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

"На правах рукопису"
УДК 004.08

ДО ЗАХИСТУ ДОПУЩЕНО
Завідувач кафедри

_____ Олександр РОЛІК

“ ____ ” _____ 2023 р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

на здобуття ступеня магістра

за освітньо-науковою програмою

«*Інтегровані інформаційні системи*»

зі спеціальності 126 «*Інформаційні системи та технології*»

на тему:

«Підсистема аналізу великих масивів текстової інформації з використанням ключових слів та фраз»

Виконала:
студентка 2 курсу, групи ІА-11мн
Коваль Юлія Володимирівна _____

Керівник:
к.т.н., доцент кафедри інформаційних систем
та технологій КПІ ім. Ігоря Сікорського,
Писаренко Андрій Володимирович _____

Рецензент:
професор кафедри обчислювальної техніки, д. ф.-м., с.н.с.
Гордієнко Юрій Григорович _____

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

Київ – 2023 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-наукова програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ Олександр РОЛІК

«__» _____2023 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Коваль Юлія Володимирівна

1. Тема дисертації «Підсистема аналізу великих масивів текстової інформації з використанням ключових слів та фраз»

науковий керівник дисертації Писаренко Андрій Володимирович к.т.н., доцент кафедри інформаційних систем та технологій КПІ ім. Ігоря Сікорського, затверджені наказом по університету від «20» березня 2023 р. № 1275-с.

2. Строк подання студентом дисертації “ 12 ” травня 2023 р.

3. Об’єкт дослідження: процес аналізу великих масивів текстової інформації.

4. Предмет дослідження: методи та технології аналізу великих масивів текстової інформації з використанням ключових слів та фраз, які дозволяють ефективно обробляти та здійснювати аналіз текстів з метою отримання важливої інформації.

5. Перелік завдань, які потрібно розробити:
розглянути і проаналізувати наукові джерела та існуючі методи та технології аналізу текстів, включаючи машинне навчання та оброблення природної мови; розробити алгоритми для оброблення великих масивів текстової інформації з використанням ключових слів та фраз, які дозволять автоматично визначати

значущі текстові елементи; реалізувати програмну систему для аналізу текстової інформації, яка забезпечить автоматичне визначення ключових слів та фраз; провести експериментальну перевірку розробленої системи на реальних текстових масивах; проаналізувати результати експерименту та зробити висновки щодо ефективності розробленої системи; запропонувати можливі шляхи покращення системи та перспективи подальшого дослідження в даній галузі; сформулювати рекомендації щодо застосування розробленої системи в практичній діяльності для аналізу великих масивів текстової інформації.

6 Перелік графічного (ілюстративного) матеріалу

Алгоритм попередньої обробки великих масивів текстової інформації. Діаграма компонентів. Діаграма прецедентів. Діаграма станів. Діаграма послідовності підсистеми. Даталогічна модель. Діаграма потоків.

7. Орієнтовний перелік публікацій

«Аналіз великих масивів текстової інформації з використанням ключових слів та фраз засобами штучного інтелекту» Коваль Ю.В. IV Міжнародна науково-практична конференція молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2023)»

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання “ 31 ” січня 20 23 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Систематизація результатів огляду літератури	12.02	
2	Порівняльний аналіз існуючих методів та технологій аналізу великих масивів текстової інформації	20.02	
3	Розробка алгоритму для оброблення великих масивів текстової інформації з використанням ключових слів та фраз	26.03	
4	Покращення процесу аналізу текстової інформації	10.04	
5	Розробка програмного забезпечення	15.04	
7	Проведення експериментальних досліджень розробленого алгоритму	15.04	
8	Оформлення документації	19.04	
9	Подання роботи на попередній захист	20.04	
10	Подання роботи на основний захист	17.05	

Студент _____ *Юлія КОВАЛЬ*

Науковий керівник _____ *Андрій ПИСАРЕНКО*

РЕФЕРАТ

Магістерська дисертація: 109 с., 28 рис., 3 табл., 34 джерела, 1 додаток.

Актуальність зумовлена необхідністю підвищення ефективності роботи з великими масивами текстової інформації.

Мета дослідження – підвищення ефективності оброблення та аналізу текстів з метою отримання важливої інформації.

Для досягнення поставленої мети сформульовано та вирішено наступні **задачі**:

- розглянути і проаналізувати наукові джерела та існуючі методи та технології аналізу текстів, включаючи машинне навчання та оброблення природної мови;
- розробити алгоритми для оброблення великих масивів текстової інформації з використанням ключових слів та фраз, які дозволять автоматично визначати значущі текстові елементи;
- реалізувати програмну систему для аналізу текстової інформації, яка забезпечить автоматичне визначення ключових слів та фраз;
- провести експериментальну перевірку розробленої системи на реальних текстових масивах;
- проаналізувати результати експерименту та зробити висновки щодо ефективності розробленої системи;
- запропонувати можливі шляхи покращення системи та перспективи подальшого дослідження в даній галузі;
- сформулювати рекомендації щодо застосування розробленої системи в практичній діяльності для аналізу великих масивів текстової інформації.

Об’єкт дослідження – процес аналізу великих масивів текстової інформації.

Предмет дослідження – методи та технології аналізу великих масивів текстової інформації з використанням ключових слів та фраз, які дозволяють

ефективно обробляти та здійснювати аналіз текстів з метою отримання важливої інформації.

Методи дослідження: механізми попередньої обробки та очистки текстових даних, механізми токенізації, лематизації та векторизації текстових даних, метод оцінки важливості термінів в документі, глибинне навчання, машинне навчання, наївний Баєсовий класифікатор.

Наукова новизна одержаних результатів полягає у удосконаленні методу аналізу текстів з метою отримання важливої інформації.

Публікації. «Аналіз великих масивів текстової інформації з використанням ключових слів та фраз засобами штучного інтелекту» Коваль Ю.В. IV Міжнародна науково-практична конференція молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2023)».

АНАЛІЗ ТЕКСТОВОЇ ІНФОРМАЦІЇ, КЛЮЧОВІ СЛОВА, АНАЛІЗ ТЕКСТУ, МАШИННЕ НАВЧАННЯ, ГЛИБИННЕ НАВЧАННЯ.

ABSTRACT

Master's dissertation: 109 p., 28 figures, 3 tables, 34 sources, 1 appendix.

The relevance of the research is due to the need to improve the efficiency of working with large amounts of textual information.

The purpose of the study is to increase the efficiency of text processing and analysis in order to obtain important information.

To achieve this goal, the following **tasks** were formulated and solved:

- review and analyze scientific sources and existing methods and technologies for text analysis, including machine learning and natural language processing;
- to develop algorithms for processing large amounts of textual information using keywords and phrases that will automatically identify significant textual elements;
- to implement a software system for analyzing textual information that will ensure automatic detection of keywords and phrases;
- to conduct an experimental test of the developed system on real text arrays;
- analyze the results of the experiment and draw conclusions about the effectiveness of the developed system;
- to suggest possible ways to improve the system and prospects for further research in this area;
- to formulate recommendations for the application of the developed system in practical activities for the analysis of large amounts of textual information.

The object of research is the process of analyzing large amounts of textual information.

The subject of the research is methods and technologies for analyzing large amounts of textual information using keywords and phrases that allow to effectively process and analyze texts in order to obtain important information.

Research methods: mechanisms of pre-processing and cleaning of text data, mechanisms of tokenization, lemmatization and vectorization of text data, method of

assessing the importance of terms in a document, deep learning, machine learning, naive Bayesian classifier.

The scientific novelty of the obtained results lies in the improvement of the method of text analysis in order to obtain important information.

Publications. "Analysis of large arrays of textual information using keywords and phrases by means of artificial intelligence" Koval Y.V. IV International Scientific and Practical Conference of Young Scientists and Students "Software Engineering and Advanced Information Technologies (SoftTech-2023)".

TEXT INFORMATION ANALYSIS, KEYWORDS, TEXT ANALYSIS, MACHINE LEARNING, DEEP LEARNIN.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

AD – Active Directory;

API – Application Programming Interface;

AWS – Amazon Web Services;

CNN – Convolutional Neural Network;

IDE – Integrated Development Environment;

IDF – Inverse Document Frequency;

LSI – Latent Semantic Indexing;

NLP – Natural Language Processing;

NLU – Natural Language Understanding;

NLTK – Python Natural Language Toolkit;

OCR – Optical Character Recognition;

RNN – Recurrent Neural Network;

TF – Term Frequency;

TF-IDF – Term Frequency-Inverse Document Frequency;

HTML – Hypertext Markup Language;

CSS – Cascading Style Sheets.

ВСТУП

В сучасному світі кількість текстової інформації зростає експоненційно, і з кожним днем її обсяг стає все більшим та більшим. Це створює величезне завдання для багатьох організацій та підприємств, які мають аналізувати ці масиви тексту, щоб отримати корисну інформацію та зробити важливі рішення. У таких умовах аналіз великих масивів текстової інформації з використанням ключових слів та фраз стає дедалі більш актуальним та важливим завданням. Для розв'язання цього завдання застосовуються різноманітні методи та технології, включаючи машинне навчання, глибоке навчання та інші. Ця робота присвячена огляду методів та підходів до аналізу великих масивів текстової інформації з використанням ключових слів та фраз, а також їх застосування в різних сферах, включаючи бізнес, науку та технології.

Аналіз тексту з використанням ключових слів та фраз дозволяє здійснювати ряд важливих завдань, таких як:

- розуміння стану справ у певній галузі – аналіз текстових даних з допомогою ключових слів та фраз може допомогти зрозуміти, які теми та питання є актуальними у певній галузі, які проблеми найбільш важливі для галузі, і які тенденції переважають у галузі;
- виявлення попиту на товари та послуги – аналіз текстових даних може допомогти виявити, які товари та послуги є найбільш популярними серед користувачів, що може бути корисним для бізнесу при прийнятті рішень щодо розвитку та маркетингу своїх продуктів;
- виявлення схожих документів – аналіз тексту з допомогою ключових слів та фраз може допомогти виявити схожі документи, що може бути корисним для здійснення пошуку та порівняння даних;
- покращення обробки та індексації текстової інформації – ключові слова та фрази можуть бути використані для покращення обробки та індексації текстової інформації, що допоможе забезпечити швидкий та ефективний доступ до неї;

– виявлення негативних відгуків – аналіз текстових даних з допомогою ключових слів та фраз може допомогти виявити негативні відгуки користувачів про певні товари та послуги, що може допомогти бізнесу виправити недоліки та покращити якість своїх продуктів та послуг.

Ці завдання є важливими як для бізнесу, так і для досліджень у різних галузях, тому аналіз тексту з використанням ключових слів та фраз є дуже важливим інструментом. Наприклад, у бізнесі він може допомогти зрозуміти потреби та уподобання клієнтів, аналізувати конкуренцію, моніторити репутацію компанії та багато іншого.

У дослідженнях, таких як медична діагностика, аналіз соціальних мереж, наукові дослідження тощо, аналіз тексту також є необхідним інструментом для збору та обробки даних. Крім того, аналіз тексту з використанням ключових слів та фраз може допомогти виявляти нові залежності та тенденції, які можуть бути невидимими при звичайному огляді даних. Наразі, аналіз тексту з використанням ключових слів та фраз є важливим інструментом у різних галузях та допомагає забезпечити якісний аналіз даних та прийняття обґрунтованих рішень.

Отже, аналіз великих масивів текстової інформації з використанням ключових слів та фраз - це важлива та актуальна задача в багатьох сферах, таких як бізнес, дослідження, медіа тощо. Цей підхід дозволяє отримати цінну інформацію з текстових даних, що дозволяє приймати обґрунтовані рішення та робити передбачення. Для аналізу текстових даних можна використовувати різні методи та підходи, включаючи статистичний аналіз, машинне навчання та глибоке навчання. Вибір методу залежить від конкретної задачі та доступної кількості даних.

Об’єкт дослідження

Об’єктом дослідження магістерської дисертації є процес аналізу великих масивів текстової інформації.

Предмет дослідження

Предметом дослідження магістерської дисертації є методи та технології аналізу великих масивів текстової інформації з використанням ключових слів та фраз, які дозволяють ефективно обробляти та здійснювати аналіз текстів з метою отримання важливої інформації.

Мета

Метою магістерської дисертації є підвищення ефективності оброблення та аналізу текстів з метою отримання важливої інформації.

Для досягнення мети були поставлені та вирішені такі задачі:

- розглянути і проаналізувати наукові джерела та існуючі методи та технології аналізу текстів, включаючи машинне навчання та оброблення природної мови;
- розробити алгоритми для оброблення великих масивів текстової інформації з використанням ключових слів та фраз, які дозволять автоматично визначати значущі текстові елементи;
- реалізувати програмну систему для аналізу текстової інформації, яка забезпечить автоматичне визначення ключових слів та фраз;
- провести експериментальну перевірку розробленої системи на реальних текстових масивах;
- проаналізувати результати експерименту та зробити висновки щодо ефективності розробленої системи;
- запропонувати можливі шляхи покращення системи та перспективи подальшого дослідження в даній галузі;
- сформулювати рекомендації щодо застосування розробленої системи в практичній діяльності для аналізу великих масивів текстової інформації.

Практичне значення

Практичне значення цієї дипломної роботи полягає в можливості використання розробленої системи аналізу великих масивів текстової інформації з використанням ключових слів та фраз у різних галузях та сферах діяльності. Зокрема, система може бути застосована для аналізу текстової інформації отриманої з систем транскрибування мовлення, моніторингу та аналізу соціальних мереж, аналізу відгуків користувачів про певний продукт або послугу, аналізу текстової інформації в наукових дослідженнях, політичній аналітиці, та багатьох інших галузях. Застосування такої системи може допомогти зекономити час та зусилля при обробці великих масивів текстової інформації та забезпечити більш точний та зручний аналіз цієї інформації.

1 ТЕОРЕТИЧНІ АСПЕКТИ АНАЛІЗУ ТЕКСТОВОЇ ІНФОРМАЦІЇ

Аналіз великих масивів текстової інформації з використанням ключових слів та фраз є важливою задачею у багатьох галузях, таких як маркетинг, наука, медицина та інші. Це викликає потребу у розробленні методів та інструментів для ефективного та швидкого аналізу тексту з метою виявлення корисної інформації, що може бути використана для прийняття рішень. У цьому розділі будуть розглянуті різні підходи та методи для аналізу великих масивів текстової інформації з використанням ключових слів та фраз. Також будуть представлені інструменти, які використовуються для реалізації цих методів, та їх функціонал. Розглянемо технології, методи та підходи до аналізу текстів з використанням ключових слів та фраз більш детально.

1.1 Основні поняття та терміни

Аналіз великих масивів текстової інформації з використанням ключових слів та фраз включає в себе велику кількість термінів, які перед початком роботи необхідно розглянути. Існує багато технологій, методів та підходів до аналізу текстів з використанням ключових слів та фраз, основні з них:

- обробка даних – спочатку необхідно провести попередню обробку тексту, яка може включати токенізацію (розділення тексту на окремі слова або токени), лематизацію (перетворення слова до його базової форми) та стемінг (відкидання закінчень слів для зведення їх до основного вигляду);
- Natural Language Processing – це область комп'ютерних наук, яка вивчає, як комп'ютер може розуміти та аналізувати людську мову; NLP включає в себе різноманітні методи та технології для обробки природної мови, такі як синтаксичний та семантичний аналіз, машинний переклад, генерація тексту, розпізнавання мови та інші [1];

- Баєсова класифікація – цей метод може використовуватись для автоматичного визначення того, чи містить текст певні ключові слова або фрази, він базується на використанні статистичних методів і вирішує задачу класифікації, визначаючи, чи належить документ до певного класу [2];
- Term Frequency-Inverse Document Frequency – цей метод використовується для оцінки важливості кожного слова в тексті відносно всього набору документів; він враховує частоту вживання слова в документі та зважує його значення на основі частоти вживання слова в інших документах [3];
- Кластеризація – цей метод може використовуватись для групування документів за їх вмістом та ключовими словами; кластеризація може бути проведена на основі багатьох алгоритмів, таких як K-середніх, ієрархічна кластеризація, DBSCAN та інших [4];
- Word2Vec – цей метод використовує нейронні мережі для створення векторних подань слів, вектори слова можуть використовуватись для порівняння слів за семантикою та знаходження схожих слів;
- Deep Learning – це підгалузь машинного навчання, яка використовує нейронні мережі з багатьма шарами для розв'язання складних задач, Deep Learning використовується для класифікації тексту, створення різноманітних моделей машинного навчання та багато іншого.

Вибір підходу до аналізу великих масивів текстової інформації залежить від конкретної задачі та потреб користувача. Комбінація декількох методів може дати кращий результат, ніж застосування одного методу.

Аналіз великих масивів текстової інформації з використанням ключових слів та фраз є важливим інструментом для отримання корисної інформації з великих обсягів текстових даних. Деякі з основних причин використання цього типу аналізу включають:

- розуміння думки та настроїв – аналіз великих масивів текстової інформації з використанням ключових слів та фраз може допомогти в

розумінні думки та настроїв людей щодо певних тем, товарів чи послуг; наприклад, він може бути використаний для аналізу соціальних медіа, оглядів товарів та рецензій на книги, щоб зрозуміти думку людей про певний продукт;

- виявлення трендів – аналіз великих масивів текстової інформації з використанням ключових слів та фраз може бути використаний для виявлення трендів та патернів у певній галузі;
- виявлення проблем – аналіз великих масивів текстової інформації може бути використаний для виявлення проблем та слабких місць у продуктах або послугах компанії, це може допомогти компанії вдосконалювати свої продукти та підвищувати рівень задоволеності клієнтів;
- підвищення ефективності маркетингу – аналіз великих масивів текстової інформації може допомогти компанії визначити ефективність своїх маркетингових кампаній та прогнозувати результати майбутніх кампаній.

Загалом, аналіз великих масивів текстової інформації з використанням ключових слів та фраз може допомогти компаніям та організаціям зрозуміти своїх клієнтів, виявити проблеми та слабкі місця у продуктах і послугах, а також прогнозувати майбутні тренди та результати маркетингових кампаній. Цей тип аналізу може бути корисним для різних галузей, включаючи маркетинг, соціологію, політику, медіа, науку та технології. Застосування різноманітних інструментів та алгоритмів для аналізу текстових даних може допомогти зробити корисні висновки та прийняти важливі рішення для підвищення ефективності та конкурентоспроможності бізнесу.

Необхідно також не забувати про попередню обробку текстових даних. Перед аналізом тексту необхідно провести попередню обробку для покращення якості результатів аналізу. Загалом, цей процес включає в себе наступні етапи:

- токенізація – розбиття тексту на окремі слова (токени) або фрази (n-грами), токени можуть бути розділені за допомогою пробілів, розділових знаків або регулярних виразів [5];

- лематизація – приведення слова до його нормальної форми (леми), наприклад, слова «бігти», «бігає», «бігаємо» будуть перетворені на лему «бігти»;
- стемінг – приведення слова до його основної форми (стему) шляхом відкидання закінчень та інфіксів, наприклад, слова «бігти», «бігає», «бігаємо» будуть перетворені на стем «біг»;
- вилучення стоп-слів – вилучення слів, які зазвичай не несуть суттєвої інформації про контекст тексту, таких як «і», «та», «як», «це» та інші;
- векторизація – перетворення тексту на числові вектори, що можуть бути використані для подальшого аналізу.

Ці етапи попередньої обробки можуть відрізнятися в залежності від конкретної задачі та типу даних, які аналізуються. Однак вони є важливими для досягнення якісних результатів при аналізі тексту.

1.2 Методи та алгоритми аналізу тексту

В сучасному світі, коли величезні об'єми даних зберігаються в електронному вигляді, аналіз тексту стає все більш важливим завданням для підприємств та наукових установ. Текстова інформація може містити цінну інформацію про думки та відгуки клієнтів, погляди на політичні питання, відгуки про товари та послуги, а також багато іншого.

У зв'язку з цим, існує безліч методів та алгоритмів для аналізу текстової інформації, які можуть допомогти зрозуміти значення та важливість даних, які містяться в тексті. Ці методи та алгоритми можуть допомогти визначити питання, які є найбільш важливими для клієнтів, виявити нові тенденції на ринку, підвищити рівень задоволеності клієнтів та покращити бізнес-стратегії. У даному розділі буде розглянуто основні методи та алгоритми аналізу тексту, такі як переробка даних, обробка природної мови, Баєсова класифікація, Term Frequency-Inverse Document Frequency, кластеризація, машинне навчання, метод векторного представлення слів та Deep Learning.

Обробка даних (data processing) – це процес збору, обробки та аналізу даних з метою отримання корисної інформації. Він може використовуватися для аналізу великих масивів текстової інформації з використанням ключових слів та фраз [6].

Один з найпоширеніших методів обробки даних для аналізу текстових даних – це навчання моделей машинного навчання. Для цього використовуються алгоритми навчання з учителем та без учителя, такі як нейронні мережі, різні методи кластеризації, виявлення тем, використання природної мови та інші.

Для виконання аналізу текстових даних спочатку потрібно зібрати та обробити великі масиви даних. Після цього можна використовувати різні методи машинного навчання для аналізу цих даних та виявлення ключових слів та фраз. Наприклад, можна використовувати алгоритми кластеризації для групування текстових даних за схожими темами. Далі можна використовувати природну мову та аналізувати використання ключових слів та фраз в кожному кластері. Це дозволяє виявити тенденції та тренди у великих масивах текстових даних та зробити висновки про поведінку та потреби споживачів [7].

Обробка даних є потужним інструментом аналізу великих масивів текстової інформації з використанням ключових слів та фраз. Вона дозволяє компаніям та організаціям отримувати цінну інформацію для прийняття важливих рішень та покращення бізнесу. Також до методів обробки даних для аналізу текстових даних можуть відноситися і методи обробки мовленнєвих даних (Natural Language Processing або NLP) [8]. Вони дозволяють розуміти зміст текстів, знаходити зв'язки між словами, здійснювати машинний переклад, виконувати аналіз емоцій тощо.

Усі ці методи обробки даних дозволяють ефективно аналізувати великі масиви текстової інформації та виявляти ключові слова та фрази, що дозволяє зробити висновки щодо тем, що цікавлять споживачів або тенденцій на ринку. В результаті, компанії можуть покращувати свої продукти, пропонувати нові рішення та збільшувати свою прибутковість.

Метод обробки даних включає в себе всі кроки, необхідні для очищення та підготовки даних до подальшого аналізу. В загальному вигляді цей процес включає в себе наступні етапи:

- збір даних – цей етап включає збір даних з різних джерел, таких як бази даних, веб-сторінки, API, файлові системи, тощо;
- очищення даних включає видалення некоректних, неповних, дубльованих або несуттєвих даних, цей процес може також включати видалення шуму та вирівнювання даних;
- трансформація даних – перетворення даних у формат, зрозумілий для подальшого аналізу, наприклад, зменшення розмірності даних, видалення стоп-слів, лематизацію, тощо;
- інтеграція даних – об'єднання даних з різних джерел у єдину базу даних;
- валідація даних, на даному етапі дані проходять перевірку на відповідність певним стандартам та правилам;
- аналіз даних – цей етап включає застосування методів машинного навчання, статистичного аналізу, графічного аналізу, тощо для отримання нових знань з даних;
- візуалізація даних включає створення графіків, діаграм та інших візуальних засобів для представлення даних у зрозумілій формі.

Процес обробки даних може бути виконаний вручну або автоматизований за допомогою різноманітних програмних засобів, таких як Python, RapidMiner, IBM Watson та інші.

Наступним методом є обробка природної мови. Це напрямок штучного інтелекту, який дає комп'ютерам можливість автоматично отримувати сенс із природного, створеного людиною тексту. Цей напрямок вивчає взаємодію між людиною та комп'ютером з використанням мови, що використовується людиною. Вона використовує лінгвістичні моделі та статистику для навчання технології глибокого навчання обробці та аналізу текстових даних, включаючи зображення рукописного тексту. Основною метою обробки природної мови є розуміння, аналіз та генерація природної мови з метою поліпшення взаємодії людей з

комп'ютерами. Наприклад, такий метод обробки природної мови, як оптичне розпізнавання символів (OCR), перетворює зображення тексту на текстові документи, знаходячи і розпізнаючи слова на зображеннях [9].

Natural Language Processing – це галузь комп'ютерної лінгвістики, яка займається обробкою та аналізом мовленнєвої інформації з метою розуміння її змісту та контексту. Використання NLP у аналізі великих масивів текстової інформації з використанням ключових слів та фраз дозволяє зрозуміти зміст тексту, визначити його тему та виявити ключові терміни та відносини між ними [10].

Один з основних інструментів NLP – це аналізатор частин мови, який визначає, які слова є іменниками, дієсловами, прикметниками тощо. Це дозволяє виявляти зв'язки між словами та відносини між ними.

Інший інструмент NLP – це аналізатор синтаксичних залежностей, який дозволяє визначити зв'язки між словами у реченні. Наприклад, він може визначити, що слово «кіт» є підметом у реченні «Кіт біжить». Цей інструмент дозволяє виявляти зв'язки між словами та зрозуміти їхню роль у тексті.

Ще один інструмент NLP – це аналізатор тональності, який дозволяє визначити, чи містить текст позитивну, негативну або нейтральну оцінку. Це може бути корисно для відстеження настроїв споживачів щодо продукту або бренду.

Усі ці інструменти NLP можуть бути використані для аналізу великих масивів текстової інформації з використанням ключових слів та фраз. Вони дозволяють зрозуміти зміст текстів, виявити теми, які цікавлять споживачів, та виявити ключові слова та відносини між ними.

