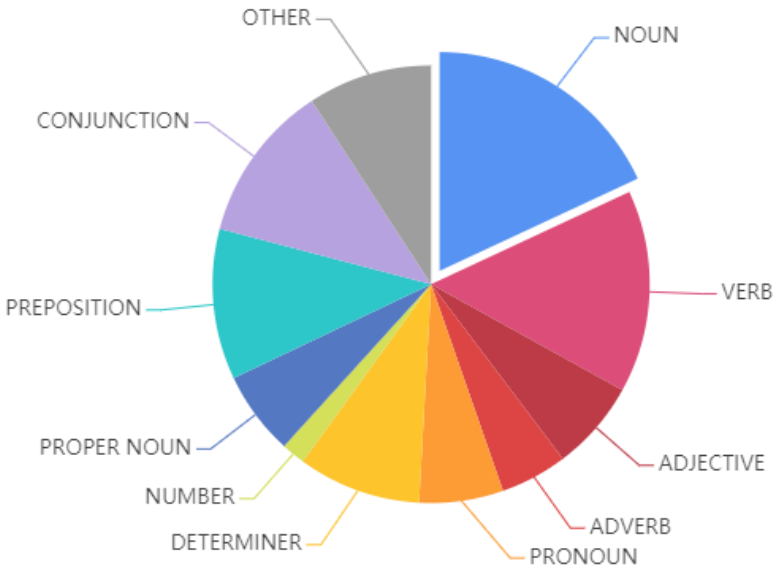


Рисунок 4.3 – Розподіл частин мови у текстах BNC

Таблиця 4.1 – Базова інформація про тексти BNC

	Частота
Токени	112 346 000
Слова	96 135 000
Речення	6 052 000
Документи	4 000

Оскільки навчальні (та різноманітні додаткові дані) у області NLP у значній мірі представлені лише для англійської мови, текстовий корпус для презентації моделі російської мови був створений самостійно за допомогою сервісу Sketch Engine [29]. Це інструмент онлайн-аналізу тексту, який працює з великими зразками мови, званими текстовими корпусами, для визначення того, що типово і часто використовується в мові, а що є рідкісним, застарілим, що виходить з ужитку або якими новими словами або граматикою починають користовуватися. Він дозволяє виконувати наступні дії по аналізу корпусу: відшукувати типові комбінації (колокацію) слів, створювати тезаурус для обраного слова, визначати неологізми, виявляти ключові слова та багато іншого.

Sketch Engine також служить програмним забезпеченням для побудови власних корпусів. Він надає доступ до інструменту по створенню корпусу, який використовує технологію WebBootCaT для автоматичного створення текстового корпусу з відповідних веб-сторінок. Дані, завантажені з Інтернету, очищаються, при необхідності дедуплікуються, а нетекстові дані видаляються для отримання лінгвістично цінного текстового матеріалу. Користувач може вказати який контент слід завантажити, шляхом обрання вихідних слів, що представляють деяку тему. За допомогою цієї функції був створений корпус, що представляє модель російської мови для системи реферування. Він був створений на основі 300 документів, зібраних з різноманітних російськомовних веб-сайтів, що охоплюють більше ніж сорок предметних областей, наприклад таких як: мистецтво, історія, політика, психологія, соціологія, наука, програмна інженерія, штучний інтелект, новини, мандрівки, сучасні технології, космос, психологія, образ життя, соціальні мережі, економіка, менеджмент, стартапи, література, поезія, книжки, здоров’я, блоги, музика, культура, фотографія, гумор та ін. Веб ресурси, що були обрані представляють собою частини книг, художної літератури, науково-популярних та інших блогів (таких як medium.com чи habr.com), спеціалізованих видань, усе це, а також велика дисперсія тем забезпечують

дійсно репрезентативний зразок мови. Основну інформацію про тексти створеного корпусу можна побачити у таблиці 4.2 та на рисунку 4.4.

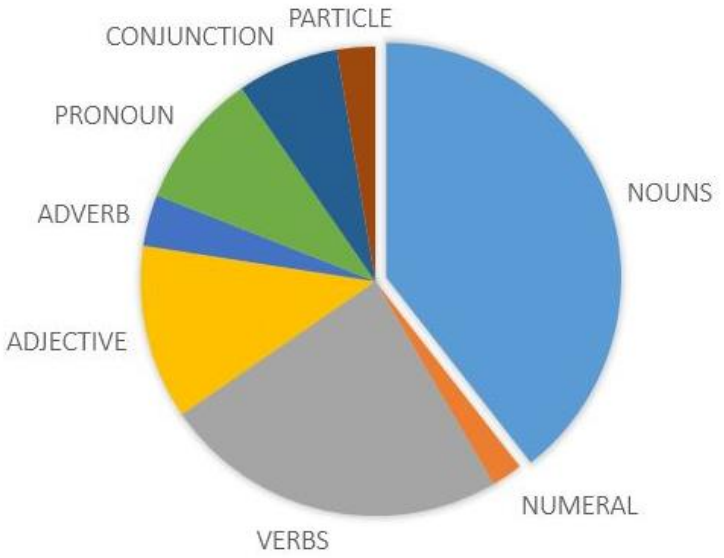


Рисунок 4.4 – Розподіл частин мови у корпусі російської мови

Таблиця 4.2 – Базова інформація про корпус російської мови

	Частота
Токени	1 180 727
Слова	908 818
Речення	56 750
Документи	300

4.2.3 Наочні результати роботи системи

Для того, щоб наочно показати роботу підсистеми для неї був реалізований консольний інтерфейс. Користувач вводить текстовий фрагмент, що його цікавить, і отримує результуючу анотацію через консольний вивід. Але передбачається, що такий модуль у майбутньому буде використовуватися у якості API для систем, що потребують вирішення задачі автоматичного анотування текстів/документів, розміщених у мережі.

Робота модулю продемонстрована на рисунках 4.5- 4.10. Для наочної демонстрації були обрані три новинні статті (одна на англійській мові, та дві – на російській). Одна з них (на англійській мові) є звичайною новинною статтею, що стосується деякої події та не містить слів, притаманних конкретній предметній області, та є загальним прикладом. Текст статті про YouTube містить більшу кількість специфічних слів ніж попередній, але їх можна вважати загальновживаними, адже вони стосуються сучасного інтернет-простору, та використовуються великою кількістю людей. Остання стаття – найбільш вузьконаправлена, вона містить велику кількість визначень, термінів, пояснень, що стосуються однієї теми та найчастіше вживаються саме спеціалістами цієї області. Також вхідні тексти є різними зв розміром.