Одним з прикладів використання NLP є аналіз тональності текстів. Це означає виявлення позитивної, негативної або нейтральної оцінки змісту тексту. Цей метод може бути корисним для відстеження настроїв споживачів щодо продукту або бренду. Іншим прикладом використання NLP є аналіз синтаксичних залежностей в текстах. Це означає виявлення зв'язків між словами та визначення

їх ролі в реченні. Цей метод може бути також корисним для покращення автоматичного розпізнавання мовлення або для виявлення патернів у текстах.

В результаті застосування NLP у аналізі великих масивів текстової інформації з використанням ключових слів та фраз, компанії можуть отримати ряд переваг:

– розуміння потреб та побажань споживачів – аналіз текстової інформації, яку залишають споживачі (відгуки, коментарі, питання тощо), може допомогти компанії зрозуміти, які продукти чи послуги потрібні клієнтам, що їх задовольняє або навпаки – не задовольняє;
– покращення кількості продажів – за допомогою аналізу ключових слів та фраз, які зустрічаються у текстах, компанії можуть зрозуміти, які терміни найбільш привертають увагу споживачів, це може допомогти виробникам покращити маркетингові стратегії, зокрема, ключові слова та фрази, які використовуються у рекламі, описах продуктів тощо;
– покращення якості продуктів та послуг – аналіз відгуків та коментарів споживачів дозволяє компанії знати, що саме потрібно покращити у своїх продуктах чи послугах, наприклад, компанія може виявити, що багато споживачів скаржаться на низьку якість пакування продукту, тому її можна покращити для задоволення клієнтів;
– зменшення ризику втрати клієнтів – швидке реагування на питання та проблеми клієнтів може допомогти компанії зберегти їх вірність; аналіз відгуків та коментарів дозволяє компанії виявляти проблеми та незадоволення клієнтів та швидко виправляти їх.

Отже, застосування NLP у аналізі великих масивів текстової інформації з використанням ключових слів та фраз може допомогти компаніям збільшити ефективність своїх бізнес-процесів та покращити взаємодію зі своїми клієнтами. Він дозволяє отримати більш детальну та точну інформацію про те, як клієнти сприймають продукти та послуги, що дозволяє компаніям покращувати свої продукти, маркетингові стратегії та обслуговування клієнтів.

Наприклад, NLP може допомогти виявити ключові терміни та теми, які найбільш часто згадуються у відгуках та коментарях клієнтів, що дозволяє компанії побачити, що саме цікавить та турбує їх клієнтів. Крім того, NLP може допомогти автоматично обробляти великі обсяги тексту, що зменшує час та зусилля, які необхідно витратити на аналіз. Однак, варто зазначити, що NLP має свої обмеження. Наприклад, він може не завжди правильно розуміти контекст та вирази, що використовуються в тексті, а також може виявляти неточності у роботі з мовами, які мають складну граматику чи словник. Тому перед застосуванням NLP у аналізі великих масивів текстової інформації необхідно враховувати його обмеження та проводити необхідні перевірки результатів.

Для реалізації обробки природної мови використовуються спеціалізовані програмні бібліотеки, такі як Python Natural Language Toolkit, IBM Watson Natural Language Understanding, Google Cloud Natural Language API та інші. Вибір конкретної бібліотеки залежить від потреб користувача та області застосування.

Це одним методом аналізу тексту є Баєсова класифікація. Це метод машинного навчання, який використовує статистичну модель для визначення класу об'єкту на основі ймовірності належності до певного класу. Цей метод базується на теоремі Баєеса, яка встановлює зв'язок між умовною і безумовною ймовірностями [11]. Баєсова класифікація є ефективним методом для аналізу великих масивів текстової інформації з використанням ключових слів та фраз та використовується для класифікації текстів на основі ймовірності належності до певного класу.

В аналізі великих масивів текстової інформації Баєсова класифікація може бути використана для автоматичної класифікації документів за темою, заснованою на наборі ключових слів та фраз. Для цього спочатку створюється модель, яка враховує ймовірності того, що певні слова або фрази використовуються у тексті певної тематики. Потім, коли з'являється новий документ, модель розраховує ймовірності того, що документ належить до кожного з можливих класів на основі використаних слів та фраз. Документ класифікується як належний до класу з найбільшою ймовірністю.

Один з найбільш поширених варіантів Баєсової класифікації – наївний Баєсовий класифікатор (Naive Bayes Classifier). Цей класифікатор передбачає, що кожен атрибут (тобто слово або фраза) у документі незалежний від інших атрибутів. Це, звичайно, не завжди точно, але такий підхід дозволяє спростити розрахунки та прискорити процес класифікації.

У Баєсової класифікації об'єкти представляються у вигляді векторів ознак, які можуть бути числовими, категоріальними або бінарними. Для кожного класу створюється статистична модель, яка описує розподіл ймовірностей ознак у цьому класі [12]. Після цього для нового об'єкту визначається ймовірність належності до кожного класу, і об'єкт призначається до класу з найбільшою ймовірністю.

Один з основних недоліків наївного Баєсового класифікатора полягає в тому, що він не враховує залежності між словами. Наприклад, в розглянутому вище прикладі з класифікацією спаму, якщо слова «ліки» і «рецепт» з'являться разом у повідомленні, то це може бути ознакою того, що повідомлення є спамом. Але Наївний Баєсовий класифікатор буде розглядати ці два слова як незалежні атрибути.

Для врахування залежностей між словами можна використовувати більш складні моделі класифікації, такі як Багатовимірна Баєсова класифікація (Multivariate Bayesian Classification) або Умовна випадкова поля (Conditional Random Fields) [13]. У більш складних моделях класифікації зазвичай використовуються більш широкий набір атрибутів, таких як морфологічні та синтаксичні особливості тексту, стилістичні ознаки тощо. Це дозволяє покращити точність класифікації та зменшити кількість помилок.

Баєсова класифікація є потужним методом для аналізу великих масивів текстової інформації з використанням ключових слів та фраз. Вона може бути використана для класифікації текстів за тематикою, розпізнавання образів, визначення тональності текстів або для виявлення спаму. Важливо пам'ятати, що використання Баєсової класифікації потребує попереднього створення моделі та навчання її на тренувальних даних. Цей метод є досить простим і ефективним, особливо якщо кількість ознак невелика.

В аналізі тексту також часто використовують такий метод як Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF – це метод обробки текстів, який дозволяє оцінювати важливість термінів в документі або колекції документів. Він широко використовується у задачах аналізу великих масивів текстової інформації з використанням ключових слів та фраз. Метод обробки текстів за оцінкою важливості термінів є одним з найбільш поширених методів у прикладній лінгвістиці та обробці природної мови [14]. Основна ідея методу полягає в тому, щоб визначити, які слова або фрази є важливими у тексті, і використовувати цю інформацію для подальшої обробки тексту.

TF-IDF складається з двох компонентів: Term Frequency (TF) та Inverse Document Frequency (IDF). TF – це відношення кількості входжень терміну в документ до загальної кількості слів в документі. Це оцінює важливість терміну у конкретному документі. Чим більше термін зустрічається у документі, тим вище його значення TF. IDF – це оцінка того, наскільки важливим є термін у колекції документів. Це обчислюється як логарифм ділення загальної кількості документів в колекції на кількість документів, які містять термін. Чим менше документів містять термін, тим вище його значення IDF.

TF-IDF – це добуток значень TF та IDF. Це дає оцінку важливості терміну у документі в порівнянні з іншими документами у колекції. Терміни з високим значенням TF-IDF є важливими для конкретного документу, а терміни з низьким значенням TF-IDF можуть бути загальними термінами, які не мають великої важливості для конкретного документу [15].

Існують різноманітні програмні інструменти для обробки тексту за оцінкою важливості термінів, серед них:

- Python Natural Language Toolkit – це відкрите програмне забезпечення для обробки природної мови з великою кількістю функцій, включаючи функції для обчислення TF-IDF;
- Gensim – це бібліотека Python для обробки природної мови з функціями для обчислення TF-IDF та інших методів визначення важливості термінів;

– RapidMiner – це програмне забезпечення для аналізу даних, яке має вбудований модуль для обробки природної мови та функції для обчислення TF-IDF та інших методів визначення важливості термінів.

Використання методів обробки тексту за оцінкою важливості термінів дозволяє автоматично визначати ключові слова та фрази в тексті, що допомагає здійснювати аналіз та класифікацію документів, з окремими абзацами та реченнями.

Загалом, методи обробки тексту за оцінкою важливості термінів дозволяють автоматично визначати ключові слова та фрази у тексті, що допомагає здійснювати аналіз та класифікацію документів, зокрема у сферах маркетингу, наукових досліджень, реклами, фінансів, медицини та багатьох інших галузях, де обробка текстів є необхідною. TF-IDF може бути використаний для багатьох задач, таких як класифікація документів, підрахунок схожості документів та визначення ключових слів у документах. Він також може бути використаний для зменшення розміру колекції документів, видаляючи менш важливі документи за допомогою порогового значення TF-IDF.

Кластеризація є одним з методів аналізу великих масивів текстової інформації з використанням ключових слів та фраз. Кластеризація – це процес групування об'єктів в підмножини (кластери) на основі схожості між ними. У контексті аналізу великих масивів текстової інформації з використанням ключових слів та фраз, кластеризація може допомогти виявляти спільні теми, групи слів та фраз, та підмножини текстів з схожими властивостями [16]. Цей метод дозволяє групувати подібні тексти разом у кластери за допомогою алгоритмів машинного навчання. Кластеризація може бути корисною для виявлення тематичних груп текстів, знаходження спільних слів та фраз у великому масиві текстів і подальшого аналізу цих груп. Кластеризація в аналізі тексту є методом групування подібних документів разом на основі спільних характеристик. Це може бути корисним для зведення до мінімуму складності аналізу великої кількості документів та для здійснення групування за темою, настроєм, змістом тощо.

Існує кілька методів кластеризації текстових даних, таких як ієрархічна кластеризація, кластеризація на основі k-середніх (k-means clustering), агломеративна кластеризація та інші. Кожен з цих методів має свої переваги та недоліки, і вибір конкретного методу залежить від характеристик даних та мети дослідження [17].

У кластеризації текстових даних ключові слова та фрази використовуються для визначення схожості текстів. Для цього можуть бути використані різні методи відображення текстів у векторний простір, такі як TF-IDF та Word2Vec. Одним із способів кластеризації текстів є використання косинусної схожості між векторами, що представляють тексти.

Кластеризація може бути корисна для різних застосувань, таких як:

– класифікація статей на сайті за темами;
– групування соціальних медіа дописів за темою та змістом;
– аналіз користувацьких відгуків про товари та послуги за тоном та тематикою.

В цілому, кластеризація є потужним інструментом для аналізу великих масивів текстової інформації та може допомогти виявляти та розуміти спільності та відмінності в текстах. Застосування кластеризації може бути корисним в наступних ситуаціях:

– сегментація аудиторії – кластеризація може допомогти в розумінні інтересів та поведінки різних груп аудиторії, наприклад, якщо компанія має декілька продуктів, то кластеризація даних про покупців може допомогти виявити, які групи аудиторії віддають перевагу яким продуктам;
– аналіз конкурентів – кластеризація може бути корисною при аналізі конкурентів та виявленні певних груп компаній, які конкурують у певних ринкових сегментах;
– пошук злочинців – кластеризація може допомогти виявити схожість між кримінальними діями та злочинцями, що може допомогти в розслідуванні кримінальних справ;

– семантичний аналіз – кластеризація може допомогти виявити спільні теми та ідеї у текстах, що може бути корисним при аналізі контенту, наприклад, у медіа аналітиці.

Застосування кластеризації може допомогти виявляти певні закономірності та залежності в даних, що дозволяє компаніям приймати кращі рішення на основі аналізу великих масивів текстової інформації.

Основні аспекти кластеризації включають:

– вибір метрики схожості – метрика схожості визначає, як будуть порівнюватися елементи і яка кількість схожості буде вважатися достатньою для їх включення до одного кластеру, для текстової інформації, часто використовують косинусну схожість, яка враховує кількість спільних слів та їх вагу у текстах;
– вибір алгоритму кластеризації – існує багато алгоритмів кластеризації, таких як K-середніх, ієрархічна кластеризація, DBSCAN та інші, вибір алгоритму залежить від типу даних та завдання кластеризації;
– вибір кількості кластерів – кількість кластерів визначається в залежності від завдання кластеризації та алгоритму, для деяких алгоритмів, таких як K-середніх, кількість кластерів повинна бути заздалегідь задана, для інших алгоритмів, таких як ієрархічна кластеризація, кількість кластерів може бути визначена за допомогою dendrogram;
– вибір представлення даних – для того, щоб використовувати алгоритми кластеризації, необхідно перетворити текстову інформацію у числове представлення, для цього можна використовувати методи, такі як TF-IDF або Word2Vec;
– оцінка якості кластеризації – оцінка якості кластеризації є важливою, оскільки допомагає зрозуміти, наскільки ефективно було проведено кластеризацію, для оцінки якості кластеризації використовують різні метрики, залежно від мети кластеризації та характеру даних.

Кластеризація в аналізі тексту може бути застосована для різних завдань, таких як розуміння структури колекції документів, групування користувачів за спільними інтересами та виявлення подібних тематик у великому обсязі даних.

Для кластеризації в аналізі тексту використовуються різні програмні засоби, такі як Python Natural Language Toolkit (NLTK), RapidMiner, IBM Watson Natural Language Understanding, Google Cloud Natural Language API та інші.

Узагальнюючи, кластеризація є потужним інструментом для аналізу великих масивів текстової інформації з використанням ключових слів та фраз, оскільки вона дозволяє автоматично групувати текстові документи на основі схожості їх змісту. Вона є потужним інструментом для аналізу текстової інформації, який дозволяє швидко та ефективно здійснювати організацію та аналіз великих масивів даних.

Метод векторного представлення слів є одним з основних методів в області обробки природної мови та машинного навчання, який використовується для аналізу великих масивів текстової інформації з використанням ключових слів та фраз. Цей метод полягає в тому, що кожне слово або фраза у тексті представляється у вигляді вектора чисел, де кожне число відображає важливість певної характеристики слова чи фрази. Один з найбільш популярних методів векторного представлення слів – Word2Vec. Це метод векторного представлення слів (word embedding), який використовується в обробці природньої мови. Word2Vec використовує нейромережеву архітектуру, яка навчається на великій кількості текстових даних, з метою знаходження взаємозв'язку між словами та відображення їх у векторному просторі. Завдяки Word2Vec можливо представити слова у вигляді числових векторів, що дозволяє комп'ютерам більш розуміти смислові взаємозв'язки між словами [18].

Word2Vec побудований на ідей, що слова, які часто зустрічаються поруч, мають схожі значення. Отже, для побудови векторного представлення слів, Word2Vec використовує методи навчання нейронних мереж, що дозволяє визначити ймовірності зустрічі слів поруч. Наприклад, для речення «my dad is lying on the bed», Word2Vec навчиться визначати, що слово «dad» часто

з'являється поруч зі словом «bed», тому вектор для «dad» і «bed» буде близьким за значенням. Отримані векторні представлення слів можуть бути використані для багатьох завдань в обробці природньої мови, таких як зведення даних, кластеризація текстів, знаходження синонімів та аналогів слова, машинний переклад, розпізнавання іменованих сутностей та багато іншого.

Word2Vec є дуже потужним методом в обробці природньої мови і може бути застосований в багатьох ситуаціях. Ось деякі з них:

– кластеризація текстів – Word2Vec може допомогти згрупувати схожі тексти за змістом, використовуючи векторні представлення слів, наприклад, це може бути корисним для групування новинних статей за темою;
– пошук синонімів та аналогів слів – завдяки Word2Vec можна знайти інші слова, які мають схожі значення або контекст використання, наприклад, можна знайти синоніми до слова «гарячий» (наприклад, «спекотний», «теплий», «жаркий» тощо) або аналоги до слова «автомобіль» (наприклад, «машину», «транспортний засіб», «вантажівку» тощо);
– переклад текстів – Word2Vec може бути використаний для автоматичного перекладу текстів з однієї мови на іншу, шляхом перетворення слів у векторні представлення і перетворення їх назад на слова у цільовій мові;
– аналіз настрою – Word2Vec може допомогти виявити настрій тексту, визначаючи емоційну забарвленість слова в контексті речення або абзацу.

Це лише кілька з прикладів використання Word2Vec, його можна використовувати в багатьох інших ситуаціях, де потрібно аналізувати великі масиви текстової інформації. Використання методу векторного представлення слів дозволяє здійснювати широкий спектр завдань в аналізі тексту, таких як класифікація текстів, пошук схожих документів, зведення текстів до заголовків, розпізнавання іменованих сутностей та інші.

Машинне навчання широко використовується в аналізі тексту. Це метод, що використовується для побудови моделей, які можуть класифікувати документи на основі їхнього змісту та використання ключових слів та фраз. Машинне навчання може бути використане для автоматичного розподілу документів за тематикою, виявлення семантичних зв'язків між документами та інших завдань. Зокрема, машинне навчання може бути використане для класифікації текстів, кластеризації документів, визначення настрою або емоцій текстів, визначення тематики текстів, а також для багатьох інших завдань, пов'язаних з аналізом тексту. Воно відіграє важливу роль в аналізі текстової інформації, дозволяючи автоматизувати процес обробки та аналізу великої кількості текстових даних, що дозволяє зробити більш точні та ефективні висновки з наданої інформації.

Deep Learning – це підгалузь машинного навчання, яка використовує нейронні мережі з багатьма шарами для розв'язання складних задач. У контексті обробки текстової інформації, Deep Learning може використовуватись для класифікації тексту, витягування інформації з тексту, створення різноманітних моделей машинного навчання та багато іншого [19]. Deep Learning є потужним інструментом для аналізу великих масивів текстової інформації з використанням ключових слів та фраз. Застосування Deep Learning у NLP дає змогу покращити точність класифікації, кластеризації та прогнозування результатів.

Один з основних методів Deep Learning у NLP – це рекурентні нейронні мережі (RNN). Вони можуть бути використані для аналізу послідовностей текстової інформації, таких як речення та абзаци, та виявлення залежностей між ними. RNN також можуть використовуватись для генерації тексту, перекладу мови та створення чат-ботів [20].

Ще один метод Deep Learning у NLP – це згорткові нейронні мережі (CNN). Вони зазвичай використовуються для аналізу коротких текстових фрагментів, таких як заголовки статей або короткі описи [21]. Використання CNN може допомогти покращити точність класифікації та виявлення ключових слів та фраз.

Також, можуть використовуватись автокодувальні нейронні мережі, які зазвичай використовуються для розпізнавання та зменшення розміру вхідних

даних. Вони можуть бути використані для створення зведених представлень текстів, що можуть допомогти в подальшому аналізі та кластеризації.

Застосування Deep Learning у аналізі великих масивів текстової інформації з використанням ключових слів та фраз дозволяє отримати більш точні результати та прогнози, що можуть мати значний вплив на прийняття рішень в бізнесі та інших сферах діяльності.

Машинне навчання та Deep Learning є потужними інструментами для аналізу великих масивів текстової інформації з використанням ключових слів та фраз. Ці методи можуть бути використані для розв'язання різноманітних задач, включаючи класифікацію документів, кластеризацію, визначення настрою, створення рекомендацій та багато іншого.

Один з найбільш популярних методів машинного навчання – це навчання з вчителем, який використовується для класифікації документів та інших задач. У навчанні з вчителем модель навчається на наборі даних, які вже містять мітки класів. Модель використовує ці дані, щоб навчитися розпізнавати певні ознаки тексту, які пов'язані з конкретним класом. Наприклад, якщо ми хочемо створити модель для класифікації статей за темами, ми можемо навчити модель розпізнавати ключові слова та фрази, які пов'язані з кожною темою, такі як «спорт», «політика», «наука» тощо. Модель буде використовувати ці ознаки для класифікації нових статей за темою.

Deep Learning є більш потужним методом машинного навчання, який використовує нейронні мережі з багатьма шарами для розпізнавання складних ознак у тексті. Цей підхід може бути особливо корисним для вирішення задач, які пов'язані з розумінням природної мови, таких як розпізнавання емоцій або зв'язків між словами та фразами.

Щодо перекладу мови, то глибоке навчання також може бути використане для автоматичного перекладу. Одним з найпопулярніших методів є використання енкодера-декодера на основі рекурентних нейронних мереж. В цьому підході, мережа енкодує вихідний текст у векторну форму, а потім декодує його в

перекладений текст [22]. Можна використовувати додаткові методи, такі як attention механізми, щоб забезпечити кращу якість перекладу.

У аналізі тексту, машинне навчання та Deep Learning можуть бути використані для розпізнавання тематики документів, класифікації текстів, виявлення емоцій, аналізу тональності, визначення іменованих сутностей та багато іншого. Наприклад, використання нейронних мереж може допомогти виявляти складні зв'язки між словами та фразами, що допомагає в покращенні якості аналізу тексту. В цілому, застосування Deep Learning у текстовому аналізі дає змогу досягти високої точності та ефективності в різних задачах, таких як класифікація, кластеризація, генерація тексту та переклад мови.

1.3 Технології візуалізації результатів аналізу текстової інформації

Візуалізація є важливою складовою процесу аналізу текстової інформації, що дозволяє отримати цінну інформацію з великих масивів тексту та зробити корисні висновки. У зв'язку з цим, розробники створюють різноманітні інструменти та технології для візуалізації результатів аналізу текстових даних. Серед таких інструментів значне місце займає мова програмування Python, яка має велику кількість бібліотек та інструментів для візуалізації даних. В цьому розділі буде розглянуто деякі з найпоширеніших технологій та інструментів для візуалізації результатів аналізу текстової інформації. Візуалізація аналізованого тексту може бути дуже корисною для розуміння змісту текстових даних та отримання корисних висновків. Вона дозволяє представити текст у вигляді:

- word clouds (хмари слів) – це візуалізаційний інструмент, що використовують для відображення найбільш повторюваних слів або фраз у тексті, слова розміщуються на зображенні залежно від їх частоти в тексті, тому більші слова відображають більш часто зустрічаються слова;
- графіки та діаграми можуть використовуватися для відображення залежностей та тенденцій у текстових даних, наприклад, графік може відображати зміни частоти вживання слів чи фраз у тексті з часом;

- heat maps (теплові карти) відображає частоту вживання слів чи фраз у тексті за допомогою кольорової шкали, таким чином, можна швидко побачити найбільш повторювані слова або фрази у тексті;
- network graphs (мережеві графи) відображають взаємозв'язки між словами, фразами чи іншими елементами у тексті, такі графи можуть допомогти зрозуміти структуру та зв'язки між різними аспектами тексту.

Ці інструменти можуть бути використані окремо або в комбінації, залежно від мети та потреб аналізу даних, вони роблять аналіз більш зрозумілим та доступним для подальшого дослідження. Загалом, візуалізація дозволяє швидко виявляти закономірності, тенденції та зв'язки між різними аспектами тексту. Наприклад, в хмарі слів можна побачити слова та фрази, що найбільш часто зустрічаються у тексті, що може допомогти зрозуміти тему та настрій тексту. Графіки та діаграми можуть допомогти відслідковувати зміни в тексті з часом, виявляти залежності між різними аспектами тексту та зробити інші висновки, які можуть бути важливими для прийняття рішень або подальшого дослідження.

Python є однією з найпопулярніших мов програмування для візуалізації результатів аналізу текстової інформації. Python має багато бібліотек, які сприяють ефективній та зручній візуалізації даних. Одна з найбільш популярних бібліотек для візуалізації даних - це Matplotlib. Ця бібліотека надає можливості для побудови різноманітних графіків, включаючи стовпчикові, лінійні, кругові та інші типи графіків та діаграм. Крім того, Matplotlib можна використовувати для побудови теплових карт та 3D-графіків. Інша популярна бібліотека для візуалізації даних - це Seaborn. Вона спрощує створення складних графіків та використання кольорових схем, що забезпечують зрозумілість та інтерпретованість графіків. Також існують спеціалізовані бібліотеки для візуалізації тексту, такі як WordCloud, що створює хмари слів, та TextBlob, яка надає можливості для аналізу настрою тексту. Python також має інтегрований модуль для візуалізації даних – pandas. Він дозволяє зручно візуалізувати дані з різних джерел, включаючи текстові файли. Загалом, Python – це потужний інструмент для візуалізації результатів аналізу текстової інформації, який

дозволяє швидко та зручно отримати графічну інтерпретацію даних та зробити корисні висновки.

Сам візуальний інтерфейс було реалізовано з використанням технологій, як HTML, CSS та JavaScript. Вони є технологіями, що використовуються для створення та оформлення веб-сторінок.

HTML – це мова розмітки, яка використовується для створення веб-сторінок. Вона дозволяє визначати структуру та зміст веб-сторінок, такі як заголовки, абзаци, списки, таблиці, зображення тощо.

CSS – це мова стилів, яка використовується для оформлення веб-сторінок. CSS дозволяє відокремити вміст від оформлення, що дозволяє змінювати вигляд веб-сторінок без зміни їх вмісту. CSS дозволяє задавати кольори, шрифти, розміри, відступи, рамки та інші елементи дизайну веб-сторінок.

JavaScript – це скриптова мова програмування, яка використовується для створення динамічних ефектів на веб-сторінках, таких як анімація, зміна вмісту сторінки без перезавантаження, перевірка форм, валідація вводу користувача тощо. JavaScript також використовується для створення веб-додатків та інших веб-інтерфейсів.