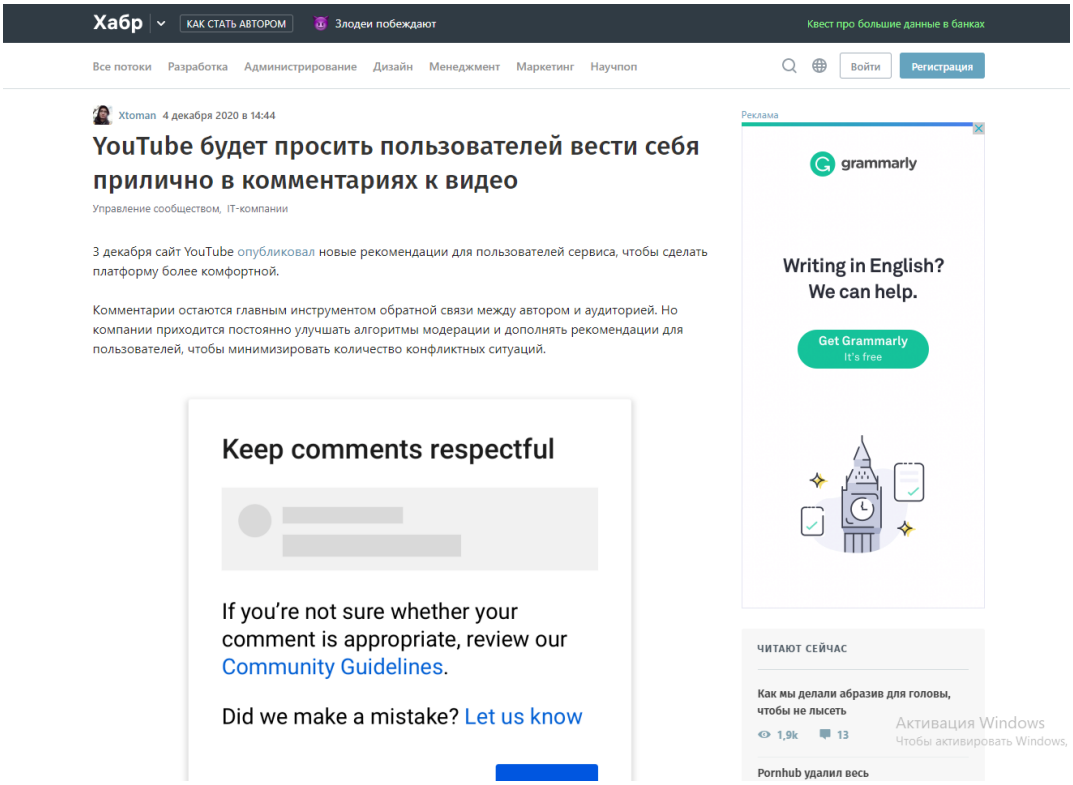


Рисунок 4.5 – Знімок екрану зі статтею про новину

```
main
/home/IrinaD/TextSummarizer/venv/bin/python /home/IrinaD/TextSummarizer/main.py
YouTube будет просить пользователей вести себя прилично в комментариях к видео
3 декабря сайт YouTube опубликовал новые рекомендации для пользователей сервиса, чтобы сделать платформу более комфортной.
Комментарии остаются главным инструментом обратной связи между автором и аудиторией. Но компании приходится постоянно
 улучшать алгоритмы модерации и дополнять рекомендации для пользователей, чтобы минимизировать количество конфликтных
 ситуаций.
Так, если пользователь написал комментарий с негативным содержанием, на сайте сработает фильтр и появится предупреждение с
 просьбой при появлении сомнений ознакомиться с рекомендациями платформы, чтобы не оскорбить других пользователей.
YouTube считает, что такая мера даст комментатору повод дополнительно обдумать содержание сообщения перед публикацией.
 Однако пользователь сможет проигнорировать предупреждение и комментарий будет размещён. Компания планирует, что новая
 функция позволит дополнительно снизить количество комментариев, содержащих оскорбления и ненависть. Сейчас хостинг
 постоянно модерирует и удаляет подобные комментарии. Их количество с начала 2019 года увеличилось в 46 раз.
Обновление станет частью глобальной системы для уменьшения ненависти среди пользователей. Компания также планирует
 протестировать фильтрацию комментариев для авторов роликов. В YouTube Studio будут автоматически фильтроваться
 комментарии, которые могут задеть чувства создателя и актёров в видео. Функция сначала будет доступна пользователям
 Android, когда изменения появятся на других платформах, пока неизвестно.
Схожая функция была добавлена в октябре 2020 года в Instagram. При срабатывании фильтра появляется всплывающее окно с
 напоминанием сохранять уважение в комментариях. Появление такого сообщения не означает, что комментарий обязательно будет
 удалён, если его не отредактировать. Предупреждение лишь рекомендация, которая не помешает отправить комментарий.
YouTube не поясняет, как нововведение будет работать с языками, отличными от английского.


------------------------------
Article's summary:

------------------------------

    YouTube будет просить пользователей вести себя прилично в комментариях к видео.
Так, если пользователь написал комментарий с негативным содержанием, на сайте сработает фильтр и появится предупреждение с
 просьбой при появлении сомнений ознакомиться с рекомендациями платформы, чтобы не оскорбить других пользователей.
Однако пользователь сможет проигнорировать предупреждение и комментарий будет размещён.
Сейчас хостинг постоянно модерирует и удаляет подобные комментарии.
В YouTube Studio будут автоматически фильтроваться комментарии, которые могут задеть чувства создателя и актёров в видео.
Предупреждение лишь рекомендация, которая не помешает отправить комментарий.


Process finished with exit code 0
```