Також для розширення можливостей HTML, CSS та JavaScript було використано Vue.js. Це JavaScript-бібліотека для побудови інтерактивних користувацьких інтерфейсів. Вона дозволяє створювати компоненти з відокремленим логікою та представленням, які можна повторно використовувати. Вона використовується, щоб додати до нього динамічних функцій та можливостей. Наприклад, він може бути використаний для створення динамічних форм, інтерактивних діалогових вікон, валідації введених даних, зміни стилів в залежності від подій та іншого.

Для реалізації візуалізації аналізу у даній роботі було використано мову Python та її вбудовані бібліотеки Matplotlib та WordCloud, що дозволяють швидко та якісно візуалізувати результати аналізу текстової інформації у розроблюваній підсистемі у вигляді хмар слів, різноманітних графіків, діаграм та теплових карт.

Також для цієї потреби було використано HTML, CSS та JavaScript з її бібліотекою Vue.js.

Висновки до розділу 1

В розділі визначено основні поняття та терміни процесу аналізу великих масивів текстової інформації за ключовими словами та фразами та розглянуто різні підходи та методи для розв'язання цієї задачі, включаючи методи оброблення тексту за оцінкою важливості термінів, кластеризацію, векторне представлення слів та машинне навчання та Deep Learning. Підсумовуючи огляд підходів та методів розв'язання задачі аналізу великих масивів текстової інформації з використанням ключових слів та фраз, можна сказати, що це досить складна задача, яка вимагає використання різноманітних методів та інструментів.

Такі методи обробки тексту, як векторне представлення слів, лематизація, токенізація, класифікація, Term Frequency-Inverse Document Frequency та машинне навчання є найбільш ефективними підходами до аналізу текстових даних. Ці методи забезпечують високу точність та швидкість обробки даних, що є дуже важливим для розв'язання таких задач, як аналіз настроїв, визначення тематики тексту, автоматичне визначення ключових слів та фраз. На основі широкого аналізу існуючих методів, обрано методи, що найбільш оптимально підходять для розв’язання задачі аналізу великих масивів текстової інформації. Цими методами є класифікація на основі наївного Баєсового класифікатора, векторизація та Term Frequency-Inverse Document Frequency. Для візуалізації аналізованого тексту обрано мову Python та її вбудовані бібліотеки Matplotlib та WordCloud, а також HTML, CSS та JavaScript з його бібліотекою Vue.js.

Застосування аналізу текстових даних є дуже широким, зокрема в маркетингу, бізнесі, наукових дослідженнях, політиці та інших галузях. Розвиток технологій та поява нових методів та інструментів дозволяє здійснювати більш точний та швидкий аналіз текстових даних, що є дуже важливим для ефективного прийняття рішень та досягнення успіху в різних галузях діяльності.

2 РОЗРОБЛЕННЯ ПІДСИСТЕМИ АНАЛІЗУ ВЕЛИКИХ МАСИВІВ ТЕКСТОВОЇ ІНФОРМАЦІЇ

Аналіз тексту використовується підприємствами для отримання корисних відомостей із безлічі джерел неструктурованих даних. При прийнятті рішень вони використовують зворотний зв'язок із різних джерел: електронної пошти, соціальних мереж та опитувань клієнтів, наприклад, розпізнаний текст з розмов з ними. Однак обробка такого величезного обсягу інформації без відповідного програмного забезпечення стає непосильним завданням, тому програмна реалізація системи аналізу великих масивів текстової інформації стає надзвичайно важливим завданням.

В цьому розділі буде розглянуто важливість безпеки та захищеності даних у цій системі, також буде представлені та описані алгоритми та технології, що будуть використані для оброблення тексту, описано розроблені діаграми, що описують систему, а також реалізовано та протестовано систему.

2.1 Безпека та захищеність даних

Безпека та недоторканість даних являє собою один із найважливіших критеріїв оцінювання готово продукту. Навряд знайдеться індивід, який поставить швидкість чи зручність системи вище від безпеки інформації. Однак якщо кожен користувач здатен оцінити важливість даних, безпосередньо для нього, та не претендувати на їх безпеку, то компанія не може собі дозволити такого. Через що прийнято рішення використовувати Azure AD (Azure Active Directory). Це хмарна служба ідентифікації та управління доступом від Microsoft, що надає можливості аутентифікації та авторизації користувачів, контролю доступу до ресурсів, управління групами користувачів та інших функцій, необхідних для ефективного управління доступом до хмарних та локальних ресурсів. Azure AD дозволяє організаціям керувати доступом користувачів до різних ресурсів, таких як програми, файли та веб-сайти, забезпечувати безпеку доступу та використовувати

багатофакторну аутентифікацію. Крім того, Azure AD може бути використаний як рішення одної точки входу для різних хмарних та локальних сервісів Microsoft, таких як Office 365, Azure, Dynamics 365, та інші. Відповідну взаємодію клієнта з автентифікатором та веб-застосунком продемонстровано на рисунку 1.

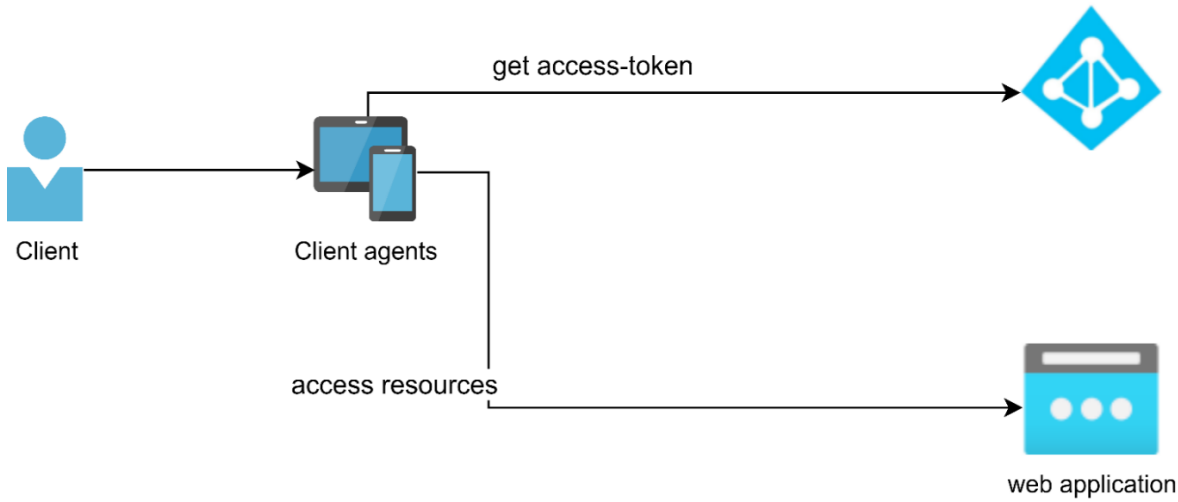

Рисунок 1 – Схема отримання Azure AD токена

Таким чином, перш ніж розпочати роботу в системі, користувач повинен увійти через свій обліковий запис Microsoft. Таке рішення звільняє нас від «розробки велосипеду», так як безпекові рішення гігантів є більш продумані та помірковані. Доступ до будь якого ресурсу застосунку можливий лише при вході до облікового запису конкретної організації, що повністю унеможливлю несанкціоноване отримання бодай якоїсь корисної інформації, шляхом перебору посилань.

Специфіка обробки великим масивів текстових даних полягає в тому, що дані є великим. Відповідно до цього типові реляційні бази даних можуть не підходити по ряду причин. Зберігання великих обсягів даних в реляційних базах даних може бути дорогим і важким для масштабування. Реляційні бази даних можуть мати обмеження на кількість записів, що можуть бути збережені в таблицях. При великому об'ємі даних звернення до бази даних може бути повільним через обмеження швидкості обробки даних та обмеження ресурсів системи. Реляційні бази даних можуть бути складними для запитів і аналізу

великих масивів текстових даних, оскільки вони розроблені для обробки структурованих даних. З цих причин звичайні реляційні бази даних можуть не бути ефективним вибором для зберігання великих масивів текстових даних.

Відповідно до описаних вище проблем, є доцільним зберігання текстового файлу в Azure Blob Storage. Azure Blob Storage є розподіленою об'єктною системою зберігання, яка може зберігати дуже великі обсяги даних в хмарному середовищі. Blob Storage дозволяє зберігати та керувати мільйонами об'єктів даних, включаючи текстові файли, у форматі блобів. Azure Blob Storage має багато функцій, що роблять його підходящим для зберігання великих масивів текстових даних, зокрема:

- масштабованість – Azure Blob Storage може легко масштабуватися в залежності від потреб вашого бізнесу, надаючи можливість зберігати великі обсяги даних;
- висока доступність – система забезпечує високий рівень доступності та надійності даних, тому що дані реплікуються автоматично для захисту від втрати даних;
- розширені функції безпеки – система надає багато варіантів для налаштування безпеки даних, включаючи рівень доступу, шифрування та ідентифікацію користувачів;
- швидкий доступ до даних – Azure Blob Storage дозволяє швидко отримувати доступ до даних, незалежно від їх обсягу.

2.2 Вибір та опис алгоритмів оброблення тексту

Аналіз тексту дозволяє швидко отримувати точну інформацію. До того ж цей процес повністю автоматизований і послідовний, а результати стають основою прийняття рішень. Наприклад, використання програмного забезпечення для аналізу тексту дозволяє миттєво виявляти негативні тональності в повідомленнях соціальних мереж та вчасно вживати заходів щодо вирішення проблеми, або виявлення інших проблем.

Аналіз великих масивів текстової інформації включає наступні етапи:

– збір даних – збір великих масивів текстових даних може бути виконаний за допомогою веб-скрапінгу, API, баз даних, соціальних медіа тощо;
– попередній аналіз – цей етап включає очищення даних від зайвих символів та форматування тексту для подальшого аналізу, також може виконуватись попереднє визначення тематики тексту та розподіл даних за категоріями;
– токенізація – це процес розбиття тексту на окремі слова (токени), щоб легше аналізувати даний текст,
– лематизація – це процес приведення слів до їх базової форми (леми), щоб уникнути дублювання інформації;
– виділення ключових слів та фраз – цей етап включає виділення найбільш важливих слів та фраз у тексті, що можуть допомогти в зрозумінні змісту тексту;
– кластеризація – це процес групування схожих текстів разом, щоб знайти спільні теми та визначити релевантність даних для конкретної тематики;
– класифікація – це процес визначення категорії, до якої належить даний текст, це може бути використано для визначення сутностей, позначок, тональності тощо;
– візуалізація може допомогти в зрозумінні структури та зв'язків між різними текстами, вона може використовуватись для створення графіків, хмар тегів, діаграм та інших візуальних представлень даних;
– видобуток інформації – на цьому етапі використовуються методи NLP, щоб видобути ключову інформацію з тексту, таку як імена, дати, місця, ключові слова та фрази, це може допомогти скласти загальну картину про тему тексту та зрозуміти важливі питання, що стосуються дослідження;
– аналіз – даний етап використовує різні методи, такі як кластеризація, категоризація та зведення даних для зрозуміння взаємозв'язків та структури даних, наприклад, можна використовувати кластеризацію для

групування текстів за схожими темами, або категоризацію для встановлення того, до якої категорії відноситься текст;

– візуалізація – на цьому етапі використовуються інструменти візуалізації для подання результатів аналізу, наприклад, можна використовувати графіки та діаграми для відображення структури даних та їх взаємозв'язків;
– інтерпретація та висновки – на останньому етапі проводиться інтерпретація результатів аналізу та формулюються висновки, наприклад, можна зробити висновок про те, які ключові слова та фрази використовуються в тексті, яка тема є основною, та які питання стосуються дослідження, ці висновки можуть бути корисні для прийняття рішень та розробки стратегій відповідно до результатів аналізу.

Підготовка текстових даних перед аналізом є дуже важливим етапом і може суттєво впливати на результати аналізу. Текстові дані, як і інші джерела інформації включають в себе велику кількість зайвого, непотрібного та навіть шкідливого контенту. Перед початком аналізу текстових даних необхідно провести підготовку, яка включає в себе очищення даних від зайвих символів, приведення до нижнього регістру, токенізацію, лематизацію, видалення стоп-слів та векторизацію. Якщо не здійснювати підготовку текстових даних, то результати аналізу можуть бути неточними та неправильними. Наприклад, якщо не видаляти стоп-слова, то аналіз може видавати високу важливість таких слів як «і», «або», «та», які не несуть суттєвої інформації для аналізу. Важливо зазначити, що дані можуть бути зібрані з різних ресурсів та мати купи помилок і неточностей. Тому перед тим, як проводити аналіз даних, необхідно виконати їх підготовку.

Загальний алгоритм попередньої обробки текстових даних може включати в себе наступні етапи:

– зчитування тексту з вихідного джерела, наприклад, файлу або веб-сторінки;
– видалення службових тегів, таких як HTML-теги, тощо;

- видалення зайвих символів, таких як розділові знаки, символи пунктуації, числа тощо;
- розділення тексту на окремі слова-токени;
- перетворення тексту до нижнього регістру – всі літери у нижньому регістрі;
- видалення стоп-слів, які не мають суттєвого значення для аналізу, наприклад, «я», «він», «та» тощо;
- лематизація - перетворення слів до їхніх базових форм;
- створення вектора слів, який можна використовувати для подальшого аналізу тексту.

Варто зазначити, що конкретний алгоритм попередньої обробки може залежати від конкретного завдання аналізу та властивостей текстових даних, однак концептуально буде складатися з описаних кроків. Загальний алгоритм попередньої обробки текстових даних продемонстровано на рисунку 2.

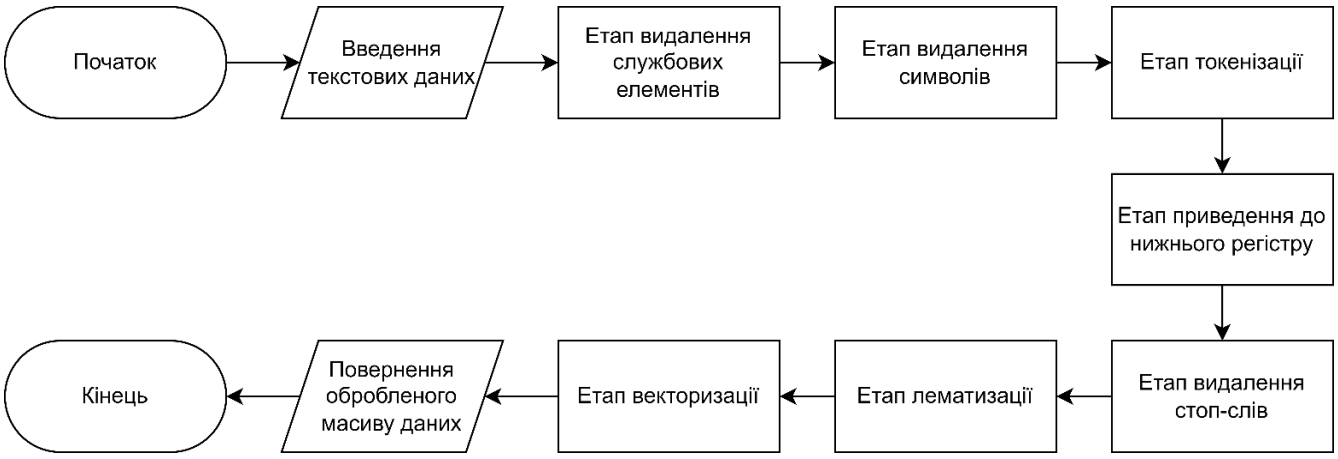

Рисунок 2 – Загальний алгоритм попередньої обробки текстових даних

Як видно з рисунка 2 загальний алгоритм являє собою послідовний лінійний процес, повне представлення якого наведено в додатку «Алгоритм попередньої обробки великих масивів текстової інформації». Кожен етап виконує свої функції, та є важливим кроком у підготовці текстових даних до подальшої обробки та аналізу, що допомагає зрозуміти контекст тексту, знизити шумову складову та забезпечити якісні результати аналізу.

Перш за все, для обробки текстових даних необхідно отримати самі дані. Зазвичай великі обсяги текстових даних можна отримати з попередньо підготовлених джерел, наприклад, використовуючи систему транскрибування голосових повідомлень та систему подальшої обробки та аналізу текстових даних. Також можна завантажувати дані з відкритих джерел. Аналіз відкритих даних може допомогти виявляти нові залежності та тенденції, розробляти нові стратегії та бізнес-моделі, підвищувати якість продукту або послуг, зменшувати витрати, підвищувати ефективність роботи та багато іншого. Відкриті дані можуть бути використані для створення аналітичних звітів, прогнозування попиту, виявлення аномалій, розробки систем рекомендацій, управління взаємодією з клієнтами, оцінки ризиків та багатьох інших завдань. Саме тому, відкриті джерела є перспективним інструментом для отримання текстових даних.

Отримання текстових даних з відкритих інтернет джерел може бути складним завданням, яке вимагає використання різноманітних інструментів та методів. Основні етапи процесу отримання текстових даних з відкритих джерел включають:

- визначення джерела – визначити джерело, з якого будуть отримані дані, це може бути веб-сторінка, база даних, API тощо;
- використання API, якщо доступний API, підключитись до нього та взяти необхідну інформацію;
- використання краулерів – програм, які сканують веб-сторінки та витягують з них інформацію, якщо джерело не має API, можна використовувати краулерів, щоб отримати дані;
- фільтрація даних – після отримання даних їх необхідно очистити від непотрібної інформації (наприклад, реклами, коментарів, посилань);
- збереження даних – після очищення даних їх необхідно зберегти у відповідному форматі, наприклад, CSV, JSON, SQL тощо;
- використання отриманих даних – після аналізу даних можна використовувати їх для різних цілей, наприклад, для розробки нових продуктів, виявлення тенденцій та залежностей, прогнозування попиту.

Видалення тегів та тобто службових елементів є важливою частиною попередньої обробки текстових даних, оскільки вони не мають значення для аналізу тексту і можуть перешкоджати правильному визначенню тематики, настроїв та інших важливих характеристик тексту. Цей етам дозволяє отримати чистий текст, що є основою подальшого аналізу. Крім того, видалення тегів може знизити обсяг даних, що підлягають аналізу, і покращити ефективність обробки. Важливо враховувати, що попереднє видалення тегів не завжди є необхідним, залежить від конкретної задачі та характеру даних. Наприклад, для аналізу структури сторінок веб-сайтів теги можуть бути корисними.

Найрозповсюдженішим джерелом текстових даних є не тільки HTML, а й XML, Markdown, LaTeX, PDF, RTF, DOCX, TXT, CSV та інші. Кожен з цих форматів може мати свою власну структуру та теги, які потрібно видаляти або обробляти перед аналізом даних. Тому важливо заздалегідь з'ясувати формат даних та проаналізувати його структуру, щоб ефективно виконати попередню обробку даних. Через це, перед початком обробки текстових даних, важливо провести аналіз вхідного файлу та визначити його тип, щоб мати змогу ефективно обробити текстові дані та виключити можливі помилки при обробці. На рисунку 3 продемонстровано загальний алгоритм етапу видалення службових елементів.

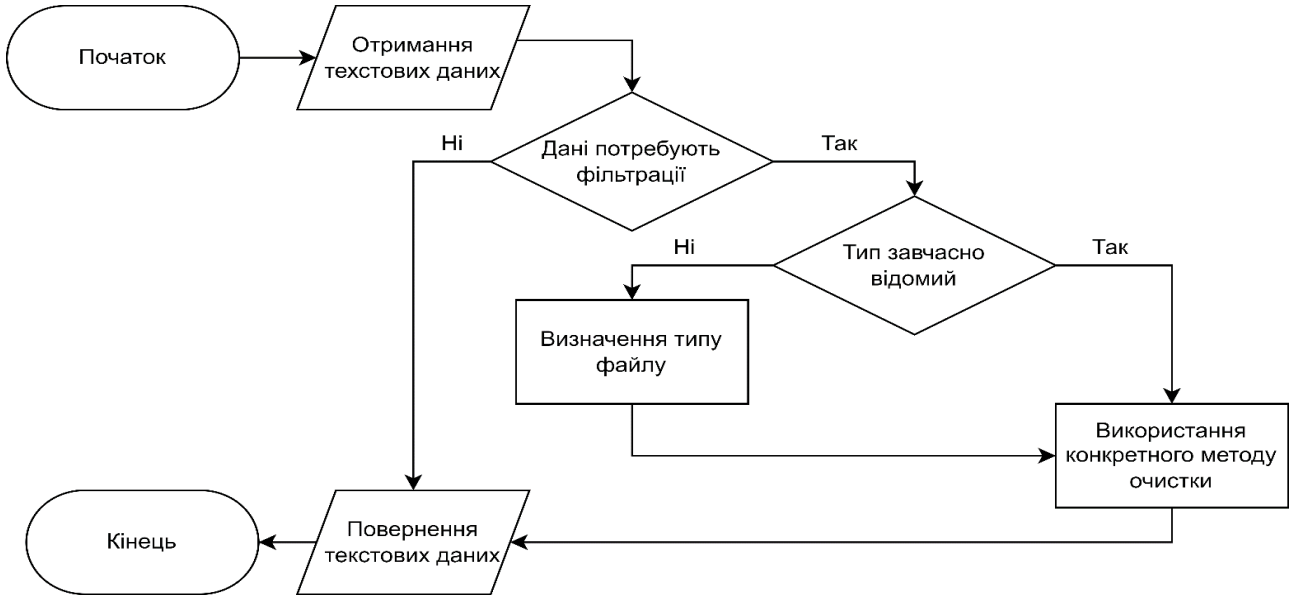

Рисунок 3 – Загальний алгоритм етапу видалення службових елементів

Етап видалення службових елементів концептуально може бути, як «загальним» по більшості можливих типах, так і «точковим», з попереднім визначенням типу файлу та подальшим пошуком конкретних символів. Хоча перший варіант простіший при розробці, однак другий – значно більш економічний та доцільний до використання. До того ж, при отриманні тексту з тандемних джерел, етап видалення службових елементів може загалом бути відсутнім.

Етап видалення символів – це частина попередньої обробки текстових даних, яка полягає в видаленні різних символів, які не несуть корисної інформації для подальшого аналізу тексту. До таких символів можуть належати знаки пунктуації, розділові знаки, числа, символи математичних операцій, спеціальні символи та інші. Вони можуть бути видалені або замінені на пробіли. Цей етап важливий для покращення якості векторизації та аналізу тексту, оскільки він дозволяє зменшити кількість слів, які можуть бути враховані при аналізі, а також уникнути спотворення результатів аналізу через наявність непотрібних символів.

Етап токенізації є одним з основних етапів попередньої обробки текстових даних. Токенізація є важливим кроком для обробки тексту, оскільки вона дозволяє використовувати окремі слова та символи як окремі одиниці, що можуть бути подальшим чином аналізовані. Цей етап полягає у розбитті тексту на окремі слова, вони же токени, за допомогою спеціальних правил, алгоритм якого продемонстровано на рисунку 4.

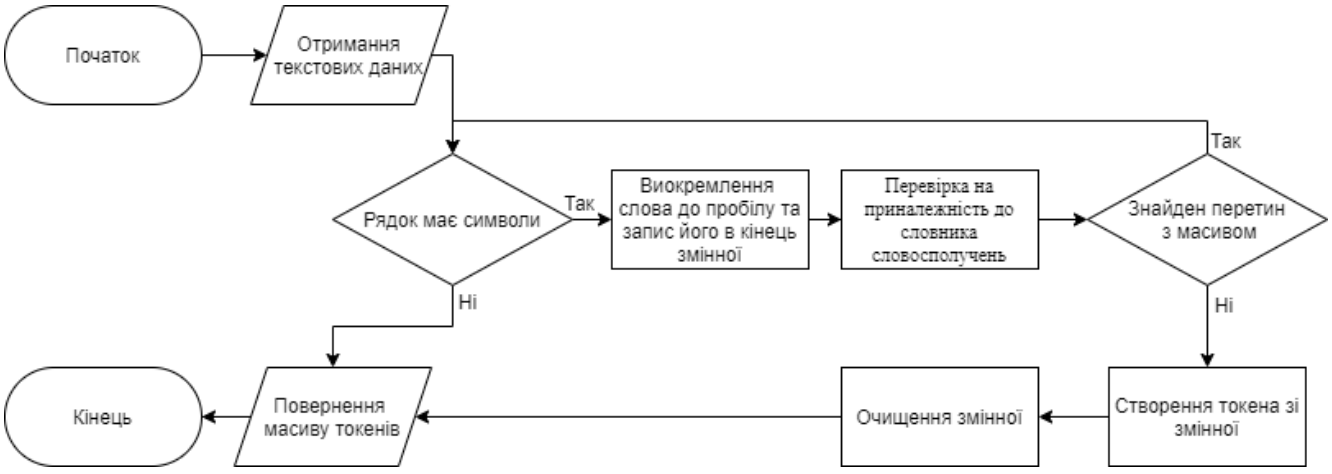

Рисунок 4 – Загальний алгоритм етапу токенізації

Найпростіші правила токенізації полягають у використанні пробілів як роздільників між словами. Однак, це не завжди є ефективним варіантом, оскільки деякі слова можуть містити пробіли у своїй структурі, наприклад, «Новий Орлеан». Тож текстові дані у вигляді рядка подаються на обробку. Через те що власні назви можуть включати в себе декілька слів, сенс яких кожного окремо буде не до кінця завершеним, доводиться перевіряти отримане слово на перетин із базою власних назв. У випадку позитивного перетину, до змінної додається нове слово та повторно проводиться перетин. Такий процес повторюється, допоки є збіги з масивом власних назв, після чого слово чи слова токенізуються та додаються до фінального масиву. Таким чином, речення перетворюються на списки слів, які можуть бути подальше використані для аналізу.

Етап приведення до нижнього регістру – це процес, який полягає у заміні всіх літер верхнього регістру на відповідні літери нижнього регістру у тексті. Це дозволяє уникнути проблем з узгодженням словникових форм слів. Його алгоритм характеризується наступними кроками:

- отримання масиву токенів;
- проходження по кожному слову та переведення його в нижній регістр;
- збереження результатів.

Цей етап є важливим кроком у попередній обробці текстових даних, оскільки він допомагає зменшити кількість унікальних слів у тексті та забезпечити єдність формату слів. Крім того, використання одного регістру дозволяє уникнути дублювання інформації та зменшити кількість помилок при пошуку слів у тексті. Наприклад, слова «Книга», «книгА», «книга» та «КНИГА» будуть перетворені на «книга», що дозволяє розглядати їх як одне слово при подальшому аналізі даних. Алгоритм може бути доволі простий, банальна перевірка кожного символу та подальша заміна верхнього регістру на нижній. Однак такий підхід не зважає на регістр абревіатур, що, при специфічних умовах, може призвести до втрати певного логічного навантаження. Відповідно, для таких потреб можна використовувати базу загальновідомих абревіатур з їх

розшифруваннями, для виключення конкретних токенів з приведення до нижнього регістру. Це дозволить більш широко зберегти значення тексту, та й така база даних стане в нагоді при тлумаченні значення тексту.

Етап видалення стоп-слів є важливим етапом попередньої обробки текстових даних для покращення якості аналізу. Стоп-слова – це слова, які не містять важливої інформації та не впливають на глобальний зміст тексту. Їх видалення допомагає уникнути включення найбільш часто зустрічаються слів за рахунок зменшення кількість слів для аналізу і сприяє підвищенню точності та ефективності аналізу. Список стоп-слів може включати часто вживані слова, такі як «і», «в», «на», «як», «так», «але» та інші. Для видалення стоп-слів можна використовувати готові списки, які зазвичай доступні в бібліотеках для обробки текстів в програмуванні, або можна створити власний список стоп-слів для конкретної задачі аналізу. Алгоритм видалення стоп-слів може бути наступним:

- визначити список стоп-слів, які потрібно видалити з тексту;
- отримати текст розділений на токени;
- перевірити кожне слово на приналежність до списку стоп-слів;
- якщо слово є стоп-словом, то видалити його з тексту;
- повернути оновлений текст без стоп-слів.

Видалення стоп-слів дозволяє скоротити довжину тексту і зменшити об'єм даних для подальшого аналізу, а також забезпечує більш точний інтерпретацію того, що справді важливо для розуміння тексту. Проте, варто зазначити, що видалення стоп-слів може призвести до втрати деякої інформації, тому потрібно обирати список стоп-слів з обережністю та ретельно аналізувати відфільтрований текст перед подальшим аналізом.

Етап лематизації в процесі попередньої обробки текстових даних включає в себе перетворення слів до базової форми, тобто леми. Лематизація забезпечує здатність розпізнавати спільні корені слова та ігнорувати закінчення, відмінювання та інші флексії, що дозволяє розуміти смислові зв'язки між різними формами одного слова [23]. Також це допомагає зменшити розмір словникового запасу і знизити кількість варіантів подання термінів у тексті, що поліпшує

результати подальшої обробки. Алгоритм лематизації зазвичай виконується за допомогою морфологічних правил, які базуються на мовних знаннях про суфікси, префікси, закінчення та інші характеристики слів, що відрізняють їх від інших слів. Його реалізація зазвичай включає наступні кроки:

– визначення частин мови слова – програма повинна знати, що слово є іменником, прикметником, дієсловом тощо;
– видалення закінчень та префіксів – програма повинна видалити закінчення та префікси слова, щоб знайти корінь слова;
– визначення базової форми – програма повинна знайти базову форму слова шляхом порівняння кореня зі словником лем;
– врахування виключень – деякі слова мають винятки від стандартного процесу лематизації, програма повинна мати можливість розпізнати ці слова та обробити їх окремо.

На рисунку 5 продемонстрована схема алгоритму лематизації.

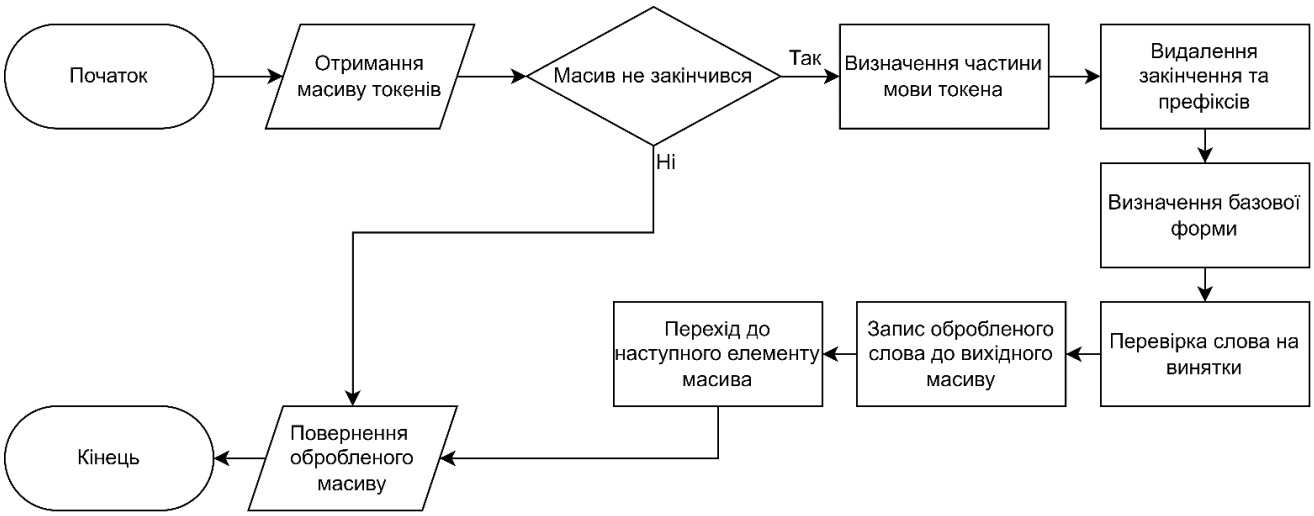

Рисунок 5 – Алгоритм лематизації масиву токенізованих текстових даних

Окрім того, існують стандартизовані словники, які містять базові форми слів, що дозволяють легко визначати їх леми. Наприклад, український стандартний морфологічний словник містить понад 100 тисяч слів і форм. Лематизація є важливим етапом попередньої обробки текстових даних, особливо

для мов, які мають складну морфологію. Вона дозволяє знизити розмір словникового запасу та забезпечити більш точні результати аналізу тексту.

Етап векторизації є важливим етапом в обробці тексту для подальшого застосування алгоритмів машинного навчання. Задачі машинного навчання, зазвичай, вимагають, щоб текст був представлений у векторному форматі, де кожен токен (слово, символ або фраза) має своє числове представлення. Етап векторизації – це процес перетворення текстових даних в числові вектори, які можна використовувати для подальшого аналізу та моделювання [24]. Цей етап включає в себе кілька кроків:

– створення словника – створення списку унікальних слів, які зустрічаються в текстових даних, це можна зробити за допомогою попередньої обробки тексту, такої як видалення стоп-слів та лематизація;
– побудова векторів – кожен текст перетворюється в числовий вектор, де кожна компонента вектора відповідає одному слову зі словника, ці значення можуть бути визначені за допомогою різних методів, таких як підрахунок частоти зустрічання слова (TF) та зважження частоти зустрічання слова (TF-IDF);
– нормалізація векторів – зазвичай вектори нормалізуються, щоб кожен вектор мав однакову довжину та був уніфікованим за шкалою, це можна зробити, наприклад, за допомогою відстані Евкліда;
– застосування моделей – отримані вектори можуть бути використані для подальшого аналізу та моделювання, таких як класифікація текстів, кластеризація текстів, пошук схожих документів тощо.

В цілому, векторизація є важливим етапом в аналізі текстових даних, оскільки дозволяє перетворити текст у векторний формат, який може бути використаний в подальших аналітичних процесах. Такі числові представлення тексту є необхідними для застосування алгоритмів машинного навчання, таких як класифікація, кластеризація, регресія тощо. Векторизація тексту також дозволяє виконувати аналіз схожості текстів та рекомендації на основі текстових даних.

Також на етапі векторизація виділяються ключові слова з тексту. Ключові слова (або терміни) – важливі слова, які підкреслюють основні теми та поняття, що містяться в тексті. Ключові слова допомагають читачеві швидко зрозуміти основну тему та зміст тексту, забезпечують точку та конкретну інформацію про зміст тексту, а також допомагають в аналізі та класифікації текстової інформації. Метод TF-IDF використовується для того, об виділити ключові слова та фрази з тексту [25]. У результаті використання цього методу ми отримаємо значення TF-IDF для кожного слова в тексту, таким чином можна виділити ключові слова, оскільки чим вище значення TF-IDF для слова в тексті, тим важливішим це слово є для цього тексту, тому можна вважати, що слова з найвищими значеннями TF-IDF є ключовими словами цього тексту.

Для визначення тематики тексту було обрано класифікувати текст. Класифікація є одним з основних методів аналізу даних та машинного навчання, що дозволяє автоматично призначати кожному об'єкту вхідних даних певний клас або категорію на основі його характеристик. Цей процес полягає в тому, щоб навчити комп'ютер класифікувати нові об'єкти, використовуючи знання, які він здобув на основі попередніх даних [26]. Класифікація текстових даних зазвичай здійснюється з використанням методів машинного навчання, зокрема, навчання з учителем. Це означає, що алгоритм отримує вхідні дані у вигляді попередньо класифікованого набору текстів, які вже мають відповідні мітки. Алгоритм використовує ці дані для створення моделі, яка може прогнозувати клас тексту на основі його властивостей. Таким чином, істотно важливо для системи наявність підготованих тренувальних даних. Структура даних наведена на рисунку 6.

Рисунок 6 – ER-модель класифікації текстових даних

На моделі сутність-зв’язок представлено як за типами класифікуються текстові дані. Зв'язок «багато до багатьох» використовується в даному випадку для того, щоб забезпечити гнучкість та можливість зв'язувати багато текстів з багатьма типами. Цей тип зв'язку дозволяє зручно та ефективно моделювати такі сценарії, де кожний текст може класифікуватися будь-якою кількістю типів, а також один тип може класифікувати багато різних текстів. Такий зв'язок допомагає зберігати дані про тексти та типи у структурованому та організованому форматі, що спрощує подальшу обробку та аналіз даних. Основна доцільність використання такого підходу полягає у тому, що, зазвичай, кожен текст, твір, стаття тощо, буде мати безліч можливих характерних рис, які дозволять більш гнучко та точно розуміти та класифікувати подані текстові дані.

Для класифікації та аналізу великих масивів текстової інформації було вирішено використовувати наївний Баєсовий класифікатор. Аналіз тексту на основі наївного Баєсового класифікатора включає в себе наступні кроки:

- визначення категорій – перш за все, необхідно визначити категорії текстів, на які можна розділити дані, наприклад, у випадку класифікації новин можна використовувати категорії «спорт», «політика», «економіка» тощо;
- побудова моделі – наступним кроком є побудова моделі на основі навчальних даних, вона включає у себе статистичні дані про те, які слова або фрази використовуються в текстах кожної категорії;
- тестування – після побудови моделі необхідно протестувати її на нових даних; тексти, які не входили в навчальний набір, можна класифікувати за допомогою моделі і порівняти результати з відомими категоріями;
- оцінка результатів – після тестування можна оцінити результати аналізу тексту, найчастіше використовується метрика точності, що відображає відсоток правильно класифікованих текстів від загальної кількості.

Наївний Баєсовий класифікатор використовує теорему Баєса для класифікації текстових документів. Загалом, теорема Баєса використовується для обчислення умовної ймовірності, тобто ймовірності події А при умові, що відома

подія B сталася. У випадку наївного Баєсового класифікатора, ми використовуємо теорему Баєса для обчислення ймовірності того, що текстовий документ належить до певного класу, з урахуванням входження певних слів або фраз в цей документ. Зазвичай ці слова або фрази називаються «ознаками» (features). Для того, щоб застосувати наївний Баєсовий класифікатор до текстового документа, спочатку необхідно підготувати набір тренувальних даних, який складається з текстових документів, для яких ми знаємо, до якого класу вони належать [27]. Для кожного класу ми будуємо «модель», яка представляє собою набір умовних ймовірностей, що описують відношення між входженням певних слів або фраз та належністю документу до певного класу. Класифікація наївного Баєсового класифікатора здійснюється за формулою Баєса:

$$P(C|D) = \frac{P(D|C) * P(C)}{P(D)}, \quad (1)$$

де C – клас, до якого належить вхідний документ;

D – вхідний документ,;

$P(C|D)$ – ймовірність того, що вхідний документ належить класу C при умові, що відомі його документи D;

$P(D|C)$ – ймовірність входження документу D у клас C;

$P(C)$ – апріорна ймовірність класу C;

$P(X)$ – ймовірність входження ознаки X в будь-який клас.

Наївний Баєсовий класифікатор припускає, що умовні ймовірності входження слів або фраз в документи незалежно одна від одної. Це означає, що якщо ми маємо документ, що складається з N слів (або токенів), то вірогідність того, що цей документ належить до певної категорії, може бути вирахована наступним чином:

$$P(category|document) = P(category) * P(word_1|category) * P(word_2|category) * \ldots * P(word_N|category), \quad (2)$$

де $P(category|document)$ – умовна ймовірність того, що документ належить до певної категорії, даний документ;

$P(category)$ – апріорна ймовірність категорії;

$P(word_i|category)$ – ймовірність того, що слово або токен $word_i$ входить до категорії.

Після вирахування ймовірності для кожної категорії, можна порівняти їх та вибрати ту, яка має найбільшу ймовірність. Ця класифікація є потужним методом для аналізу великих масивів текстової інформації з використанням ключових слів та фраз. Вона може бути використана для класифікації текстів за тематикою, розпізнавання образів, визначення тональності текстів або для виявлення спаму. Наївний Баєсовий класифікатор приймає рішення про класифікацію на основі максимальної умовної ймовірності: він призначає вхідний зразок до класу, для якого умовна ймовірність $P(C|X)$ максимальна. Тобто, алгоритм визначає ймовірність належності нового зразка до кожного класу, на основі розрахованих умовних ймовірностей, і призначає зразок до класу з найвищою ймовірністю. Зазвичай, наївний Баєсовий класифікатор добре працює на великих наборах даних з багатьма ознаками. Він швидко навчається та швидко працює, що робить його привабливим варіантом для задач з класифікацією тексту. Результат використання наївного Баєсового класифікатора – це набір прогнозів для кожного з елементів вхідного набору даних, де кожен прогноз відноситься до певного класу. Це дозволяє визначити тематику тексту та його класифікацію [28].

На останньому етапі аналізу великих масивів текстової інформації проводиться трактування результатів та формулюються висновки. Наприклад, можна зробити висновки про те, які ключові слова та фрази використовуються в тексті, яка тема є основною, та які питання стосуються дослідження. Ці висновки можуть бути корисні для прийняття рішень та розробки стратегій відповідно до результатів аналізу.

2.3 Реалізація та тестування системи

Аналіз текстової інформації зазвичай виконується на великих обсягах даних, що вимагає певної ефективної інфраструктури та інструментів для обробки цих даних. У цьому розділі розглянуто реалізацію та складові підсистеми аналізу

великих масивів текстової інформації з використанням мов програмування Python та бібліотек NLTK, WordCloud, Matplotlib, а також HTML, CSS та JavaScript з бібліотекою Vue.js. Також будуть описані процес тестування розробленої системи та її результати.

Структура розробленої системи, що зображена на рисунку 7, складається з трьох компонентів:

– компонент «Візуальний інтерфейс»;
– компонент «Текстовий процесор»;
– компонент «База даних».

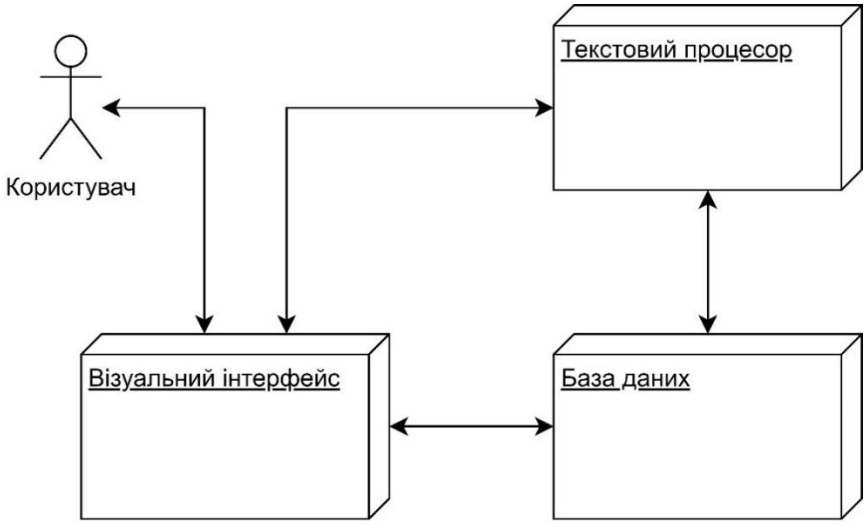

Рисунок 7 – Структурна схема розробленої системи

Діаграма, що представлена у додатку А на плакаті 2 «Діаграма компонентів» показує, як компоненти програмного забезпечення пов'язані між собою, та використовується для візуалізації структури програмного забезпечення та його компонентів. Вона називається діаграмою компонентів та дозволяє розуміти, які компоненти входять до складу системи, як вони пов'язані між собою та як вони взаємодіють з зовнішніми системами та користувачами. Підсистеми розробленої системи мають між собою певні зв’язки, а також вони мають зв’язки з зовнішніми підсистемами, а саме диском та Azure authentication, що використовуються для аутентифікації, реєстрації та збереження даних. Кожна з

трьох підсистеми має свої компоненти, які виконують певні функції. Підсистема «Візуальний інтерфейс» включає в себе наступні компоненти:

– аналіз тексту;
– передача файлу;
– аутентифікатор.

Підсистема «Текстовий процесор» складається з компонентів:

– наївний Баєсовий класифікатор;
– приймач даних;
– текстовий препроцесор.

В свою чергу остання підсистема представлена наступними компонентами:

– приймач даних;
– генератор Id;
– створювач запису;
– отримувач даних.

Для реалізації візуальної складової, що служить для представлення інформації на веб-сторінці та забезпечення користувачеві інтерактивності у цій системі використано наступні технології:

– HTML;
– CSS;
– JavaScript з бібліотекою Vue.js;
– Python та її вбудовані бібліотеки Matplotlib та WordCloud.

Також Python використано для створення back-end-у, тобто частини сайту, з якою взаємодіє сервер, що забезпечує обробку запитів користувачів, взаємодію з базою даних, збереження інформації на сервер тощо. До того ж мова Python була використана для реалізації попередньої обробки та проведення аналізу текстових даних. Для цього процесу було обрано бібліотеку NLTK, що є бібліотекою для обробки природної мови у програмній мові Python. Вона містить багато інструментів для роботи з текстом, таких як токенізація, лематизація, стемінг та інше.

Основною сторінкою розробленої системи є сторінка «Аналіз тексту», котру представлено на рисунку 8. Ця сторінка надає можливість виконати аналіз тексту, передивитись помилки, а також виправити їх за потреби, вона є основною у розробленій підсистемі. Аналіз тексту можна виконати коректно українською, англійською та німецькою мовами. Інтерфейс доступний на трьох мовах, які доступні також для аналізу. На сайті доступні ці мови через те, що тренування наївного Баєсового класифікатора проводилось цими мовами задля досягнення кращого результату. Тренування здійснювалось шляхом використання веб-сайту Вікіпедія, яка має доступний API, котрий було підключено до системи та взято необхідну інформацію. Такий підхід дозволив перевірити правильність аналізу та класифікації, так як статті з цього ресурсу є структурованими та не несуть зайвої інформації. До того ж вони представлені на різних мовах, що дозволило натренувати наївний Баєсовий класифікатор належним чином для них.

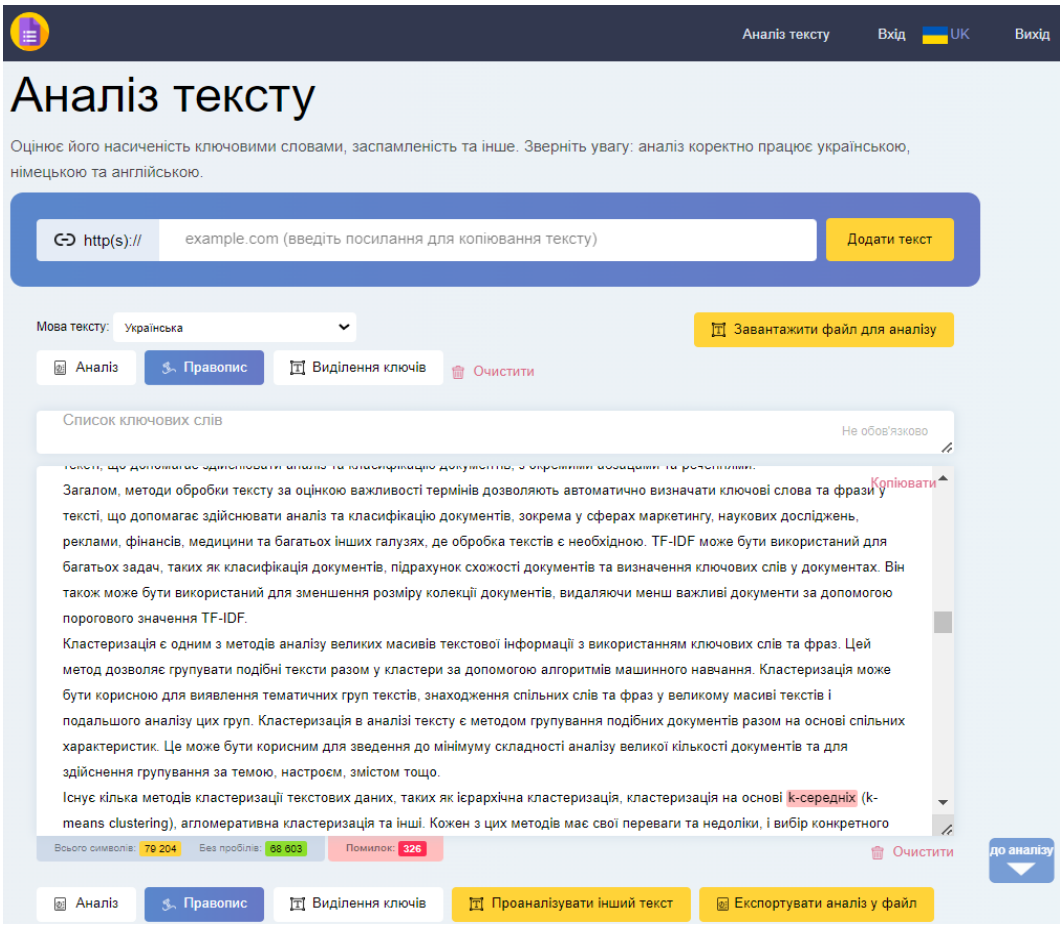

Рисунок 8 – Сторінка «Аналізу тексту» розробленої підсистеми

На цій сторінці користувач має можливість ввести текст для аналізу мануально, також він може проаналізувати вміст певної сторінки, увівши посилання на неї у відповідне поле, завантажити файл для аналізу, файли можуть бути із розширенням .doc, .docx, .pdf, .txt, .html, .json, .cvs та інші, також користувач може обрати мову тексту мануально, або вона буде обрано автоматично розробленою підсистемою після отримання тексту, перевірити введений текст на правопис, мануально ввести ключові слова та виділити їх, якщо в цьому є потреба, та провести аналіз наданого тексту. Для того, щоб мати доступ до цієї сторінки, користувач повинен бути авторизований у систему, це можна зробити на сторінці, що представляє функціонал для входу у систему, натиснувши на кнопку «Вхід» на панелі навігації. Сторінку «Вхід» представлено на рисунку 9:

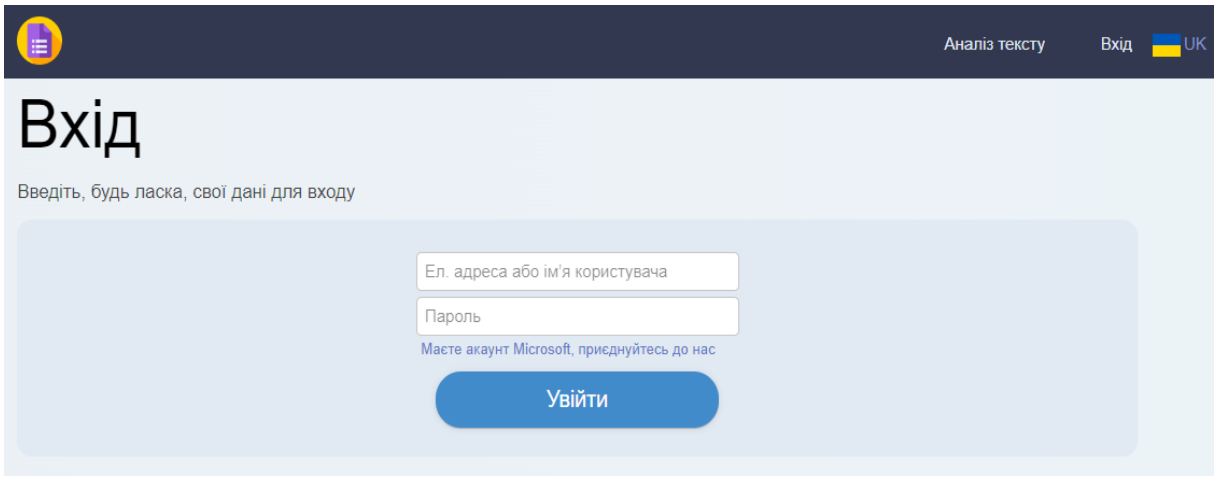

Рисунок 9 – Сторінка «Вхід»

Для авторизації користувач повинен мати акаунт Microsoft, через який він отримує доступ до використання можливостей підсистеми. Якщо юзер ще не входив у розроблену підсистему, він має можливість увійти у свій аккаунт Microsoft та таким чином отримати доступ до сайту. Якщо ж користувач ще немає цього акаунту, він може його створити. Ці дії користувач може виконати, натиснувши на кнопку «Маєте акаунт Microsoft, приєднуйтесь до нас», яка

відкриє стандартну сторінку для входу в акаунт компанії Microsoft, яке зображено на рисунку 10:

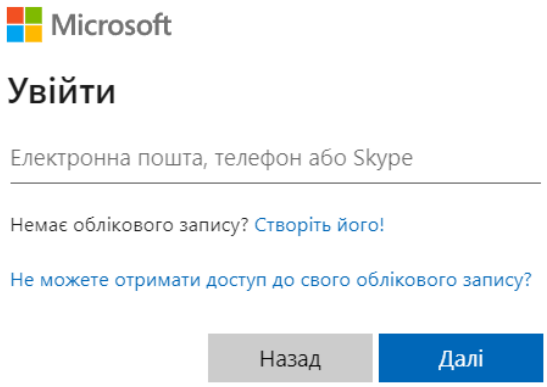

Рисунок 10 – Вхід у Microsoft акаунт

Якщо користувач не авторизувався у системі, навіть, натиснувши «Аналіз тексту» на панелі навігації, він побачить текст на цій сторінці з проханням увійти у систему, натиснувши на який, юзер зможе перейти на сторінку «Вхід» та виконати відповідні дії. Цю сторінку представлено на рисунку 11. Таким чином, юзер, що не є авторизованим, не може переглянути та скористуватись функціоналом розробленої підсистеми.