Рисунок 4.6 – Результат реферування статті с рисунку 4.5

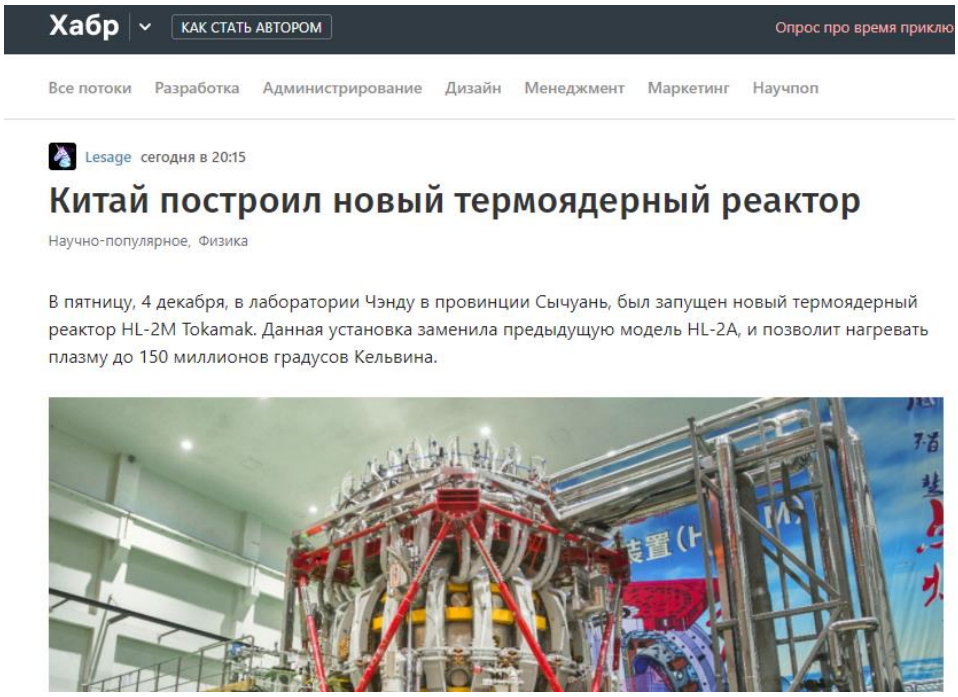


Рисунок 4.7 – Знімок екрану зі статтею про новину

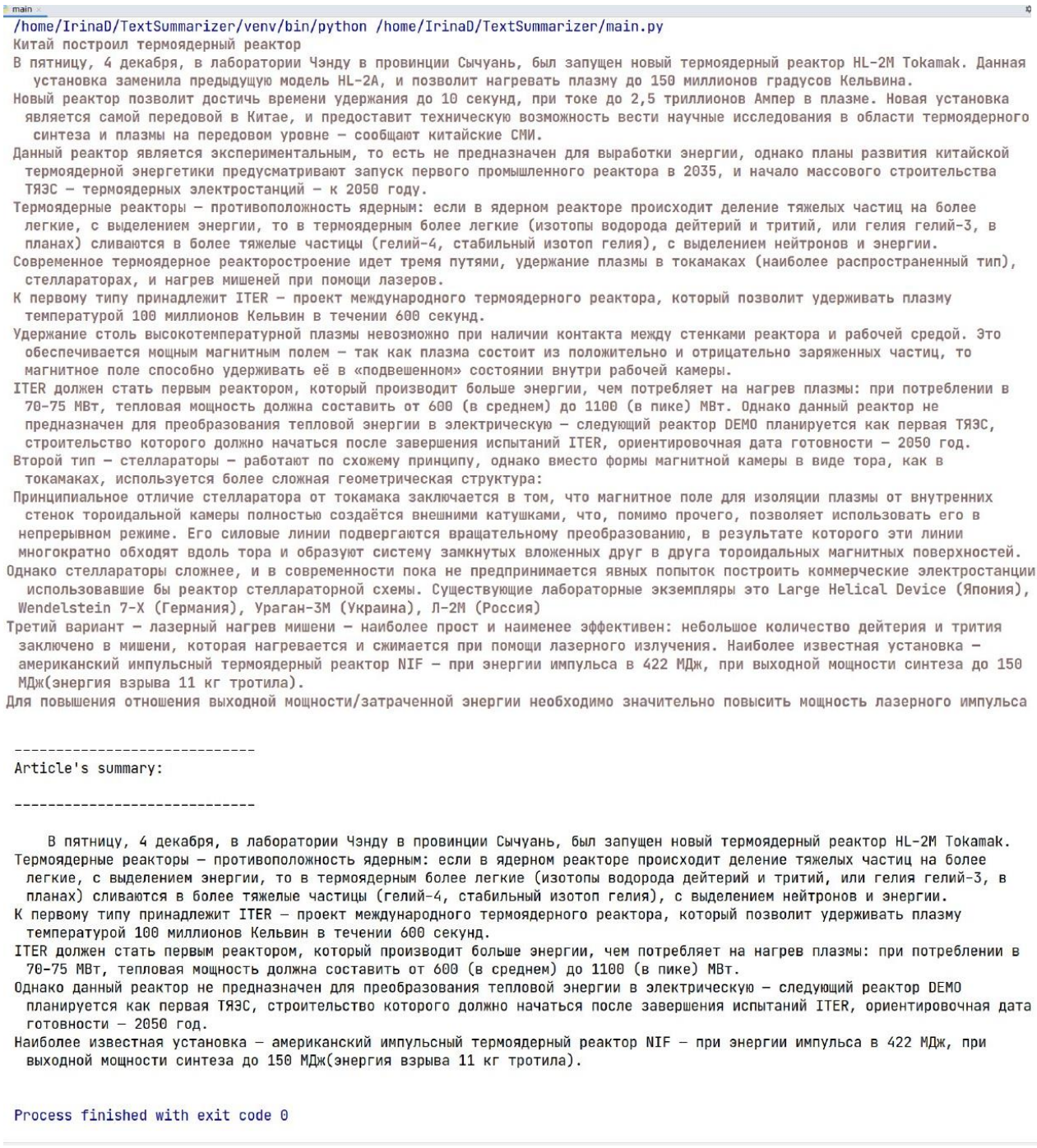


Рисунок 4.8 – Результат реферування статті с рисунку 4.7

Рисунок 4.9 – Знімок екрану зі статтею про новину

```
main
/home/IrinaD/TextSummarizer/venv/bin/python /home/IrinaD/TextSummarizer/main.py
NBA star Karl-Anthony Towns says he has lost 7 family members to Covid-19
(CNN)Minnesota Timberwolves star Karl-Anthony Towns is entering his sixth NBA season with a heavy heart.
"I've been through a lot, obviously starting out with my mom," Towns said on a video call with reporters on Friday.
Towns' mother, Jacqueline Towns, died in April of complications from Covid-19. On Friday, the 25-year-old Towns told
 reporters that six other family members have died because of Covid-19.
"Last night I got a call that I lost my uncle," Towns said. "I feel like I've been hardened a little bit by life and
 humbled."
Towns said he's trying to keep his family safe. His father, Karl-Anthony Towns Sr., also contracted Covid-19 but recovered.
"I've seen a lot of coffins in the last seven months, eight months," Towns said. "But I have a lot of people who have -- in
  my family and my mom's family -- who have gotten Covid. I'm the one looking for answers still, trying to find how to keep
  them healthy. It's just a lot of responsibility on me to keep my family well-informed and to make all the moves necessary
  to keep them alive."
But when asked if basketball would be a type of therapy for him, Towns acknowledged the difficulty he faces as he prepares
 for the new season, saying he hasn't been in a good place mentally since his mother was hospitalized.
"I play this game more because I just loved watching my family members seeing me play a game I was very successful and good
  at. It always brought me a smile when I saw my mom at the baseline and in the stands and stuff and having a good time
 watching me play," Towns said. "So it's going to be hard to play. It's going to be difficult to say this is therapy. I
 don't think this will ever be therapy for me again. But it gives me a chance to relive good memories I had."
After his mother was first placed in intensive care in March, Towns posted an emotional video on Instagram imploring people