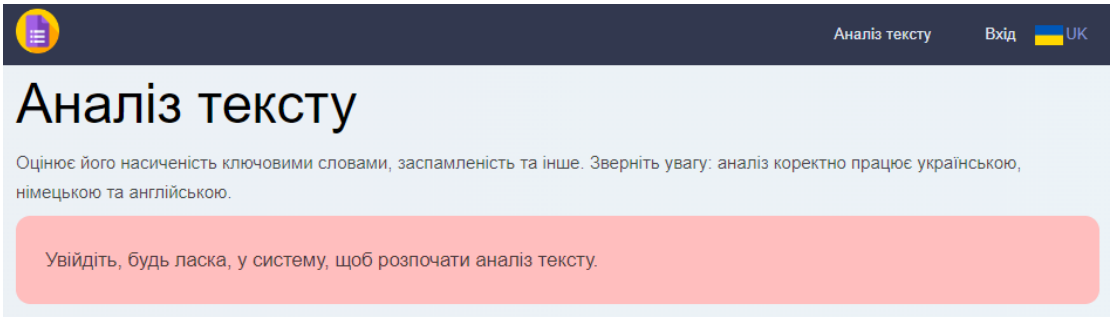

Рисунок 11 – Сторінка «Аналіз тексту», що не дозволяє скористатись функціоналом системи

Ще однією частиною сторінки «Аналіз тексту» є «Результати аналізу». Цей функціонал представлено на рисунку 12. На даній частині сторінки користувач

може побачити результати аналізу представленого ним тексту. На сторінці представлені загальні параметри тексту та три вкладки:

– «Ключові слова»;
– «Словник»;
– «Аналіз».

Перша вкладка представляє собою статистику за словами з тексту, за винятком стоп-слів. На ній користувач може побачити всі слова тексту з кількістю вживань кожного з них, їх релевантністю, тобто того, наскільки кожне слово відображає зміст, відсоток значимих слів та відсоток серед всіх слів в тексті, а також ключові слова цього тексту, вони відображуються в окремому віконечку над таблицею зі статистикою. Користувач має можливість змінення кількості записів, що відображуються у таблиці, та може переключатись між вкладками статистики. Як видно з рисунку нижче, цей текст містить десять ключових слів та фраз, близько 1670 слів у статистиці, його тематикою є класифікація текстів, аналіз та дослідження та інші його характеристики.

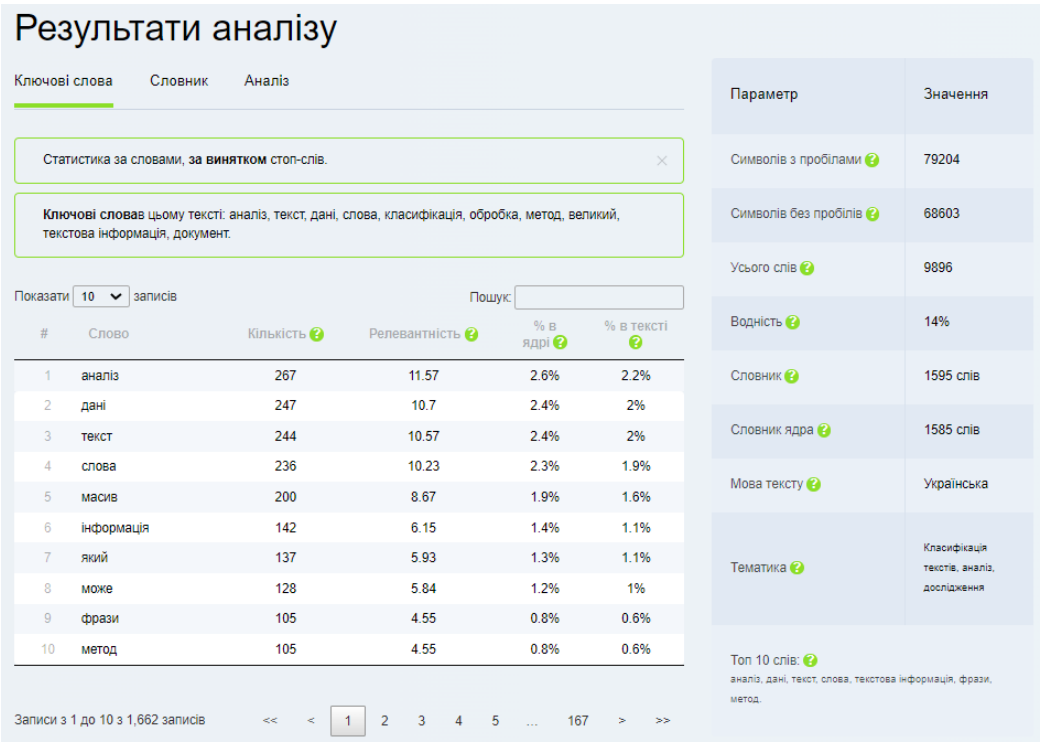

Рисунок 12 – «Результати аналізу» вкладка «Ключові слова»

Друга вкладка цієї частини зображена на рисунку 13 та містить список слів, що включає в себе текст, за винятком стоп-слів. Його відсортовано за частотою вживання слів та подано у зручному для копіювання вигляді, тобто списком.

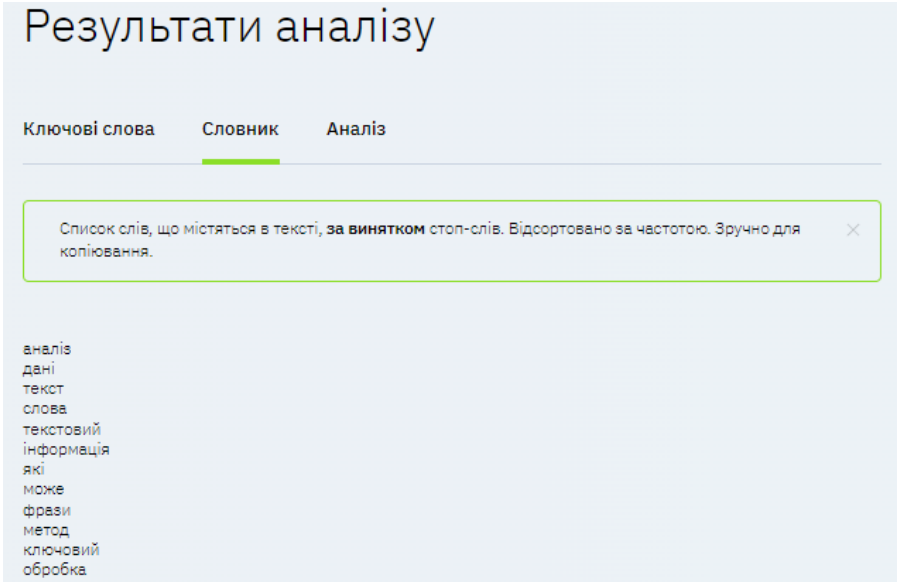

Рисунок 13 – «Результати аналізу» вкладка «Словник»

Остання вкладка містить додаткові результати аналізу, а саме:

– хмару слів;
– гістограму частотності слів;
– теплова карта вживання слів;
– кругова діаграма класифікації.

Ці елементи візуалізації аналізу тексту можуть допомогти користувачеві легше зрозуміти результати аналізу тексту та надати йому візуальне представлення даних. Вони можуть допомогти швидко визначити ключові теми та відношення між різними аспектами тексту. Наприклад, графіки частотного аналізу слів можуть показати які слова найчастіше використовуються в тексті, а кругові діаграми можуть відобразити розподіл категорій або тем тексту. Це також може допомогти користувачам легше зрозуміти залежності та відношення між різними елементами тексту. Наприклад, відображення сили зв'язків між словами або темами може допомогти користувачеві зрозуміти, які ідеї та концепції пов'язані з однією або декількома темами. Крім того, графіки та діаграми можуть

допомогти користувачеві більш ефективно інтерпретувати та використовувати результати аналізу тексту. Наприклад, зображення залежності між категоріями та кількістю слів в кожній з них може допомогти визначити теми, які потребують більш детального аналізу, та теми, які можуть бути відкинуті. Візуалізація аналізу представлена на рисунку 14.

Рисунок 14 – «Результати аналізу» вкладка «Аналіз»

Процес тестування системи показав, що розроблена система зчитує текст з наданих файлів та ресурсів, дозволяє користувачеві виправити помилки в представленому тексті за необхідності, визначити ключові слова, коректно аналізує великі масиви текстової інформації, надає статистичні дані аналізованого тексту, відображує корисні діаграми та графіки для подальшого використання та дозволяє зберегти результати аналізу в окремий документ. Вхід здійснюється за допомогою акаунтів Microsoft. Система працює належним чином та може бути допрацьована за подальшої необхідності.

Більш детально усі функції системи для користувача та сценарії використання показані у додатку А на плакаті 3 «Діаграма прецедентів». Ця

діаграма описує функціональність системи з точки зору її користувачів та інших зовнішніх агентів [29]. Вона дозволяє ідентифікувати функції, які повинна виконувати система, та відображає зв'язки між користувачами та функціями системи. Діаграма прецедентів допомагає зрозуміти, як система буде використовуватися та які функції потрібні для досягнення мети. Актором в цій системі є звичайний користувач, котрому потрібно авторизуватись, щоб отримати доступ до функціоналу системи. Користувачем цієї системи може стати будь хто та використовувати її у своїх власних цілях. Після авторизації користувач має декілька сценаріїв використання системи:

- змінити мову візуального інтерфейсу системи;
- завантажити текстову інформацію для аналізу зі сторонніх інтернет-джерел;
- ввести текстову інформацію мануально;
- вставити текстові дані для аналізу;
- завантажити документ для аналізу;
- вийти з системи.

Наступним кроком є початок аналізу доданого тексту. Ця дія пропонує різні сценарії використання, а саме:

- переглянути правильність правопису в тексті;
- виділити ключі в тексті;
- очистити дані зі сторінки;
- переглянути статистику за словами тексту;
- переглянути ключові слова та фази тексту;
- переглянути тематику тексту;
- переглянути відомості про текст;
- переглянути словник слів тексту;
- скопіювати словник слів тексту;
- переглянути візуалізований аналіз тексту;
- зберегти аналіз тексту у файл;

– вийти з системи.

Цей набір сценаріїв використання системи забезпечує користувача повним доступом до системи та аналізу текстової інформації задля забезпечення потреб юзера у розробленій системі.

На плакаті 4 «Діаграма станів», що знаходиться у додатку А, зображено загальний процес користування розробленою підсистеми юзером з варіантами його дій у цій системі. Взагалі, ця діаграма використовується для візуалізації процесів, що залежать від станів. Вона дозволяє описати різні стани системи та переходи між ними залежно від певних подій або умов. Наприклад, діаграма з цього плакату містить такі умови, як «Дію відхиллено», «Підтвердження», «Дію підтверджено», «Дані невірні», «Змінити мову», «Очистити дані» та інші. Кожна з цих умов провокує певну подію.

При вході в систему, на початку роботи користувача з системою, вона приймає стан відображення сторінки авторизації на візуальному інтерфейсі клієнта. Без авторизації у системі, користувач не зможе переглянути контент, що відповідає основній сторінці, та відповідно, не зможе скористатись основними функціями системи. Сторінка авторизації, як і основна сторінка, надають користувачеві можливість зміни мови візуального інтерфейсу. Якщо користувач прийняв рішення змінити мову візуального інтерфейсу, система переходить до стану змінення мови, після чого відображає сторінку на потрібній юзеру мові. Після вході у систему користувач може побачити та скористатись усім функціоналом основної сторінки, де при натисканні на кнопку «Завантажити файл для аналізу» система перейде у стан відображення вікна вибору файлу з комп’ютера користувача, також юзер може ввести текст мануально або скопіювати його в основне поле та натисне на кнопку «Аналіз», а при введені посилання на інтернет-ресурс та натисканні на кнопку «Додати текст», система почне процес вивантаження тексту з наданого інтернет-ресурсу та у обох випадках далі перейде до стану аналізу текстової інформації. По завершенню аналізу система перейде у стан відображення аналізованих даних, статистики та візуалізованих даних. Після цього користувач може експортувати аналіз у файл та

завантажити його собі на пристрій, у цьому випадку система перейде спочатку у стан формування файла з аналізом, а потім у стан відображення вікна для вибору місця збереження файлу.

Задля демонстрації послідовної взаємодії між різними об’єктами в рамках UML використовується діаграма послідовності [30]. Цю діаграму подано у додатку А на плакаті 5 «Діаграма послідовності підсистеми». Вона слугує для зображення та опису процесів у системі та демонструє процес внесення текстової інформації у системи та процес аналізу, що починається в момент, коли користувач додав текст для аналізу, та закінчується отриманням результатів аналізу. На ній також зображено порядку обміну повідомленнями та даними між складовими системи. Процес, який зображує діаграма, є основним процесом розробленої системи. Після додавання тексту для аналізу користувачем у веб-застосунок, він формує POST-запит та відправляє його у базу даних, яка в свою чергу створює новий запис та повертає ID створеного запису у веб-застосунок. Після цього створюється новий POST-запит, котрий містить у собі ID тільки що створеного запису у базі даних та текст, що надав користувач, та надсилається текстовому процесору, де й відбувається основний процес аналізу. Також текстовий процесор виконує попередню обробку тексту для досягнення більшої точності при аналізі, після чого підготовлений текст аналізується. Далі знову створюється POST-запит на оновлення запису у базі даних. Цей POST-запит містить у собі результати аналізу та ID тексту. Після отримання надісланого базі даних запиту, вона оновлює записи відповідно до наданого ID та надсилає зворотне підтвердження оновлення записів текстовому процесору. Передостаннім етапом є формування POST-запиту для веб-застосунку, який містить у собі результати аналізу, доданого користувачем, тексту. Цей запит оброблює веб-застосунок та в результаті показує користувачеві візуалізовані результати аналізу тексту.

Розроблена система потребує постійної роботи з даними. Для роботи з даними використовуються бази даних. Для коректного виконання цієї задачі база даних повинна мати коректну структуру та відповідати правилам цілісності та

бути нормалізованою. Даталогічна модель є моделлю логічного рівня і являє собою відображення логічних зв'язків між елементами даних безвідносно до їхнього змісту й середовищу зберігання [31]. Для демонстрації структури було створено даталогічну модель, яка представлена у додатку А на плакаті 6 «Даталогічна модель».

Основними елементами цієї діаграми є сутності, мж якими присутня взаємодія. Основними сутностям розробленої системи є:

– тексти;
– файли;
– веб-ресурси;
– ключові слова;
– класифікації тексту;
– класифікації;
– мови;
– стоп-слова;
– частини мови;
– виняткові словосполучення.

Сутність «тексти» містить у собі усі тексти, що різними способами були завантажені у систему. В залежності від того, як саме це було зроблено, у вигляді файла чи з інтернет-ресурсу, заповнюються відповідні таблиці «файли» та «веб-ресурси». Кожна з цих таблиць містить посилання на Id запису у таблиці «тексти», де і зберігається текст наданого ресурсу. Вони також містять загалі характеристики завантажених даних. На таблицю «мови» посилаються такі таблиці, як «тексти», «виняткові словосполучення» та «стоп-слова». Таблиця «стоп-слова» зберігає всі стоп-слова для кожної мови, які використовуються при попередній обробці тексту на етапі видалення стоп-слів. Також в попередні обробці є етап токенізації, в якому приймають учать словосполучення з таблиці «виняткові словосполучення». Після аналізу тексту часткові результати аналізу записуються до таблиць «ключові слова» та «класифікації тексту». Інша частина аналізу, а саме візуальне представлення, формується на основі збережених даних.

При необхідності, ця структура може бути розширена, але лише за умови дотримання правил цілісності та нормалізації.

Для структурування та опису бізнес-процесів у системі можна використовувати графічне представлення у вигляді діаграм. Одним із видів діаграм для відображення потоків процесів є flowchart. Діаграма, яка описує роботу акторів системи, а саме, авторизованого юзера та юзера без авторизації, в рамках одного сеансу розробленої системи зображена у додатку А на плакаті 7 «Діаграма потоків». Як видно з цієї діаграми, користувач, що не був авторизований, немає доступу до всіх функцій системи. Він може переглянути основну сторінку, змінити мову, якщо інформація незрозуміла йому, але якщо він хоче отримати доступ до функціоналу системи, йому треба авторизуватись за допомого акаунта Microsoft, якщо він не має цього акаунта, але має бажання отримати доступ до функціоналу системи, користувач може створити акаунт Microsoft, якщо ж ні, він може покинути сторінку. При цьому авторизований юзер має повний доступ до системи. Він також може змінити мову інтерфейсу за необхідності. Також йому надається можливість проаналізувати текст або перевірка правопису тексту, який він надає через введення посилання на веб-ресурс, або завантажує у вигляді файлу, або вводить мануально. Після цього користувач отримує інформацію, якою цікавився та виходить з акаунту, завершуючи сеанс.

Як видно з результатів роботи розробленої системи, вона обробляє та аналізує текст, виводить статистичні дані відносно заданого тексту та наївний Баєсовий класифікатор визначає тематику та класифікує текст належним чином. Для того, щоб він працював у межах норми було проведено його навчання, для аналізу та класифікації йому були надані різні статті зі сторонніх веб-ресурсів. Без навчання класифікатор зазвичай видавав випадкові класифікації, що не були ніяк не пов’язані з текстом, що був наданий для аналізу. Якщо англійською та німецькою класифікації іноді підходили, то українською це ставалось набагато рідше. Після навчання наївного Баєсового класифікатора та тренувань, він почав працювати належним чином як для англійської та німецької мов, так і для

української мови, що і було ціллю навчання. Як видно з рисунків 12-14, наразі класифікатор та TF-IDF коректно виконують аналіз попередньо обробленого тексту у розробленій системі.

Для того, щоб система видавала ще більш точні результати, можна повести різні серії тренувань наївного Баєсового класифікатора, що дозволить покращити систему та зменшити час обробки даних. Окрім вже запропонованого варіанту її можна покращити наступним чином:

- використати вузькоспеціалізованих слів у аналізі, бо вони можуть сильно підкреслювати, наприклад, тему тексту,
- використати не тільки популярних слів, а й більш рідких, тобто не тільки першу чверть відсотків слів масиву, а й частину інших;
- використати довгі слова та абревіатур в аналізі;
- враховувати загального обсягу тексту, бо зі збільшенням об’єму вага кожного слова стає меншою;
- розширення функціоналу системи, наприклад, шляхом транскрибування голосових повідомлень чи аудіофайлів з подальшим їх аналізом;
- додати аналіз настроїв та емоції, що виражені в тексті, це може бути застосоване, наприклад, для відстеження реакцій користувачів на новини або продукти, а також для виявлення настроїв споживачів у соціальних мережах;
- відобразити історію аналізу для кожного користувача.

Загалом, ця система може бути використана для багатьох цілей та для різноманітних завдань, пов'язаних з обробкою та аналізом текстових даних. Основні варіанти застосування систем аналізу текстової інформації включають:

- моніторинг та аналіз соціальних мереж та інтернет-форумів – за допомогою системи можна відстежувати публікації про певну компанію, бренд, товар або послугу, виявляти негативні відгуки та швидко реагувати на них;

- аналіз ринку та конкурентів – система може допомогти відстежувати та аналізувати публікації про конкурентів, збирати та обробляти дані про їхні продукти та послуги, виявляти тенденції на ринку;
- пошук та аналіз новин – система може бути використана для збору та аналізу новинних статей за ключовими словами та фразами, що дозволить швидко відстежувати новини з певної галузі або на певну тему;
- аналіз власного контенту – система може допомогти відстежувати ефективність власного контенту, аналізувати відгуки та коментарі на сайті чи соціальних мережах, виявляти потенційні проблеми та негативні відгуки та реагувати на них;
- аналіз користувачів – розроблена система може бути використана для аналізу поведінки користувачів на сайті чи в додатку, збору та аналізу даних про їхній інтереси та потреби, виявленні потенційних клієнтів та інше;
- аналіз розпізнаного тексту з голосових повідомлень – система може допомогти ідентифікувати ключові теми, визначити найбільш важливі питання, а також виявити тенденції та зміни в поведінці співрозмовника.

Таким чином, розроблена система може мати різні сфери застосувань, вона може зазнати подальшого розвитку та удосконалення шляхом розширення її функціоналу та подальших тренувань, що позитивно відобразиться на результатах, які надає система.

Висновки до розділу 2

На основі виділених проблем безпеки, обрано Azure Blob Storage, де завантажені текстові файлі будуть надійно захищені. Після аналізу існуючих технологій, методів і алгоритмів аналізу та оброблення текстової інформації з метою отримання головної інформації для реалізації підсистеми аналізу вирішено використовувати алгоритм попередньої обробки текстової інформації, бо

застосування попередньої обробки даних допомагає покращити якість моделей машинного навчання, знижує розмірність простору векторів, що зменшує вимоги до обчислювальних ресурсів, та забезпечує більш точне інтерпретування результатів, також використовувався наївний Баєсовий класифікатор для класифікації та виділення тематики тексту та TF-IDF для визначення ключових слів у тексті, а також їх вагомості.

Якщо говорити про попередню обробку масиву текстових даних, то вона є важливою складовою будь-якого проекту, пов'язаного з аналізом тексту. Доцільність використання попередньої обробки текстових даних перед їх подальшим аналізом залежить від конкретної задачі. Однак, взагалі можна сказати, що вона допомагає покращити якість та ефективність обробки тексту. На останньому етапі було вирішено виділити ключові слова за допомогою методу TF-IDF.

В результаті розроблено та відтестовано систему аналізу великих масивів текстової інформації, котра складається з трьох частин, та була побудована на основі клієнт-серверній архітектурі для зручності взаємодії з системою у віддаленому форматі. Програмна реалізація потребує використання БД для збереження даних. Для забезпечення доступу до системи необхідна наявність акаунту Microsoft, який за необхідності користувач може створити.

В результаті розробки структури системи у форматі відповідних схем, діаграм та моделей, описано структуру системи, функціонал, що доступний, користувачеві, порядок його дій при взаємодії з системою, а також послідовної взаємодії між різними об’єктами.

Для зручності роботи з системою та позитивного користувацького досвіду, було важливо створити простий та зрозумілий візуальний інтерфейс системи, який дозволить користувачам на легко взаємодіяти з нею. Також запропоновані варіанти покращення системи, напрямки подальших досліджень у цій області, а також варіанти можливих її застосувань у прикладних задачах. Розроблена система може набути розвитку та вдосконалень, але й наразі розроблений функціонал може бути використаний для задоволення потреб користувача.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

В сучасному світі текстовий аналіз стає все більш популярним і важливим завдяки своїм потужним можливостям у вирішенні різноманітних задач. Попередньо оброблений та векторизований текст є ключовим елементом в аналізі текстових даних, оскільки це дозволяє відокремити важливу інформацію від надлишкової та зробити її придатною для подальшого аналізу. Крім того, такий текст можна класифікувати за певними критеріями, що дозволяє здійснювати різноманітні завдання. В сучасному світі обробка текстових даних є надзвичайно важливим етапом аналізу даних. За останні десятиліття обсяги текстових даних, що зберігаються та оброблюються, значно зросли, завдяки популярності інтернету, соціальних мереж, месенджерів та інших комунікаційних засобів. Це призвело до необхідності вдосконалення методів обробки текстових даних, які забезпечують високу точність та ефективність аналізу. Результати аналізу текстових даних можуть бути корисними для прийняття рішень в різних сферах, таких як маркетингові дослідження, політичні аналізи, аналізи наукових досліджень та багато інших. Тому важливо розуміти, як правильно обробляти та аналізувати текстові дані для отримання максимальної користі з них.

Щоб ефективно використовувати текстову інформацію, потрібні відповідні інструменти аналізу і обробки даних. У цьому розділі буде розглянуто експериментальне дослідження системи аналізу великих масивів текстової інформації за ключовими словами та фразами. З метою підвищення показників якості аналізу великих масивів текстової інформації було проведено серію експериментальних досліджень.

3.1 Опис даних для дослідження

Як вже було зазначено, дані для дослідження можуть бути взяті з різних джерел, таких як бази даних, веб-сторінки, API, файлові системи, тощо. Обрати правильні дані для експериментальних досліджень при аналізі великих масивів

текстової інформації є критично важливим кроком у забезпеченні точності та достовірності отриманих результатів. Необхідно враховувати такі фактори, як джерело даних, обсяг, якість та репрезентативність вибірки. Наприклад, якщо досліджується вплив вживання певного продукту на настрій людей, то необхідно вибрати відповідну вибірку, яка буде представляти цільову аудиторію. Якщо вибірка не буде репрезентативною, то результати можуть бути неточними і неадекватними. Також важливо використовувати дані, які мають високу якість та достовірність. Якщо дані недостатньо якісні, то результати дослідження можуть бути неадекватними і необ'єктивними. Отже, правильне обрання даних для експериментальних досліджень є важливим етапом у забезпеченні точності та достовірності результатів аналізу великих масивів текстової інформації. Репрезентативний текст – це текст, який представляє деякий певний тип мовленнєвої діяльності або жанру в мові, і який може слугувати зразком для аналізу та порівняння з іншими текстами того ж типу.

Формування моделі прийняття рішення по класифікації, відбувається з використанням принципів машинного навчання, а саме навчання із учителем. Тож для цього потрібно зібрати не малу кількість текстових даних, бажано із завчасно відомими типами їх класифікації. Ці дані повинні містити вхідні значення та позначки або класи, до яких вони належать. Для того, щоб належним чином навчити наївний Баєсовий класифікатор, який було обрано для визначення тематики тексту та його належності до певної класифікації, необхідно надавати йому для тренування тексти, що є структурованими та не несуть зайвої інформації, а також ті, класифікація яких або вже є зазначеною, або не є важким завданням визначити її самостійно. Навчання наївного Баєсовий класифікатора проводилось для української, англійської та німецької мов.

Більше за все нас цікавить українська мова, бо вона не є поширеною настільки, наскільки є поширеними англійська та німецька. Для навчання було використано різні корпуси текстів, які створені для мови, що досліджується. Для англійської мови використано корпуси, такі як «Brown Corpus» та «Reuters Corpus», які містять тексти з різних джерел, включаючи новинні статті, наукові

публікації, художні тексти та інші. Для української мови використано «Український національний корпус текстів» та «Корпус сучасної української мови». Для німецької мови – «German Reference Corpus» та «German Newspaper Corpus». Ці корпуси є найбільш поширеними та включають у себе багато текстів за різними тематиками, стилями, розмірами тощо. Кожен з названих корпусів містить свій унікальний набір текстів, які можуть бути репрезентативними для певної сфери, жанру або періоду часу, а також підходять для задачі класифікації.

До того ж для збору даних було використано API «Вікіпедії» для навчання наївного Баєсового класифікатора. Веб-ресурс «Вікіпедія» містить багато статей на різних мовах, тому для навчання кожної моделі по кожній мові було використано тільки тексти відповідної мови. До того ж цей ресурс містить багато різних текстів з різноманітними тематиками та стилями, що також є хорошою ознакою для тренування. Статті даного сайту завчасно мають в своєму описі список категорій, таким чином вдається автоматизувати процес навчання із вчителем.

Неможна також забувати про попередню обробку тексту, котра дозволяє покращити швидкість роботи алгоритмів, що має особливу важливість в задачах реального часу, таких як аналіз соціальних медіа, класифікація повідомлень тощо. Цей процес є важливим етапом у побудові моделей машинного навчання, оскільки вона допомагає покращити якість та швидкість роботи алгоритмів, знизити розмірність простору векторів та забезпечити більш точне інтерпретування результатів.

3.2 Аналіз результатів

Метою дослідження є підвищення ефективності оброблення та аналізу текстів з метою отримання важливої інформації. Під ефективністю розуміється те, наскільки швидко виконується аналізу текстових даних, та якою є точність його результатів. Для підвищення показників якості, а саме для прискорення процесу аналізу великих масивів текстової інформації без втрати точності, або її

підвищення, було проведено експериментальні дослідження, де наївному Баєсовому класифікатору надавався для аналізу певний відсоток слів з тексту, слова були взяті з векторизованого масиву, який був сформований в попередній обробці тексту. Для аналізу використовувались тексти, які включають в себе десятки тисяч слів. Кількість слів у тексті кожної мови знаходилась у діапазоні від 50 тисяч слів до 70 тисяч. Мова кожного тексту відповідала мові, якою було навчено відповідну модель. Ці тексти було використано для аналізу після навчання кожної з моделей. Так як для дослідження по кожній мові було використано декілька тестів, показники якості для кожної мови є усередненими.

Графіки на рисунках 15, 18, 21 показують загальну тенденцію показників якості аналізу великих масивів текстової інформації для англійської, німецької та української мов, відповідно. Результати дослідження для моделі, що була навчена текстами англійською мовою, представлені на рисунках 15-16, показали, що кращім відсотком для аналізу є проміжок 9%-21% слів тексту. Загальна тенденція на графіку точності показує, що зі збільшенням відсотку проаналізованих популярних слів зменшується точність визначення класифікації для цього тексту.

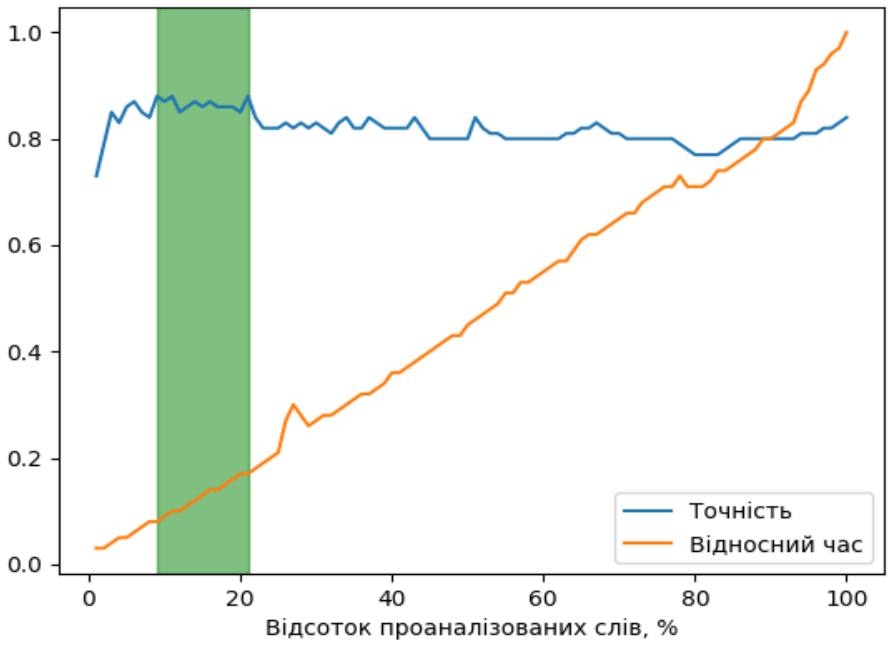

Рисунок 15 – Загальний графік показників якості в залежності від відсотку проаналізованих слів англійською мовою

Графік показників якості в залежності від відсотку проаналізованих слів з рисунку 16 відображує проміжок від 5% до 25%, як видно з нього показники якості на проміжку 9%-21% дають більшу точність у процесі класифікації та менший відносний час класифікації.

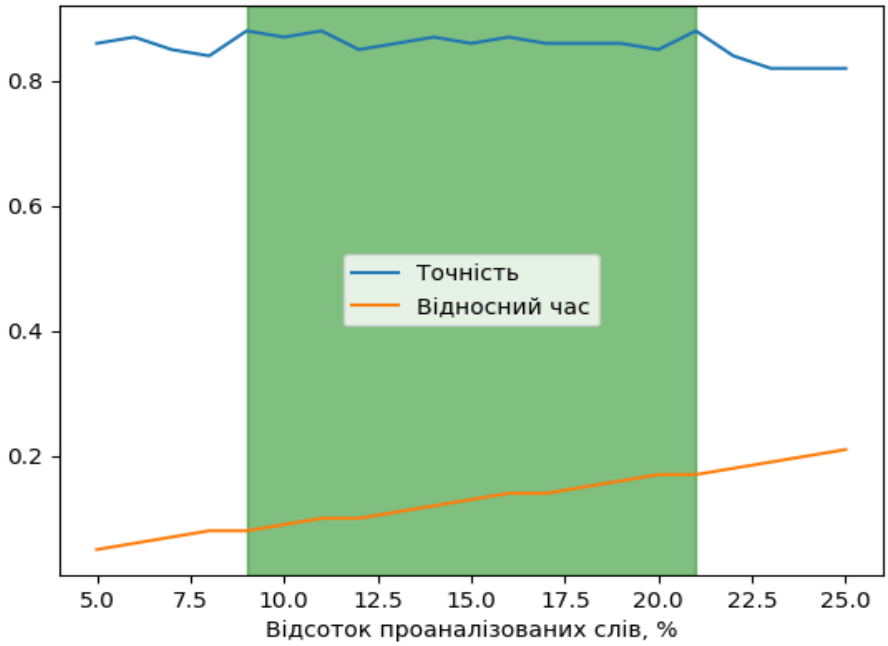

Рисунок 16 – Графік показників якості в залежності від відсотку проаналізованих слів англійською мовою у проміжку 5%-25%

Більш детально значення показників якості в залежності від відсотку проаналізованих слів англійською мовою у проміжку 9%-21% представлені у таблиці 1. Дані з таблиці відповідають даним, що представлені на рисунках 15-16.

Таблиця 1 – Значення показників якості в залежності від відсотку проаналізованих слів англійською мовою у проміжку 9%-21%

%	Точність	Відносний час
9	0.88	0.08
10	0.87	0.09
11	0.88	0.10
12	0.85	0.10
13	0.86	0.11

Продовження таблиці 1

14	0.87	0.12
15	0.86	0.13
16	0.87	0.14
17	0.86	0.14
18	0.86	0.15
19	0.86	0.16
20	0.85	0.17
21	0.88	0.17

З використанням зазначеного проміжку слів у аналізі, текст аналізується швидше та більш точно на відміну від більшого відсотку слів, що можна побачити на рисунку 17 зі значенням для 11%, 60% та 100%. На цій гістограмі наглядно видно, що відносний час аналізу збільшився на 45% та 90% відносно 11% обробленого тексту, а точність зменшилась на 8% та 4%, відповідно.

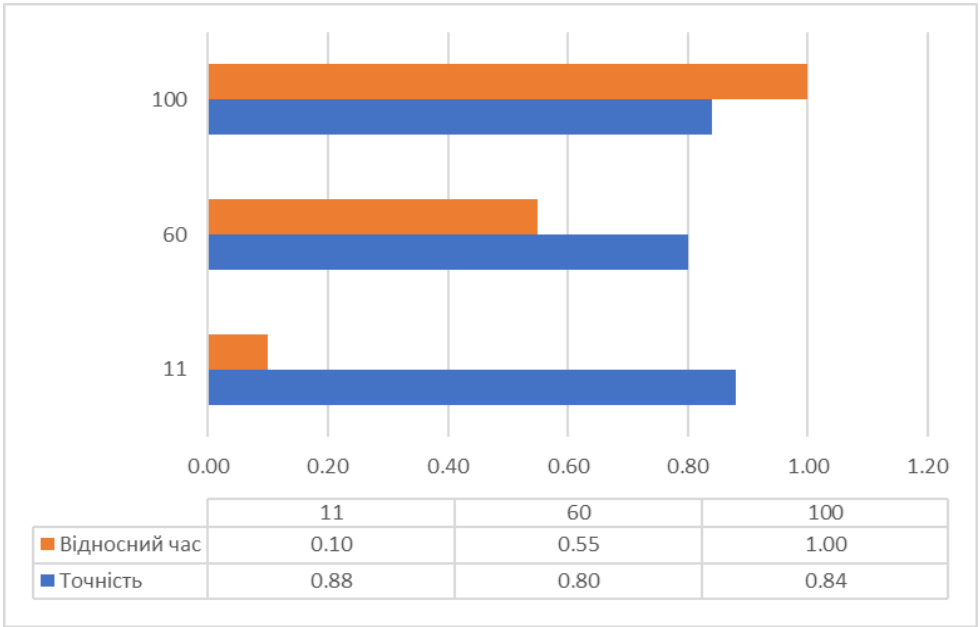

Рисунок 17 – Гістограма показників якості класифікації великих масивів текстових даних в залежності від відсотку проаналізованих слів англійською мовою

В свою чергу, результати дослідження для моделі, де для навчання використовувались німецькомовні тексти, представлені на рисунках 18-19. З них видно, що кращім відсотком для аналізу на цій мові є проміжок 11%-23% слів тексту. Після 23% проаналізованих слів спостерігається спад точності аналізу та відповідне збільшення відносного часу цього процесу. Зі збільшенням відсотку слів з тексту, що приймають участь в аналізі, модель з німецькою мова показує подібні тенденції до моделі, навченою, англомовними текстами.

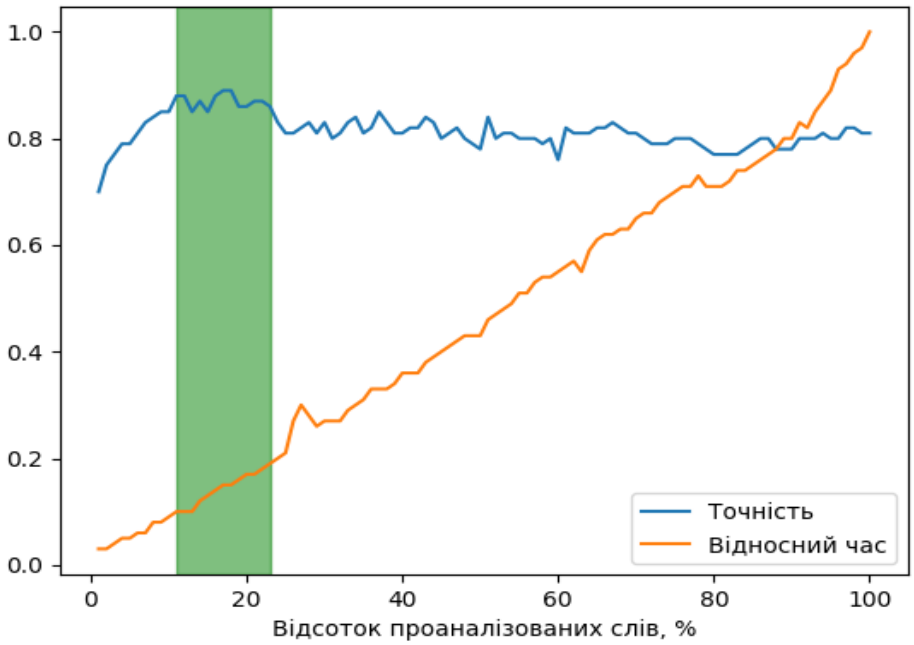

Рисунок 18 – Загальний графік показників якості в залежності від відсотку проаналізованих слів німецькою мовою

Результати дослідження для моделі, що була навчена текстами німецькою мовою, представлені на рисунку 18, показали, що кращім відсотком для аналізу є проміжок 11%-23% слів тексту. Більш наглядно це можна побачити на рисунку 19, де зображений графік показників якості в залежності від відсотку проаналізованих слів німецькою мовою у проміжку 5%-25%. Закономірно, що відносний час аналізу буде збільшуватись відповідно до збільшення відсотку проаналізованих слів, тому показником, який потребує уваги, є точність аналізу. На графіку явно простежується, що цей показник для натренованої моделі показує

більш високі значення точності на зазначеному проміжку порівняно з іншими відсотками проаналізований слів тексту.

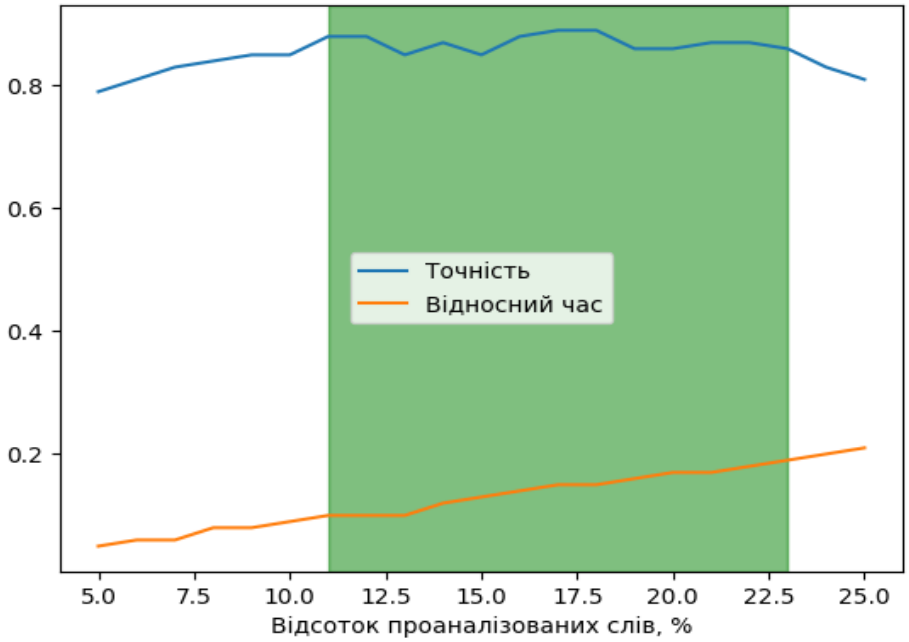

Рисунок 19 – Графік показників якості в залежності від відсотку проаналізованих слів німецькою мовою у проміжку 11%-23%

Дані, які були представлені на рисунках 18-19, подані у таблиці 2. На ній видно, яким значенням відносно відсотку проаналізованих слів відповідають значення точності та відносного часу для моделі, навченої німецькою мовою.

Таблиця 2 – Значення показників якості в залежності від відсотку проаналізованих слів німецькою мовою у проміжку 11%-23%

%	Точність	Відносний час
11	0.88	0.10
12	0.88	0.10
13	0.85	0.10
14	0.87	0.12
15	0.85	0.13
16	0.88	0.14
17	0.89	0.15

Продовження таблиці 2

18	0.89	0.15
19	0.86	0.16
20	0.86	0.17
21	0.87	0.17
22	0.87	0.18
23	0.86	0.19

Як видно з графіків на рисунках 18-19, текст аналізується швидше, при цьому значення точності є на доволі високому рівні порівняно з більшими відсотками слів тексту. Це можна побачити на рисунку 20 з відсотками слів у 17%, 61% та 100%. Відносний час аналізу збільшився на 41% та 85% відносно 17% обробленого тексту, а точність зменшилась на 7% та 8%, відповідно. Ця гістограма має схожі тенденції, як і гістограма, представлена на рисунку 17.

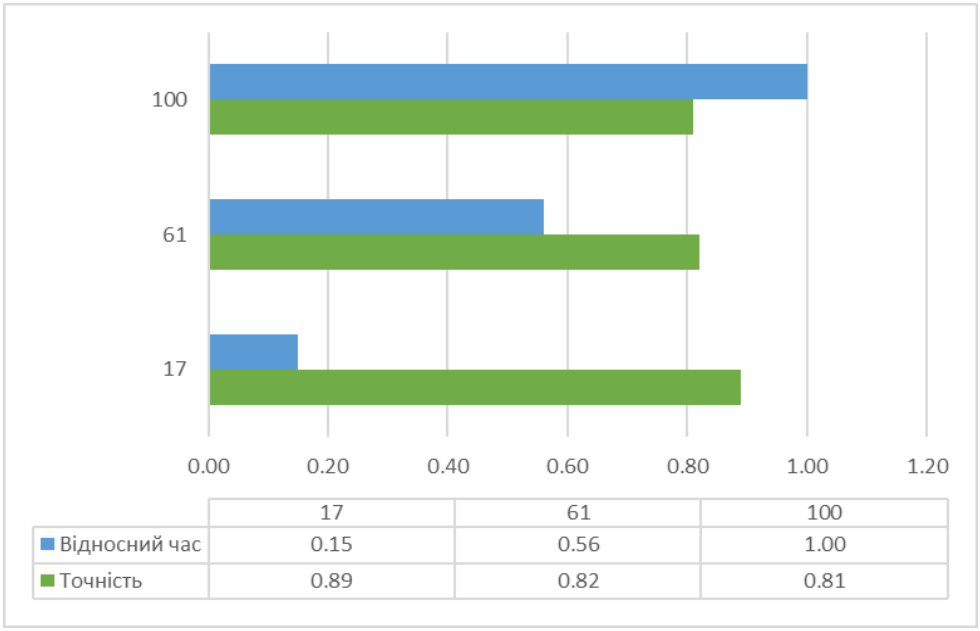

Рисунок 20 – Гістограма показників якості класифікації великих масивів текстових даних в залежності від відсотку проаналізованих слів німецькою мовою

Останньою моделлю, що використовувалась у системі, є модель, навчена для визначення тематики та класифікації текстів написаних українською мовою.

Результати дослідження для моделі, де для навчання використовувались україномовні тексти, зображені на рисунках 21-22. Ці результати показали, що кращім відсотком для аналізу є проміжок 16%-25% слів тексту. Відносно результатів, що були отримані при аналізі текстів, написаних англійською та німецькою мовами, відповідними моделями, проміжок з відсотками слів для української мови має меншу довжину та його нижня межа є вищою порівняно з іншими. При тому, що моделі були натреновані однаковою кількістю текстів, які містили у собі у середньому однакову кількість слів, точність аналізу для української мови відрізняється. Це може бути зумовлено тим, що українська мова є досить складною для аналізу порівняно з англійською та німецькою, оскільки має багато нюансів і варіантів написання слів, а також відмінювання і різні форми одних і тих же слів, тому потребується більше часу та зусиль для створення ефективних систем аналізу тексту українською мовою. Це говорить про те, що моделі, створені для аналізу україномовних текстів, треба навчати більше та довше порівняно з моделями для англійської та німецької мов.

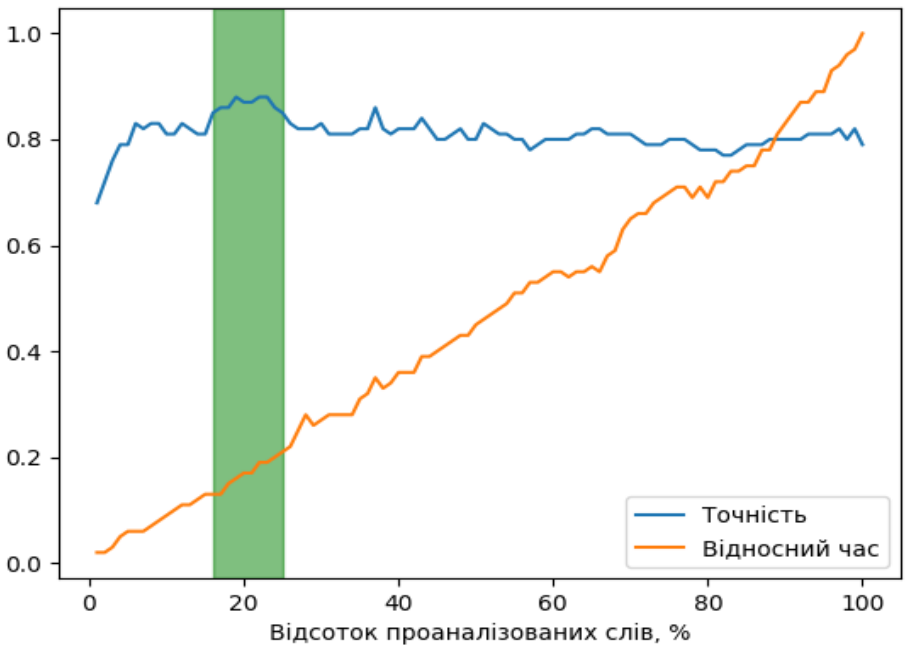

Рисунок 21 – Загальний графік показників якості в залежності від відсотку проаналізованих слів українською мовою

Наступний графік, що знаходиться на рисунку 22, містить значення показників якості в залежності від відсотку проаналізованих слів українською мовою у проміжку 5%-25%. Точність аналізу для української мови на графіку є більш пологою, та зростання цього показника відмічалось на проміжку 16%-25%, що й було виділено на графіку.