  to take the pandemic seriously and to take every precaution.
On March 15, Towns announced he would donate $100,000 to the Mayo Clinic, saying at the time in an Instagram post, "My hope
  is that we can fight this virus quicker and more efficiently by increasing the testing capabilities and availability and
 Mayo Clinic's overall Covid-19 response."
Towns plays center for the Timberwolves. He was the NBA Rookie of the Year in 2016 and has twice been named an NBA All-Star.
The Timberwolves will open the 2020-21 NBA season against the Detroit Pistons on December 23 at Target Center in Minneapolis.


Article's summary:

------------------------------

    NBA star Karl-Anthony Towns says he has lost 7 family members to Covid-19.
"I've been through a lot, obviously starting out with my mom," Towns said on a video call with reporters on Friday.
Towns' mother, Jacqueline Towns, died in April of complications from Covid-19.
On Friday, the 25-year-old Towns told reporters that six other family members have died because of Covid-19.
His father, Karl-Anthony Towns Sr., also contracted Covid-19 but recovered.


Process finished with exit code 0
```

Рисунок 4.10 – Результат реферування статті с рисунку 4.7

4.3 Оцінка роботи системи

Для перевірки роботи алгоритму був використаний набір даних «The 20 Newsgroups», що дозволить оцінити його точність за метриками, описаними у розділі 1.5. Цей набір даних включає близько 18000 документів груп новин, розділених майже рівномірно за 20 різними групами новин та розділених на дві підмножини: одна для навчання (або розробки), а інша для тестування (або для оцінки продуктивності). Колекція 20 груп новин є широковідомою та стала популярним набором даних для експериментів в таких випадках як класифікація і кластеризація текстів. В даному випадку нас цікавлять лише тексти зібраних новин, а не їх поділ на певні набори за темою (адже не стоїть задача класифікації чи кластеризації), проте такий поділ забезпечує необхідну репрезентативність текстів для оцінювання. Також цей набір даних був обраний тому, що він доступний для використання у python-бібліотеці scikit-learn (модуль sklearn.datasets) і не має необхідності створювати, шукати та завантажувати додаткові набори даних. Завантажувач цього модулю дозволяє повернути список необроблених текстів із набору, що будуть використовуватися для реферування та подальшого оцінювання.