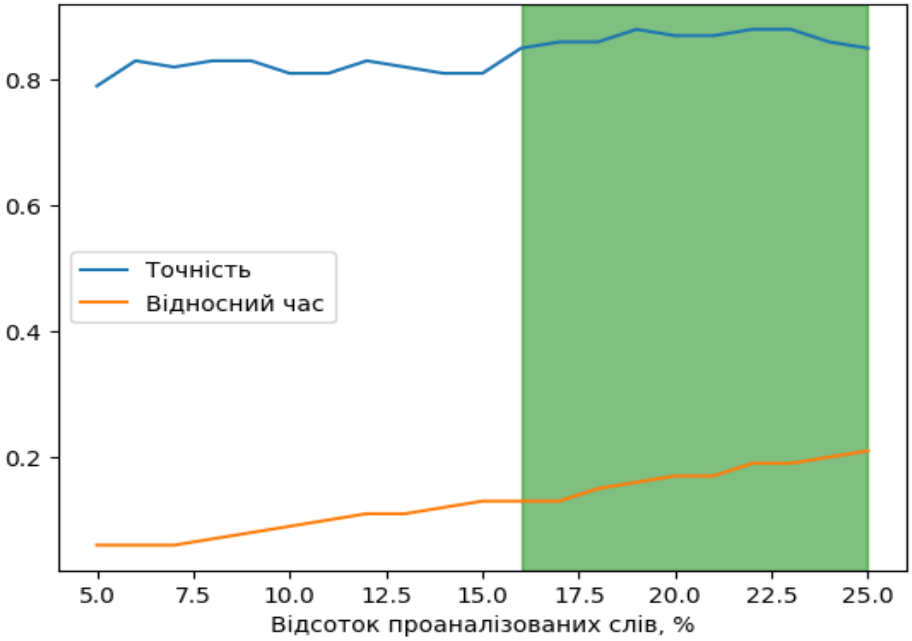

Рисунок 22 – Графік показників якості в залежності від відсотку проаналізованих слів у проміжку 5%-25% українською мовою

У таблиці 3 подано дані, що відповідають рисункам 21-22, та більш детально показують показники якості відносно відсотків проаналізованих слів для української мови. Це надає змогу зробити більш точний аналіз процесу визначення тематики та класифікації натренованими моделями для різних мов.

Таблиця 3 – Значення показників якості в залежності від відсотку проаналізованих слів українською мовою у проміжку 16%-25%

%	Точність	Відносний час
16	0.85	0.13
17	0.86	0.13
18	0.86	0.15

Продовження таблиці 3

19	0.88	0.16
20	0.87	0.17
21	0.87	0.17
22	0.88	0.19
23	0.88	0.19
24	0.86	0.20
25	0.85	0.21

На основі зазначеного проміжку слів у аналізі, текст аналізується швидше та точніше на відміну від використання більшої кількості відсотків слів. Порівняльний аналіз можна побачити на рисунку 23, що містить гістограму зі значенням показників якості для 19%, 65% та 100% проаналізованих слів. На цій гістограмі видно, що відносний час аналізу збільшився на 40% та 84% відносно 19% обробленого тексту, а точність зменшилась на 6% та 9%, відповідно.

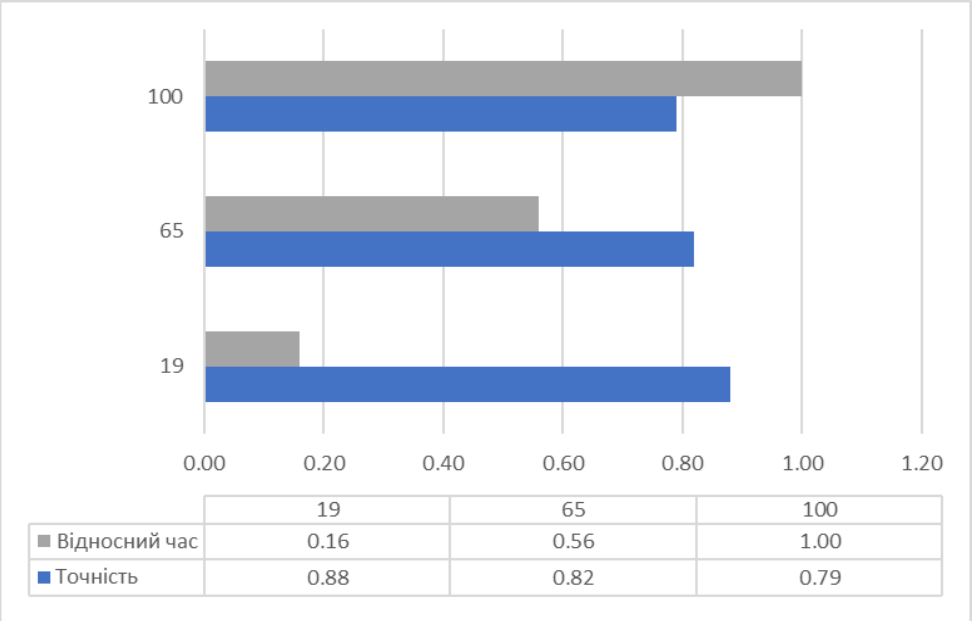

Рисунок 23 – Гістограма показників якості класифікації великих масивів текстових даних в залежності від відсотку проаналізованих слів німецькою мовою

На основі проведеного експерименту, можна сказати, що українська мова є складнішою для аналізу, порівняно з іншими мовами, зокрема з англійською та німецькою. Однією з причин цього є те, що українська мова має складну морфологію і багато форм слів. Також, в українській мові є багато синонімів та слів з близькими значеннями, що ускладнює процес аналізу. Задля покращення аналізу текстів українською мовою пропонується навчати модель на більшій кількості текстів, порівняно з моделями для англійської та німецької мов.

До того ж забагато інформації, що надається класифікатору, починає вводити його в оману, через те, що в тексті можуть бути присутні рідковживані слова, які мають високу вагу, бо їх вірогідність появи дуже низька, але вони заважають якісно оцінити та проаналізувати текст. Загально прийнято проводити аналіз на основі 100% тексту, і що нетреба нехтувати жодним словом, але якщо існує потреба проаналізувати великі масиви текстових даних швидше, то це можна зробити описаним чином.

Для аналізу було використано наївний Баєсовий класифікатор. Задля підвищення показників якості цього процесу пропонується аналізувати певний відсоток слів тексту, що представляються у вигляді векторизованого масиву, а саме, для англійської мови краще підходить використання проміжку 9%-21% слів тексту, для німецької мови – 11%-23%, а для української мови – 16%-25%. Таким чином зменшується час аналізу та підвищується його точність. Область аналізу текстових даних є досить активною галуззю досліджень і розвитку, оскільки є багато різноманітних сфер застосування.

Для досягнення більшої точності є доцільним подальше тренування моделей, особливо, україномовної моделі, для якої межі проміжку відсотків слів тексту для аналізу суттєво відрізняються від проміжків моделей, які аналізують англійську та німецьку мови.

У майбутньому можна очікувати подальшого розвитку цієї області, наприклад, може бути досліджене використання вузькоспеціалізованих слів, використання не тільки популярних слів, а й більш рідковживаних, використання довгих слів та абревіатур та інше.

Результати дослідження опубліковано у тезах IV Міжнародна науково-практична конференція молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2023)» на тему «Аналіз великих масивів текстової інформації з використанням ключових слів та фраз засобами штучного інтелекту» Коваль Ю.В. [34].

3.3 Порівняння з існуючими системами аналізу тексту

Аналіз тексту є важливою складовою багатьох бізнес- та дослідницьких задач, оскільки збільшення об'єму текстових даних, які генеруються, дуже швидко приводить до труднощів у їхньому аналізі і зрозумінні. Тому, для забезпечення ефективності та точності аналізу тексту, існує багато підсистем та інструментів. Ці підсистем можна використовувати для різних цілей та задач. Деякі з них зосереджені на витягненні інформації з тексту, такої як ключові слова, сутності, факти та зв'язки між ними, інші забезпечують можливість класифікації тексту за темою, настроєм, аудиторією тощо. Кожна підсистема підходить для певних задач. Ось деякі з найбільш відомих інструментів для аналізу тексту:

- RapidMiner – програма для машинного навчання та аналізу даних, яка також містить інструменти для обробки тексту, включаючи векторизацію, фільтрацію, токенізацію та інші;
- IBM Watson Natural Language Understanding – це інструмент для аналізу тексту від IBM, який використовується для виявлення настроїв, екстракції іменованих сутностей та категоризації тексту, він використовує штучний інтелект для аналізу та інтерпретації текстової інформації, включаючи аналіз емоцій, ключових слів та іменованих сутностей;
- Amazon Comprehend – це хмарна служба аналізу тексту, розроблена компанією Amazon Web Services, яка надає можливість розуміти значення текстових даних шляхом виявлення мови, сутностей, ключових слів, емоцій та настроїв у тексті.;

– Google Cloud Natural Language API – це інструмент для аналізу тексту від Google, який надає інструменти для аналізу настроїв, екстракції іменованих сутностей, класифікації тексту та іншого.

Це список найбільш розповсюджених та популярних рішень, що використовують для аналізу тексту на основі різних методів і підходів. Кожна з цих підсистем підходить для певних потреб.

RapidMiner – це програмний інструмент для аналізу даних, який дозволяє користувачам виконувати різні операції з даними, включаючи обробку, аналіз та візуалізацію. Цей продукт містить багато алгоритмів машинного навчання, які дозволяють побудувати моделі для прогнозування, класифікації, кластеризації та асоціаційного аналізу даних.

RapidMiner є комерційним інструментом для аналізу даних, який також має функціонал для обробки текстів. Тобто він потребує платну ліцензію для користування. Цей інструмент використовується в більшості у галузях індустрії та бізнесу, таких як медицина, виробництво, телекомунікації, маркетинг та інші. Основними функціями RapidMiner є:

– токенізація;
– стемінг;
– видалення стоп-слів;
– визначення ключових слів та фраз;
– створення тематичних моделей.

Ця система має візуальний інтерфейс, який називається RapidMiner Studio. Це інтегроване середовище розробки, яке надає графічний інтерфейс для створення, налагодження та виконання процесів аналізу даних. RapidMiner Studio має широкий набір інструментів та функцій, що дозволяють аналізувати дані, включаючи обробку тексту, візуалізацію даних, машинне навчання, аналіз соціальних мереж та багато іншого. Інтерфейс RapidMiner Studio базується на технології перетягування та відпускання, що дозволяє користувачам створювати складні процеси аналізу даних, просто перетягуючи та з'єднуючи різні компоненти, що відповідають за різні етапи аналізу. Інтерфейс цієї IDE може бути

відносно складним для звичайного користувача, оскільки це програма для аналізу даних, яка має досить широкі можливості та функціонал. Однак, для користувачів, які мають певний досвід у роботі з програмним забезпеченням, її інтерфейс може бути досить інтуїтивно зрозумілим і легким у використанні. Крім того, програма має документацію та навчальні матеріали, які допоможуть користувачам ознайомитися з її функціоналом та використовувати її ефективно. Загальний інтерфейс RapidMiner Studio представлено на рисунку 24 [32]:

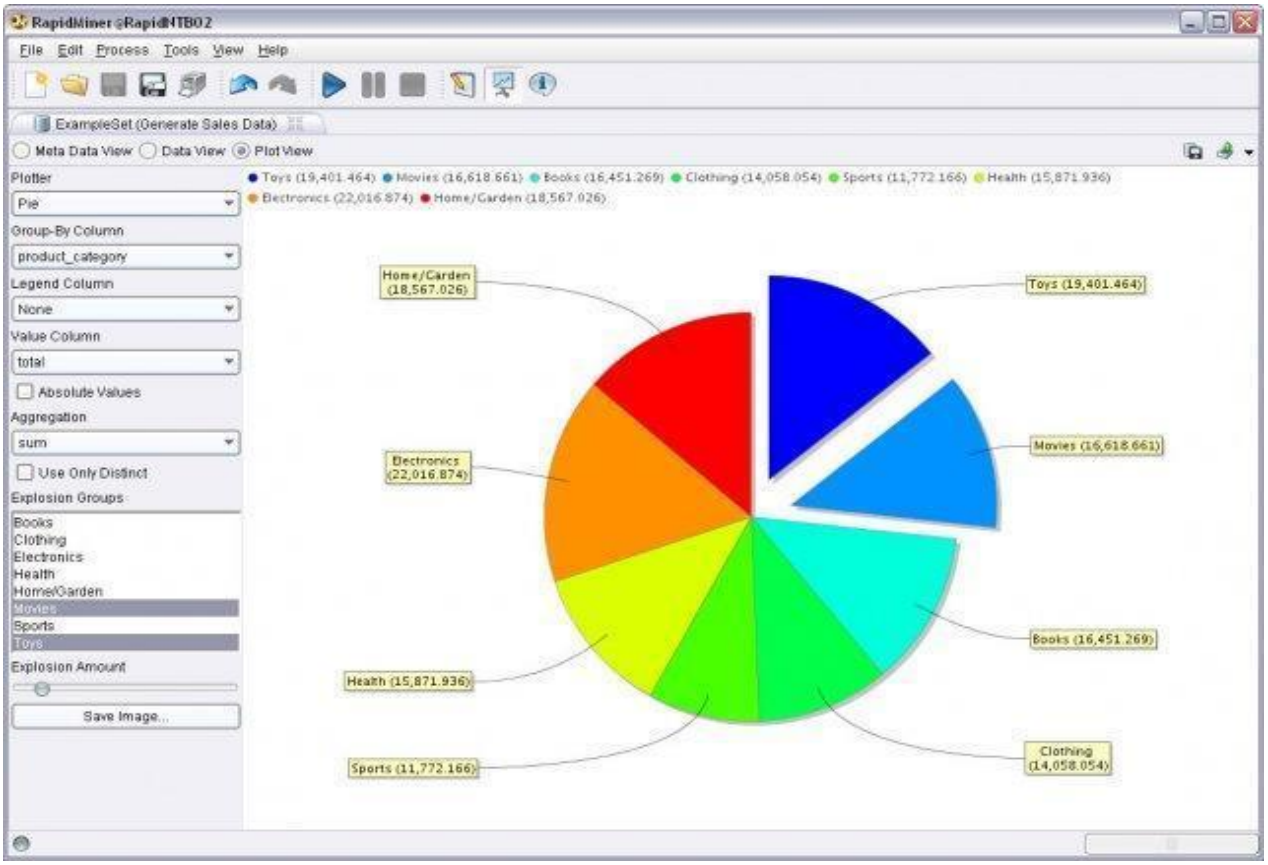

Рисунок 24 – Інтерфейс RapidMiner Studio

Крім того, RapidMiner має вбудовану платформу RapidMiner Server, яка надає можливість створювати та розгортати моделі машинного навчання в хмарному середовищі, та RapidMiner Go, яка дозволяє користувачам легко створювати та запускати проекти з аналізу даних у хмарі.

IBM Watson Natural Language Understanding є однією з найбільш потужних систем для обробки та аналізу тексту. NLU використовує ряд технологій

машинного навчання, щоб забезпечити точний та повний аналіз тексту. Програма може визначити емоційний настрій тексту, ключові слова, назви організацій та людей, а також здійснювати аналіз контенту. Цей інструмент має велику кількість функцій, основними з них є:

- визначення сутностей – ідентифікація та класифікація іменованих сутностей, таких як люди, місця, організації, дати та інші, в тексті;
- визначення ключових слів та фраз – ідентифікація та визначення найбільш важливих слів та фраз в тексті;
- аналіз емоцій – визначення тону тексту, включаючи позитивний, негативний та нейтральний настрій;
- аналіз концепцій – визначення тематичних концепцій, що містяться в тексті, та їх класифікація;
- аналіз синтаксису – визначення відносин між словами в тексті, таких як підмет, присудок та об'єкт, для зрозуміння структури речень;
- аналіз категорій – класифікація тексту за категоріями, такими як спорт, політика, бізнес та інші;
- аналіз відгуків користувачів – аналіз відгуків користувачів на продукти та послуги, включаючи визначення настрою та ключових слів та фраз.

Ці функції дозволяють використовувати IBM Watson Natural Language Understanding для різних завдань, таких як аналіз веб-сторінок, аналіз соціальних медіа, моніторинг брендів та інші.

IBM Watson NLU також має можливість аналізувати тексти на 13 мовах, серед яких англійська, французька, іспанська, німецька та інші. Крім того, вона має інструменти для роботи з веб-сторінками та соціальними мережами, що дозволяє аналізувати текст, що публікується в реальному часі.

IBM Watson Natural Language Understanding є комерційним продуктом з платними планами підписки. Ціна на IBM Watson NLU залежить від кількості запитів та обсягу обробленого тексту. Наприклад, наразі базовий план на 250 запитів в місяць коштує $0,0035 за запит. Продукт має безкоштовну пробну

версію, що дозволяє використовувати сервіс на 30 днів з лімітом в 1000 запитів на місяць.

Він має візуальний інтерфейс у веб-браузері, який дозволяє користувачам завантажувати та обробляти тексти з різних джерел, включаючи соціальні мережі, веб-сторінки, блоги та інші. У візуальному інтерфейсі користувачі можуть обрати різні види аналізу, такі як виявлення ключових слів, аналіз емоцій, визначення сутностей та багато іншого. Крім того, користувачі можуть зберігати та керувати результатами аналізу, створювати звіти та візуалізації даних. Цей сайт має відносно простий та інтуїтивно зрозумілий інтерфейс для використання, що робить його доступним для широкого кола користувачів, включаючи тих, хто не має глибоких знань у сфері програмування та обробки природньої мови. Інтерфейс включає в себе графічний інтерфейс користувача та документацію з прикладами використання, що дозволяє відносно легко відобразити результати аналізу великих масивів текстової інформації. Однак, для повного розуміння можливостей та налаштування параметрів системи, може знадобитися певний рівень знань у сфері програмування та аналізу даних. Інтерфейс IBM Watson Natural Language Understanding представлено на рисунку 25:

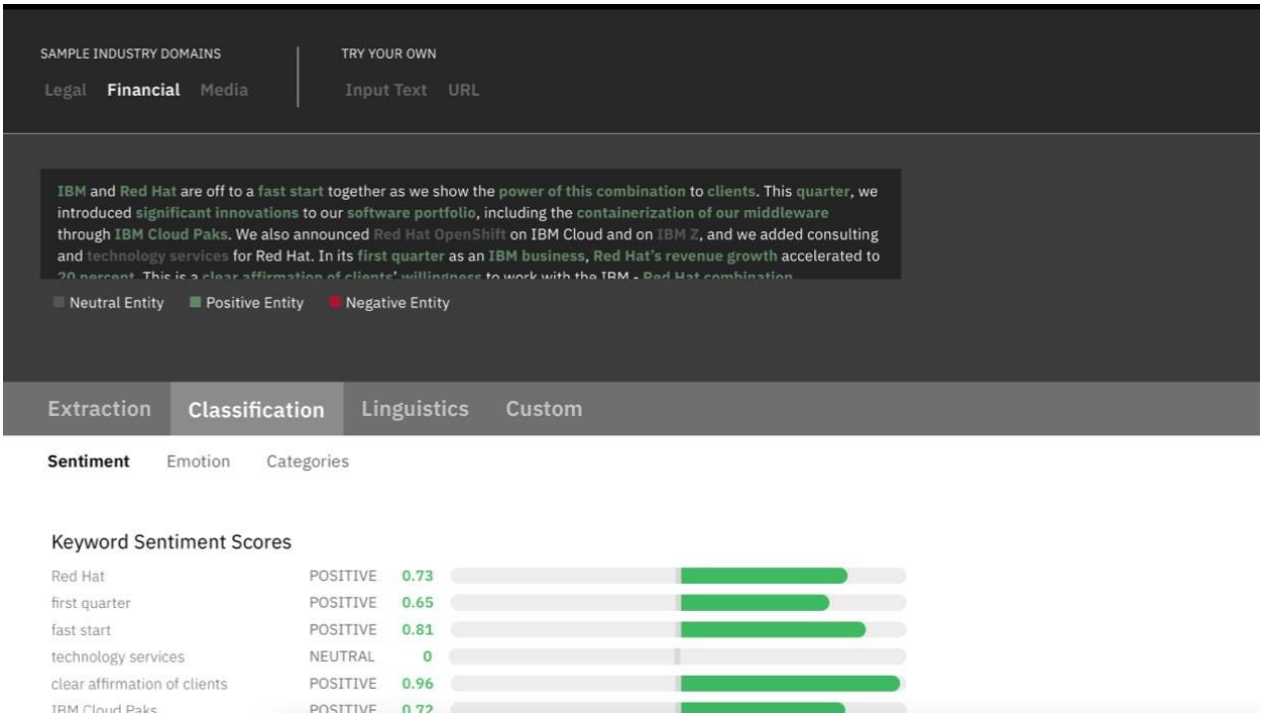

Рисунок 25 – Інтерфейс IBM Watson Natural Language Understanding

IBM Watson Natural Language Understanding також підтримує REST API, який дозволяє розробникам інтегрувати функціонал аналізу тексту у свої додатки та сервіси.

Наступною є Amazon Comprehend - це хмарна служба аналізу тексту, розроблена компанією Amazon Web Services (AWS. Вона дозволяє виявляти і розуміти значення текстових даних, що допомагає отримувати цінні уявлення та інформацію з великих обсягів текстової інформації. Основні можливості Amazon Comprehend включають:

– виявлення мови;
– виявлення сутностей;
– аналіз ключових слів;
– виявлення емоцій;
– аналіз настрою;
– виявлення мовленнєвих актів.

Ці функції дозволяють виконувати аналіз текстових даних, розуміти їх зміст та отримувати цінні висновки для подальшого використання. Amazon Comprehend може бути інтегрований з іншими сервісами AWS та іншими інструментами для розробки додатків. Вартість використання Amazon Comprehend залежить від обсягу оброблюваних даних та використовуваних функціональних можливостей. Ця система має візуальний інтерфейс в межах своєї консолі управління AWS, який дозволяє користувачам завантажувати аудіо- та відеофайли, переводити їх на текст за допомогою різних моделей мовного розпізнавання та переглядати результати перекладу. Інтерфейс також дозволяє виконувати ряд інших функцій, пов'язаних з обробкою і аналізом тексту, таких як пошук, фільтрація, класифікація тощо.

Останнім продуктом є Google Cloud Natural Language API. Це сервіс для аналізу текстів від Google. Він є платним і базується на кількості оброблюваних текстів та використаних функціях. Він надає функціонал для аналізу тексту на

різних мовах, включаючи визначення ключових слів та фраз, визначення сутностей, аналіз емоцій та інше.

Цей API має високу точність та швидкість обробки текстів, завдяки використанню передових технологій глибинного навчання. Функціонал Google Cloud Natural Language API включає в себе:

- визначення сутностей – розпізнавання іменованих сутностей в тексті, таких як місця, люди, організації та інші;
- визначення ключових слів та фраз – API може визначати найважливіші слова та фрази у тексті;
- аналіз емоцій – визначення тону та настрою тексту, а саме, позитивний, негативний або нейтральний;
- аналіз синтаксису – визначення синтаксичної структури речень у тексті та визначення залежності між словами;
- класифікація контенту – автоматичне визначення категорії тексту, таких як новини, спорт, політика тощо;
- аналіз відносин – визначення відносин між сутностями та інші семантичні зв'язки у тексті;
- аналіз сентименту – визначення, чи є текст позитивним, негативним або нейтральним, та визначати його сили;
- аналіз ентропії – визначення ступеню неочікуваності слів у тексті та інші метрики.

Вартість використання Google Cloud Natural Language API різноманітна і залежить від кількості оброблюваних текстів. Наприклад, план «Standard» надає можливість оброблювати до 5 млн символів тексту на місяць і коштує наразі близько 1 долара за 1000 символів, що становить близько 5000 доларів на місяць за максимальний обсяг текстів. Тобто, є необхідною доплата суми, що є пропорційною до кількості символів, що будуть оброблені.

Один з переваг Google Cloud Natural Language API полягає в його готовності до використання – він простий у налаштуванні та інтеграції з іншими інструментами та системами. Цей інструмент може бути використаний

безкоштовно, але з певними обмеженнями. Отже, Google Cloud Natural Language API пропонує широкий функціонал для аналізу текстів з підтримкою різних мов та гнучкість у налаштуванні. Проте, вартість використання може бути досить великою для деяких користувачів і вимагати значних витрат.

Усі згадані системи, в тому чи іншому вигляді, пропонують інструменти для аналізу текстової інформації з використанням технологій глибинного навчання. Однак, кожна з цих систем має свої переваги та недоліки, які залежать від використовуваної моделі, якості даних, мов та інших факторів.

При порівнянні результатів роботи розробленої системи з аналогами були обрані декілька з описаних систем через те, що вони є безкоштовними для персонального використання та мають візуальний інтерфейс, що співпадає з розробленою системою. На рисунку 26 надано гістограми порівняння показників якості розробленої та вже існуючих систем, а саме, гістограма часу аналізу та точності.