У якості алгоритма для порівняння був обраний TextRank (pyTextRank у реалізації python), бо він є найяскравішим представником графових підходів, які в свою чергу є класичними представниками методів, що базуються на зв’язках між реченнями, а оскільки розроблений алгоритм намагається окрім зв’язності речень враховувати, ще й ключову лексику було б логічно порівнято його з одним з «чистих» підходів (до того ж з таким загальновідомим та загальновживаним). У таблиці 4.3 наведені значення оцінки роботи моделі за косинусною близкістю та розбіжністю Дженсена-Шенона.

Таблиця 4.3 – Порівняння отриманої оцінки розробленого алгоритму з алгоритмом TextRank

	Косинусна близкість	JS розбіжність
Власний алгоритм	0,86028	0.39226
TextRank	0,83632	0,38542

Як можна побачити розроблений алгоритм трохи перевищив оцінку алгоритму TextRank за косинусною близкістю та виявся майже однаковим за розбіжністю Дженсена-Шенона. Тобто можна зробити висновок, що створений алгоритм працює не гірше (а в деяких випадках навіть краще) за алгоритм TextRank та може бути чудовою альтернативою такому алгоритму. Також варто взяти до уваги те, що розроблений алгоритм використовує просту модель для пошуку ключової лексики, тож можна зробити припущення, що подальше покращення алгоритму, наприклад, врахування синонімів при виділенні ключової лексики вхідного тексту, або використання для цього більш складних моделей, наприклад таких як , латентне розміщення Дирихле (Latent Dirichlet Allocation), зважування особливим способом деяких частин тексту (наприклад списків), зможе значно покращити оцінку роботи моделі.

Також робота алгоритму була імперативно (візуально, експертно) порівняна з роботою, вже існуючих відкритих систем (чи модулів) реферування, що доступні користувачам. Для цього були обрані статті, реферати для якої створила кожна система. Був обраний текст новини на російській мові з сайту habr.com, що стосується нових правил коментування на платформі YouTube[30], текст цієї статті не містить спеціалізованих слів (окрім слів, що стосуються сучасного інтернет-простору, таких як: коментар, сервіс, платформа та ін.) та не охоплює конкретну предметну область, тому чудово підходить у якості загального наочного прикладу, та текст статті (матеріалів наукової конференції) на англійській мові, яка стосується розподілу ресурсів та прийняття рішень у надзвичайних ситуаціях[31], ця стаття є більш вузькоспеціалізованою за предметною областю та стилем

викладення. У якості систем для порівняння були обрані система Open Text Summarizer[32] – це інструмент, що автоматично аналізує тексти на різних мовах і визначає їх найбільш важливі частини (точний алгоритм реферування не відомий, користувачам доступний інтерфейс для роботи з інструментом) та python-модуль Sumy, призначений для скорочення текстових документів, який містить реалізації деяких алгоритмів та методів реферування.

На обох статтях власний алгоритм дав прийнятні результати. Для наочності у додатку А представлені реферати, створені знайденими системи та власним алгоритмом а також текст оригінальної статті на російській мові, що підлягав підсумовуванню.

Суб'єктивні оцінки рефератів надані експертним способом, тбто надання балів (від 1 до 5) за відповідність реферату деякій характеристиці представлені в таблиці 4.4

Таблиця 4.4 – оцінка рефератів експертним методом

	TextRank	LSA	Open Text Summarizer	Власний алгоритм
Зрозумілість контексту викладення	5	3	5	5
Логічна послідовність та зв'язність викладення	5	2	4	5
Унікальність сенсового навантаження тез реерату	3	5	4	4
	13	10	13	14

Прочитавши реферати, можна помітити, що реферат, створений за методом LSA, єдиний до складу якого не потрапив заголовок (перше речення тексту), що відразу робить його більш складним до сприйняття через

відсутність початково контексту. Цей реферат містить найбільше речень, що не зустрічаються в інших рефератах, це можна пояснити тим, що LSA використовує кардинально інший підхід до реферування, ніж інші алгоритми. Проте в даному випадку цей реферат сприймається найгірше з поміж інших з точки зору користувача, адже не одразу зрозуміло саме якої теми вони стосуються та вони виглядають менш зв’язаними між собою ніж речення інших рефератів.

Реферат, створений за методом TextRank, має найменшу кількість унікальних речень у сенсі смислового навантаження, три останні речення реферату по факту виражають одну і ту ж думку. Виділення саме цих речень в даному випадку є доволі логічним, вони знаходяться у різних частинах тексту та вочевидь мають найбільше зав’язків (в контексті TextRank вони напевно є центрами своїх підграфів). З точки зору користувача тези у рефераті зрозумілі, очевидно про що йде мова, вони підтримують логічну послідовність викладення, але в цьому випадку якщо скоротити реферат на два останні речення, він не втратить жодної інфрмації.

Реферат, створений системою Open Text Summarizer є більш унікальним в плані смислового навантаження ніж TextRank-реферат, та містить речення, що взагалі не зустрічаються в інших рефератах. Він трохи поступається попередньому реферату у зв’язності, проте очевидність контексту зберігається. Можна сказати, що він несе більше корисної інформації для користувача ніж TextRank-реферат.

Реферат створений власним алгоритмом також є більш унікальним в плані смислового навантаження ніж TextRank-реферат і також містить речення, що взагалі не зустрічаються в інших рефератах. Тези у рефераті зрозумілі, очевидно про що йде мова, вони підтримують логічну послідовність викладення. Реферат також претендує на «художню» зв’язність речень, та можливість подання його у вигляді суцільного тексту, а не окремих тез (але це залежить від конкретного вхідного тексту).

ВИСНОВКИ

У атестаційній роботі були досліджені основні методи в рамках «extractive summarization» підходу до реферування текстів. Були виявлені переваги і недоліки різних підходів та простежено взаємозв'язки між ними. В ході дослідження було з’ясовано, що в рамках підходу можна виділити дві групи методів, що використовують дві різні ідеї для пошуку важливої інформації. Одні, засновані на припущенні, що найбільш важлива інформація міститься в окремих реченнях тексту. Вони використовують важливу особливість текстів – зв’язаність речень між собою та їх залежність одне від одного. Інші, загострюють увагу на пошуку ключової лексики тексту, яка найкраще передає його основний зміст. Недоліком перших підходів є дослідження ними лише формальної сторони структури речень, без прийняття до уваги синтаксичної та семантичної інформації, що погіршує ефективність систем узагальнення, других – ігнорування при пошуку важливих речень зв’язків між ними.

В атестаційній роботі представлені результати, які є, відповідно до поставленої мети, рішенням завдання анотування текстового контенту будь якої тематики, та на різних мовах. У роботі представлений метод автоматичного реферування текстів, який об’єднує обидві ідеї, для оцінки речень, він дозволяє враховувати і взаємозв’язки між реченнями і ключову лексику, що міститься в них. У рамках алгоритму створюється словник тематичної лексики вхідного тексту. Наявність елементів з нього становить основу для розрахунку зв’язків між реченнями, а також дозволяє розрахувати розподіл тематичної лексики у реченнях. Метод зберігає універсальність підходів, заснованих на зв’язках, адже використовує статистичні підходи для пошуку тематичної лексики і не потребує використання лінгвістичних баз знань. Розроблений метод не накладає обмеження та тематику вхідних текстів, не потребує складних обчислень та може бути адаптований для роботи з

різними мовами лише за рахунок зміни текстового корпусу, що репрезентує необхідну мову.

Основні результати за темою магістерської атестаційної роботи у вигляді тез доповіді, опубліковано у матеріалах міжнародної студентської наукової конференцій [1].

За результатам досліджень, проведених при виконанні атестаційної роботи, можна зробити висновок, що розроблений метод успішно ідентифікує найважливіші речення в тексті на основі інформації, взятої виключно з самого тексту. При оцінюванні роботи системи було виявлено, що алгоритм дає не гірші (за деякими показниками навіть кращі) результати ніж популярний на сьогодні алгоритм TextRank. Проте у подальшому метод може бути легко модифікований наприклад, за рахунок використання більш складних (ніж TF-IDF модель) методів для знаходження ключової лексики або методів для зважування речень, що дозволить покращити оцінку його роботи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Дорошенко. І.К. Дослідження та розробка методів автоматичного реферування текстового контенту // Актуальні питання та перспективи проведення наукових досліджень: матеріали міжнародної студентської наукової конференції (Вінниця, 6 листопада 2020 р.). О.: Друкарня ФОП Гуряєва В. М. 2020. Т. 2 С.30-31

2. Pustejovsky J., Stubbs A. Natural Language Annotation for Machine Learning. Cambridge; Farnham; 2012. 343 c

3. Wikipedia [Електронний ресурс]. Обработка естественного языка. Режим доступу: https://ru.wikipedia.org/wiki/Обработка_естественного_языка

4. Mani I., Maybury M. Advances in Automatic Text Summarization. MIT Press, 1999

5. A. Nenkova, K. McKeown. A survey of text summarization techniques. In Mining Text Data. Springer, 2012. C. 43–76

6. Hans Peter Luhn. 1958. The automatic creation of literature abstracts. IBM Journal of research and development 2, 2 (1958), 159–165.

7. Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. Computational linguistics 19, 1 (1993), 61–74.

8. Luhn H.P. The automatic creation of literature abstracts. IBM J. of Research and Development. 1958. 2. N 2. C.159-165.

9. Lin C.-Y., Hovy E.H. The Automated acquisition of topic signatures for text summarization. In Proceedings of COLING-00. Saarbrücken, Germany. 2000. P. 495-501.

10. Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. JASIS 41, 6 (1990), 391–407

11. Yihong Gong and Xin Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In Proceedings of the 24th annual

international ACM SIGIR conference on Research and development in information retrieval. ACM, 19–25

12. Josef Steinberger, Massimo Poesio, Mijail A Kabadjov, and Karel Ježek. 2007. Two uses of anaphora resolution in summarization. Information Processing & Management 43, 6 (2007), 1663–1680.

13. Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. Association for Computational Linguistics.

14. Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 68–73

15. You Ouyang, Wenjie Li, Sujian Li, and Qin Lu. 2011. Applying regression models to query-focused multi-document summarization. Information Processing & Management 47, 2 (2011), 227–237

16. Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008. Extractive summarization using supervised and semi-supervised learning. In Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1. Association for Computational Linguistics, 985–992.

17. Liang Zhou and Eduard Hovy. 2003. A web-trained extraction summarization system. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology- Volume 1. Association for Computational Linguistics, 205–211.

18. John M Conroy and Dianne P O'leary. 2001. Text summarization via hidden markov models. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 406–407.

19. Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. 2007. Document Summarization Using Conditional Random Fields.. In IJCAI, Vol. 7. 2862–2867.

20. Rasim M. Alguliev, Makrufa S. Hajirahimova, Chingiz A. Mehdiyev. MCMR: Maximum coverage and minimum redundant text summarization model. Expert Systems with Applications 38, 12 (2011)

21. Rasim M. Alguliev, Ramiz M. Aliguliyev, and Nijat R Isazade. 2013. Multiple documents summarization based on evolutionary optimization algorithm. Expert Systems with Applications 40, 5 (2013)

22. Günes Erkan and Dragomir R Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res.(JAIR) 22, 1 (2004)

23. C.Y. Lin. ROUGE: A package for automatic evaluation of summaries/ Proceedings of ACL Text Summarization Branches Out Workshop. 2004. C. 74 – 81.

24. A. Louis, A. Nenkova. Automatic Summary Evaluation without Human Models. TAC. 2008. C. N 3. C. 903-914.

25. Яцко В.А. «Симметричное реферирование: теоретические основы и методика». НТИ. 2002. 2. N 5. С. 18-28.

26. Яцко В.А. «Методика симметричного взвешивания предложений». НТИ. 2016. 2. N 2. С. 36-41.

27. Znaimo [Електронний ресурс]. Мова программування Python Режим доступу: https://znaimo.com.ua/Python

28. Wikipedia [Електронний ресурс]. Python. Режим доступу: https://uk.wikipedia.org/wiki/Python

29. Sketchengine [Електронний ресурс]. Режим доступу: https://app.sketchengine.eu/#dashboard?corpname=user%2FIryna_D_%2Fcorp

30. Habr [Електронний ресурс]. Режим доступу: https://habr.com/ru/news/t/531336/

31. Grebennik I., Reshetnik V., Ovezgeldyyev A., Ivanov V., Urniaieva I. (2019) Strategy of Effective Decision-Making in Planning and Elimination of Consequences of Emergency Situations In: Murayama Y., Velev D., Zlateva P. (eds)

Information Technology in Disaster Risk Reduction. ITDRR 2018. IFIP Advances in Information and Communication Technology. Springer, Cham Scopus

32. Splitbrain [Електронний ресурс]. Open Text Summarizer. Режим доступу: https://www.splitbrain.org/services/ots

33. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання / Нац. стандарт України. – Вид. офіц. – [Чинний від 2017-07-01]. – Київ: ДП «УкрНДНЦ», 2016. – 26 с.