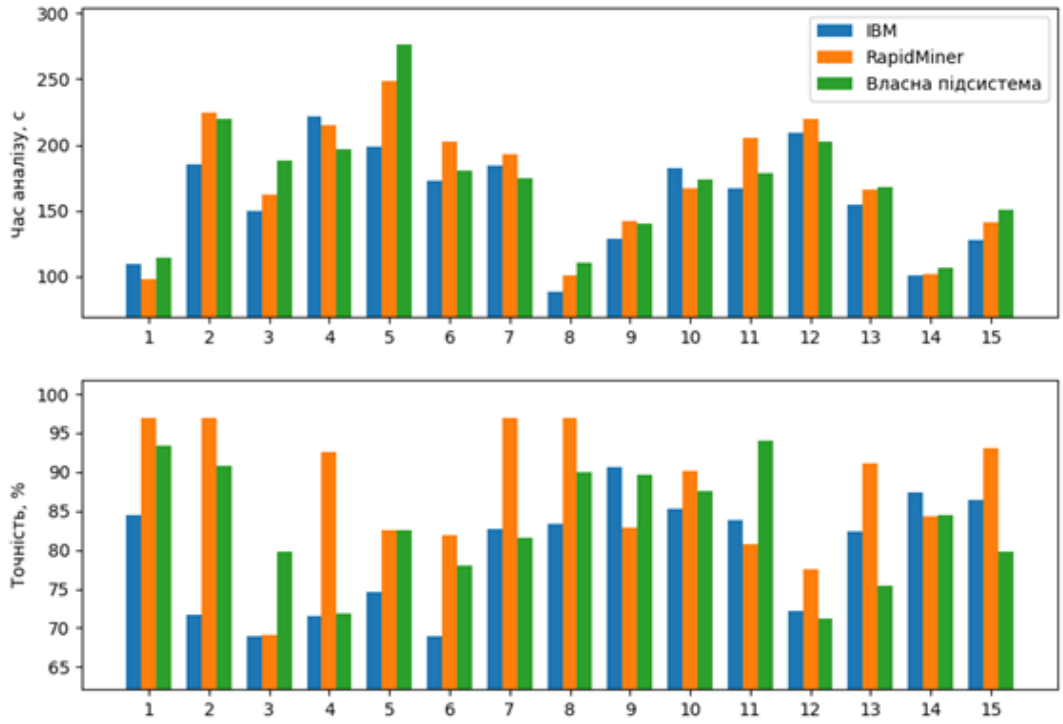

Рисунок 26 – Гістограми порівняння показників якості розробленої та вже існуючих систем при 100% аналізованого тексту

Для порівняння було використано 15 україномовних текстів, які були надані для аналізу таким системам, як IBM Watson Natural Language Understanding, RapidMiner та розробленій системі. Варто зазначити, що розроблена система обробляла 100% слів векторизованого масиву. Як видно з гістограм, час аналізу тексту у розробленій системі у середньому вищий на 8,37% відносно часу аналізу IBM Watson Natural Language Understanding, при цьому час аналізу розробленої системи відносно RapidMiner вищий на 2,15% це при аналізі повного обсягу тексту. Точність розробленої системи у порівнянні з IBM Watson Natural Language Understanding вища на 4,9%, а у порівнянні з RapidMiner нижча на 3,17%. Таким чином, в загальному розроблена система в порівнянні зі вже існуючими при аналізі 100% тексту відстає за часом аналізу та трохи за точністю.

На рисунку 27 представлені гістограми порівняння показників якості розробленої та вже існуючих систем при 19% аналізованого тексту. З них можна побачити, що показники якості системи покращились порівняно зі 100% тексту та іншими системами.

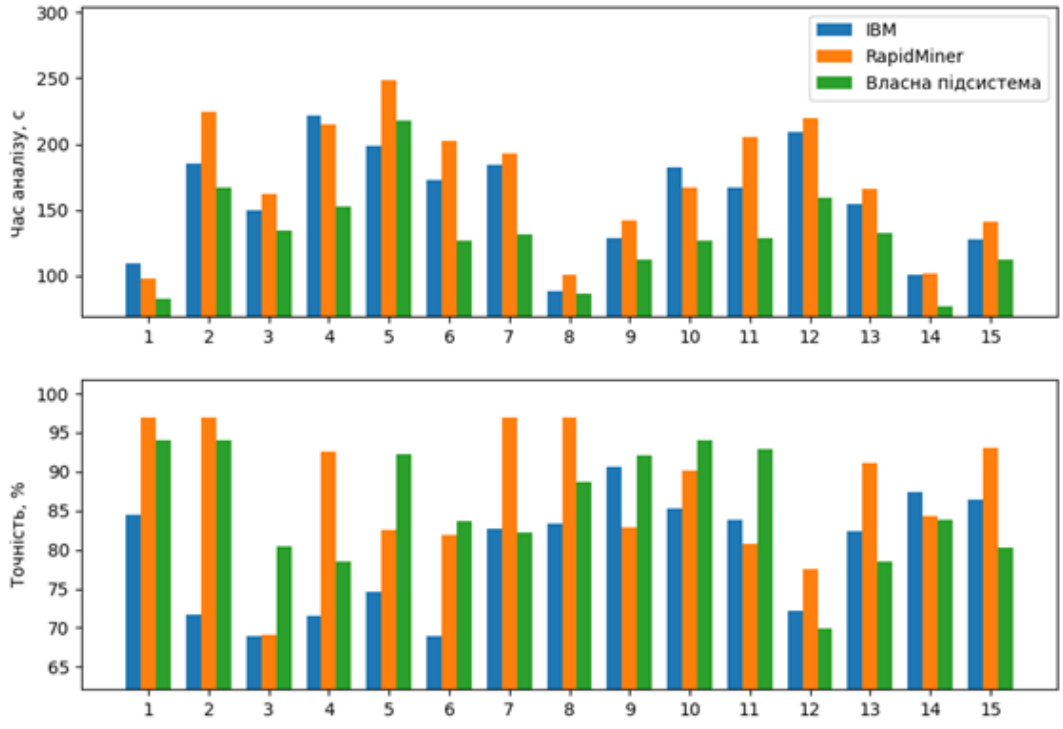

Рисунок 27 – Гістограми порівняння показників якості розробленої та вже існуючих систем при 19% аналізованого тексту

У розробленій системі пропонується використовувати певний відсоток слів з тексту для проведення аналізу, тому для порівняння було проаналізовано 19% слів з векторизованого масиву україномовного тексу у розробленій системі. З представлених гістограм можна побачити, що час аналізу зменшився, при цьому точність збільшилась. У порівнянні з IBM Watson Natural Language Understanding час аналізу зменшився на 14,97%, а точність збільшилась на 5,37%. Показники якості для розробленої системи у порівняні з RapidMiner також змінились, час зменшився на 22,74%, при збільшені точності на 1,7%.

На рисунку 28 показані графіки зміни показників якості розробленої системи з різницею між аналізом 19% та 100% слів векторизованого масиву. З наданих графіків можна зробити висновок, що при аналізі 19% слів аналізу проходить в середньому на 43,53 с швидше, а точність покращується на 4,43%.

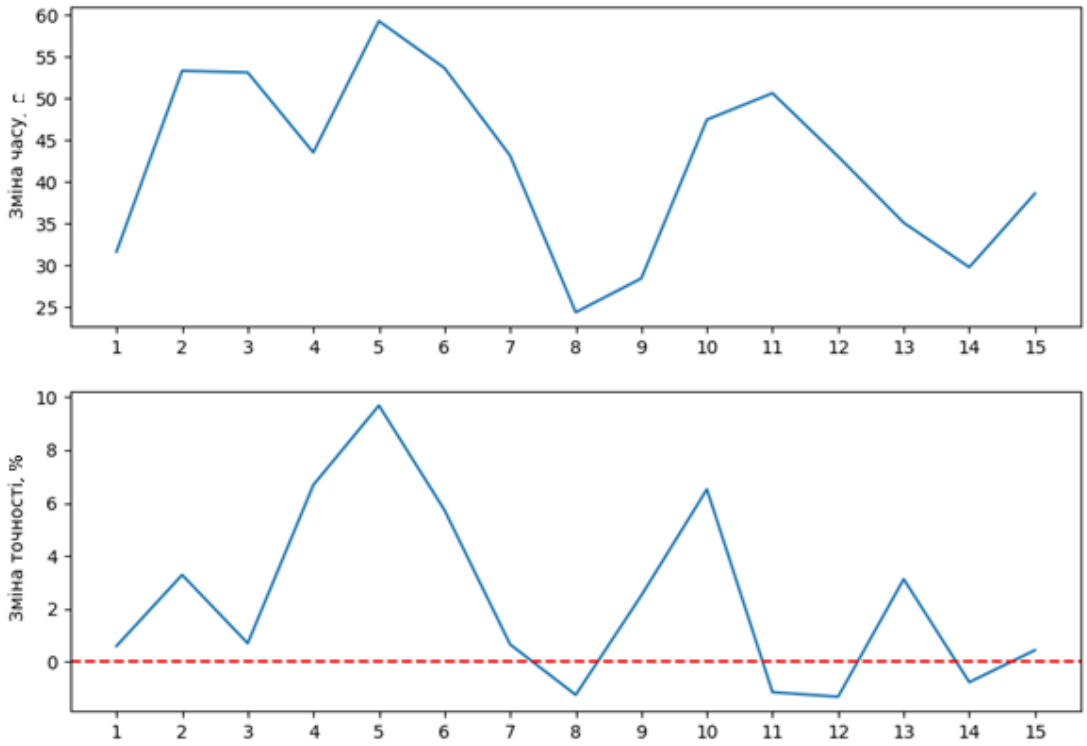

Рисунок 28 – Графіки зміни показників якості розробленої системи з різницею між аналізом 19% та 100% слів векторизованого масиву

Значення, отримані при порівняні систем, трохи відрізняться від значень у експерименті для 19% та 100% для україномовної моделі. Це може бути зумовлене тим, що для порівняння моделі були надані 15 різних текстів, для кожного з них відсоток для аналізу може відрізнятись, також модель треба, безумовно, тренувати й надалі, але в цілому, показники якості процесу аналізу для 19% слів є кращими ніж показники якості для 100%. Можна сказати, що розроблена система з запропонованим алгоритмом аналізу в порівнянні з вже існуючими та доступними звичайному користувачеві системами є більш ефективною для оброблення та аналізу текстів з метою отримання важливої інформації.

У порівнянні з іншими підсистемами аналізу тексту, Google Cloud Natural Language API та Amazon Transcribe може бути менш гнучким та має обмежений функціонал порівняно з RapidMiner та IBM Watson Natural Language Understanding. Однак, їх простота використання та висока швидкість обробки текстів роблять його привабливим вибором для багатьох проєктів, особливо для тих, де потрібна велика швидкість обробки та висока точність аналізу.

Переважна більшість розглянутих систем надає послуги на платній основі, при цьому для деяких з них є необхідною оплата самої послуги, а для деяких є необхідною доплата суми, що є пропорційною до кількості символів, що будуть оброблені. Для великої кількості користувачів та потенційних клієнтів цей показник відіграє вирішальну роль при виборі продукту для аналізу текстових даних. Розроблена ж підсистеми потребує лише входу у Microsoft акаунт. До того ж не всі існуючі системи мають візуальний інтерфейс, а саме, Amazon Transcribe та Google Cloud Natural Language API не мають інтелектуально зрозумілого інтерфейсу, який може використовувати пересічний користувач, тобто ці системи не надають можливість самостійного використання функцій та певних елементів продукту без сторонньої допомоги або додаткових запитів. В той час інші аналоги дозволяють використовувати графічний інтерфейс у вигляді сайту, що роблять IBM Watson Natural Language Understanding та власне рішення, або у вигляді інтегрованого середовища розробки, який пропонує використовувати RapidMiner.

Розроблена система має візуально зрозумілий інтерфейс для будь-якого користувача. Також вона є гнучкою та може бути розширеною відповідно до запитів. Кожна з представлених систем надає доступ до великої кількості мов. Цей показник має безпосередній вплив на спектр можливостей аналізу тексту, що може стати вирішальним фактором при виборі продукту потенційними клієнтами. Наразі розроблена система надає три мови для аналізу текстової інформації, кількість мов також може бути збільшеною за потреби. У всіх досліджуваних систем швидкість обробки даних є в загальному однаковою, тому хоч цей показник має вплив на швидкість отримання результату, він не є значущим. При цьому в розроблену систему було інтегроване рішення, яке покращує показники якості аналізу.

Висновки до розділу 3

На сьогоднішній день аналіз великих масивів текстової інформації з використанням ключових слів та фраз є розповсюдженим та актуальним. Перед початком аналізу потрібно провести попередню підготовку та обробку масиву текстових даних, у підготовці є доцільним наявність наступних етапів: видалення службових елементів, видалення символів, токенізація, приведення до нижнього регістру, видалення стоп-слів, лематизація, векторизація.

Для проведення експериментальних досліджень використано наївний Баєсовий класифікатор. Побудовані на його основі моделі натреновані окремо для англійської, німецької та української мов відповідними текстами. До того ж, тексти, які використані у ході експерименти, знаходилась у діапазоні від 50 тисяч слів до 70 тисяч. Їх було взято з відповідних корпусів текстів, а також зі структурованих ресурсів.

Під час досліджень виявлено, що задля підвищення показників якості процесу аналізу тексту можна аналізувати певний відсоток слів тексту, що представляються у вигляді векторизованого масиву. Для аналізу текстів англійською мовою можна використовувати проміжок 9%-21% слів тексту,

німецькою мовою – 11%-23%, а для українською мовою – 16%-25%. Таким чином зменшується час аналізу та підвищується його точність. Область аналізу текстових даних є досить активною галуззю досліджень і розвитку, оскільки є багато різноманітних сфер застосування. Також з отриманих результатів експерименту було зроблено висновок, що надлишкова інформація може вводити його в оману, через те, що в тексті можуть бути присутні рідковживані слова, які мають високу вагу, бо їх вірогідність появи дуже низька, але вони заважають якісно оцінити та проаналізувати текст.

Для підвищення показників якості розробленої системи надалі можуть бути проведені додаткові експериментальні дослідження, та моделі можуть бути піддані додатковим тренуванням.

У цьому розділі також проведено огляд та аналіз існуючих підсистем аналізу текстової інформації на основі технологій глибинного навчання. Amazon Comprehend, Google Cloud Natural Language API, IBM Watson Natural Language Understanding та RapidMiner. Вони є одними з найбільш популярних інструментів, які використовуються в аналізі текстових даних. Кожен з цих інструментів має свої переваги та недоліки, але в цілому вони забезпечують високу якість та точність аналізу даних, і вибір конкретної підсистеми залежить від потреб користувача та його конкретної задачі. Крім того, багато з цих систем є комерційними, що може впливати на доступність та вартість їх використання. При порівнянні часу та точності аналізу розробленої системи з декотрими існуючими для української мови, зроблено висновок, що з запропонованим алгоритмом аналізу розроблена система впоралась із завданням краще ніж існуючі системи. Також за декотрими характеристиками розроблена система може стати конкурентом до вже існуючих систем.

ВИСНОВКИ

В магістерській дисертації було розроблено та протестовано програмну систему аналізу великих масивів текстової інформації з використанням ключових слів та фраз, яка забезпечить автоматичне визначення ключових слів та фраз. Вона може бути використана для моніторингу та аналізу соціальних мереж та інтернет-форумів, аналізу ринку та конкурентів, пошуку та аналізу новин, аналізу власного контенту, аналізу користувачів або аналізу розпізнаного тексту з голосових повідомлень.

В ході роботи було розглянуто такі проаналізовано існуючі методи та технології аналізу текстів, як переробка даних, обробка природної мови, Natural Language Processing, Баєсова класифікація, Term Frequency-Inverse Document Frequency, кластеризація, метод векторного представлення слів та машинне навчання. Для реалізації системи було вирішено використовувати векторне представлення слів, лематизація, токенізація, Term Frequency-Inverse Document Frequency, класифікація та машинне навчання, бо вони є найбільш ефективними підходами до аналізу текстових даних у існуючих умовах.

Для реалізації процесу аналізу було розроблено алгоритм для оброблення великих масивів текстової інформації з використанням ключових слів та фраз, які дозволять автоматично визначати значущі текстові елементи. Аналіз включає в себе наступні етапи: збір даних, попередній аналіз, токенізація та лематизація, виділення ключових слів та фраз, кластеризація, класифікація, візуалізація, видобуток інформації, аналіз, візуалізація, інтерпретація та висновки.

Розроблену програмну систему було протестовано. Завдяки тому, що у системі використовувались попередньо розроблені та натреновані моделі, результати аналізу на англійській, німецькій та українській мовах є коректними.

При створені програмної системи з використанням розробленого алгоритму для оброблення великих масивів текстової інформації було проведено експериментальне дослідження, де моделям, що відповідають певним мовам, для аналізу надавався певний відсоток слів тексту, що представляються у вигляді

векторизованого масиву. За результатами експериментальних досліджень було зроблено висновок, що моделі, котрі були побудовані на основі наївного Баєсового класифікатора, можуть більш точно аналізувати текст на основі певних проміжків відсотків слів з тексту. Для текстів, що написані англійською мовою, цей проміжок дорівнює 9%-21%, для німецькомовних – 11%-23%, а для україномовних – 16%-25%. В результаті аналізу цих відсотків зменшується час аналізу та підвищується його точність. Різниця в межах проміжків англійської та німецької на відміну він української мови не є великою. Це зумовлено тим, що перші дві мови відносяться до західногерманської групи германських мов, а українська – до східнослов'янської підгрупи слов'янської групи індоєвропейської родини мов. Під час дослідження його мета була досягнена, а саме було підвищено ефективності оброблення та аналізу текстів з метою отримання головної інформації шляхом аналізу певної частини векторизованого масиву слів тексту. В порівнянні з декотрими вже існуючими системами розроблена система з запропонованим алгоритмом аналізу є більш ефективною для оброблення та аналізу текстів з метою отримання важливої інформації. Наукова новизна магістерської дисертації полягає у підвищенні ефективності оброблення та аналізу текстів з метою отримання головної інформації.

Для реалізації системи аналізу великих масивів текстової інформації було запропоновано та детально описано використання технологій. Для створення цієї підсистеми були використані такі технології, як Azure Active Directory в поєднанні з Azure Blob Storage, мова Python та її бібліотеки, а саме NLTK, Beautiful Soup, Matplotlib та WordCloud, HTML, CSS та JavaScript з його бібліотекою Vue.js. Також був реалізований описаний алгоритм аналізу текстових даних за ключовими словами та фразами з їх попередньою обробкою.

Безумовно, створена програмна система може в подальшому бути покращена. Це може бути зроблено за рахунок наступних дій:

- використати вузькоспеціалізованих слів у аналізі;
- використати не тільки популярних слів, а й більш рідких;
- використати довгі слова та абревіатур в аналізі;

- враховувати загального обсягу тексту;
- розширення функціоналу системи;
- додати аналіз настроїв та емоції, що виражені в тексті;
- подальші тренування існуючих класифікаторів;
- відобразити історію аналізу для кожного користувача.

Галузь аналізу текстової інформації є досить перспективною і активно розвивається. З ростом кількості даних в Інтернеті та їх різноманітності зростає і потреба в інструментах для їх аналізу та обробки. Аналіз тексту може бути застосований в різних галузях, таких як маркетинг, медіа, фінанси, наука та багато інших. Застосування методів машинного навчання та штучного інтелекту дозволяє досягати все більш точних результатів та автоматизувати процес аналізу текстової інформації. Тому галузь аналізу текстової інформації має високий потенціал для подальшого розвитку. Розвиток технологій та поява нових методів та інструментів дозволяє здійснювати більш точний та швидкий аналіз текстових даних, що є дуже важливим для ефективного прийняття рішень та досягнення успіху в різних галузях діяльності.

ПЕРЕЛІК ПОСИЛАНЬ

1. What is natural language processing (NLP)?. URL: https://www.ibm.com/topics/natural-language-processing#:~:text=the%20next%20step-,What%20is%20natural%20language%20processing%3F,same%20way%20human%20beings%20can (дата звернення 11.02.2023).
2. Баєсовский классификатор (Bayesian classifier). URL: https://wiki.loginom.ru/articles/bayesian_classifier.html (дата звернення 11.02.2023).
3. Understanding TF-IDF for Machine Learning. URL: https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/ (дата звернення 11.02.2023).
4. Основні поняття кластеризації. URL: http://csc.knu.ua/media/study/asp/mod_probl_inf_tech_sys_analysis_ivohin/lecture/lec11.pdf (дата звернення 12.02.2023).
5. Глибовець А. М., Точицький В. В. Алгоритм токенізації та стемінгу для текстів українською мовою. 2017, 5 с.
6. J. Han, M. Kamber, J. Pei. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2011. 744 с.
7. О.І. Черняк, П.В. Захарченко. Інтелектуальний аналіз даних. 2010, 837 с.
8. Колодчак О.М. Інтелектуальний аналіз даних. 2013, 10 с.
9. Bird, S., Klein, E., & Loper, E.. Natural language processing with Python: analyzing text with the natural language toolkit.. O'Reilly and Associates, 2009. 502 с.
10. С. В. Петрасова, Н. Ф. Хайрова. Сучасні інформаційні технології в лінгвістиці. 2020, 124 с.
11. Наївний баєсів класифікатор). URL: https://uk.unionpedia.org/i/%D0%9D%D0%B0%D1%97%D0%B2%D0%BD%D0%B8%D0%B9_%D0%B1%D0%B0%D1%94%D1%81%D1%96%D0%

B2_%D0%BA%D0%BB%D0%B0%D1%81%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%82%D0%BE%D1%80 (дата звернення 17. 02.2023).

12. Задача класифікації. URL: https://www.wikiwand.com/uk/%D0%97%D0%B0%D0%B4%D0%B0%D1%87%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%97 (дата звернення 18. 02.2023).
13. Naïve Bayes Classifier Algorithm. URL: https://www.javatpoint.com/machine-learning-naive-bayes-classifier#:~:text=Na%C3%AFve%20Bayes%20Classifier%20is%20one,the%20probability%20of%20an%20object. (дата звернення 18. 02.2023).
14. Understanding TF-ID: A Simple Introduction. URL: https://monkeylearn.com/blog/what-is-tf-idf/ (дата звернення: 19. 02.2023).
15. What does tf-idf mean?. URL: http://www.tfidf.com/ (дата звернення: 19. 02.2023).
16. The 5 Clustering Algorithms Data Scientists Need to Know. URL: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68 (дата звернення: 19.02.2023).
17. Обробка природної мови (NLP) у Python з кодом (Частина 3. Кластеризація текстів). URL: https://oleg-dubetcky.medium.com/%D0%BE%D0%B1%D1%80%D0%BE%D0%B1%D0%BA%D0%B0-%D0%BF%D1%80%D0%B8%D1%80%D0%BE%D0%B4%D0%BD%D0%BE%D1%97-%D0%BC%D0%BE%D0%B2%D0%B8-nlp-%D1%83-python-%D0%B7-%D0%BA%D0%BE%D0%B4%D0%BE%D0%BC-%D1%87%D0%B0%D1%81%D1%82%D0%B8%D0%BD%D0%B0-3-%D0%BA%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D1%96%D1%8F-%D1%82%D0%B5%D0%BA%D1%81%D1%82%D1%96%D0%B2-d1db5d9db541 (дата звернення: 20.02.2023).

18. Word2Vec Explained. Explaining the Intuition of Word2Vec & Implementing it in Python. URL:https://towardsdatascience.com/word2vec-explained-49c52b4ccb71 (дата звернення: 20.02.2023).
19. Глубокое обучение и машинное обучение в Машинном обучении Azure. URL: https://learn.microsoft.com/ru-ru/azure/machine-learning/concept-deep-learning-vs-machine-learning (дата звернення: 20.02.2023).
20. Vera Sorin, Yiftach Barash, Eli Konen, Eyal Klang. Deep Learning for Natural Language Processing in Radiology—Fundamentals and a Systematic Review. 2019, 57 с.
21. Abubakr H. Ombabi, Wael Ouarda, Adel M. Alimi. Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks. 2019, 53 с.
22. Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. 2014, 16 с.
23. Luc Dessart, D. John Hillier. Supernovae from blue supergiant progenitors: What a mess! 2018, 25 с.
24. Jeffrey Pennington, Richard Socher, Christopher D. Manning. GloVe: Global Vectors for Word Representation. 2014, 14 с.
25. Pranav Goswami, Akshay Dutt Dubey, Partha Pakray, Gitanjali Roy. Comparative Analysis of Different Approaches for Text Classification Using Vector Space Model. 2018, 10 с.
26. Xiaodong Zhang, Yuhong Li. A Survey of Text Classification Algorithms. 2010, 10 с.
27. Toru Masuda, Hiroaki Matsunaga, Toshifumi Noumi. Proof of new S-matrix formula from classical solutions in open string field theory (or, Deriving on-shell open string field amplitudes without using Feynman rules, Part II). 2016, 16 с.
28. A practical explanation of a Naive Bayes classifier. URL: https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/ (дата звернення: 23.03.2023).

29. Діаграма прецедентів. URL: https://sites.google.com/site/dovidnyk2075/cxefpagxo/uml/diagrama-precedentiv (дата звернення: 23.03.2023).
30. Naive Bayes Classifier Explained: Applications and Practice Problems of Naive Bayes Classifier. URL: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/ (дата звернення: 23.03.2023).
31. What is Sequence Diagram? URL: https://www.visual-paradigm.com/guide/umlunified-modeling-language/what-is-sequence-diagram/ (дата звернення: 25.03.2023).
32. Даталогічна модель бази даних. URL: https://studwood.net/1553750/ekonomika/datalogichna_model_bazi_danih (дата звернення: 12.04.2023).
33. RAPIDMINER TEXT MINING EXTENSION. URL: https://www.predictiveanalyticstoday.com/rapidminer-text-mining-extension/ (дата звернення: 12.04.2023).
34. Коваль Ю.В. Аналіз великих масивів текстової інформації з використанням ключових слів та фраз засобами штучного інтелекту. IV Міжнародна науково-практична конференція молодих вчених та студентів «Інженерія програмного забезпечення і передові інформаційні технології (SoftTech-2023)».