МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Т.в.о. завідувача кафедри
_____ Сафонова С.О.
«____»_____ 2020 р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Дослідження моделей та методів інтелектуального аналізу текстів

Освітньо-кваліфікаційний рівень "Магістр"
Спеціальність 123 "Комп'ютерна інженерія"

Науковий керівник роботи:	_____ (підпис)	С.О. Сафонова (ініціали, прізвище)
Консультант з охорони праці:	_____ (підпис)	Я.О. Критська (ініціали, прізвище)
Студент:	_____ (підпис)	Д. Ю. Іконніков (ініціали, прізвище)
Група:		КІ -18дм

Сєвєродонецьк 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень "магістр"
Спеціальність 123 – "Комп'ютерна інженерія "
(шифр і назва)
Спеціалізація
(шифр і назва)

ЗАТВЕРДЖУЮ:

Т.в.о.зав. кафедри *КНІ*
к.т.н., доц. *С.О. Сафонова*
«_____» _____2020 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Іконнікову Дмитру Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження моделей та методів інтелектуального аналізу Текстів

керівник проекту (роботи) Сафонова С.О., к.т.н., доцент
(прізвище, ім.'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від «11» 10 2019 р. № ____
2. Строк подання студентом роботи 15.01.2020
3. Вихідні дані до роботи Матеріали науково-дослідної практики.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Огляд математичних моделей та методів аналізу текстової інформації та аналіз їх можливостей.

2. Технологія Text Mining як множина методів обробки тексту.
3. Математична модель автоматичного кластерного аналізу текстової інформації.
4. Програмна реалізація аналізу текстової інформації.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	*Критська Я.О.*		

7. Дата видачі завдання *06.09.2020*

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання до магістерської роботи	*10.09.19-17.09.19*	
2	Аналіз завдання, огляд літератури	*18.09.19-25.09.19*	
3	Аналіз технічних засобів	*26.09.19-18.10.19*	
4	Розробка методу	*19.10.19-06.11.19*	
5	Програмна реалізація	*17.11.19-08.12.19*	
6	Охорона праці	*09.12.19-15.12.19*	
7	Оформлення пояснювальної записки	*16.12.19-29.12.19*	
8	Підготовка презентації та доповіді	*03.01.20-12.01.20*	

Студент _____ *Іконніков Д.Ю.*
(підпис) (прізвище та ініціали)

Науковий керівник _____ *Сафонова С.О.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Іконніков Д.Ю. Дослідження моделей та методів інтелектуального аналізу текстів.

Метою роботи є розробка методів, що базуються на використанні технології Text Mining, яка дозволяє підвищити якість і швидкість виконання автоматичної кластеризації документів.

У результаті роботи здійснена програмна реалізація системи аналізу текстового масиву.

Ключові слова: інтелектуальний аналіз тексту, технологія Text Mining, стемінг, стоп-слова, ключові слова, лексико-статистичні шаблони, кластеризація текстів.

АННОТАЦИЯ

Иконников Д.Ю. Исследование моделей и методов интеллектуального анализа текстов.

Целью работы является разработка методов, основанных на использовании технологии Text Mining, которая позволяет повысить качество и скорость выполнения автоматической кластеризации документов.

В результате работы осуществлена программная реализация системы анализа текстового массива.

Ключевые слова: интеллектуальный анализ текста, технология Text Mining, стемминг, стоп-слова, ключевые слова, лексико-статистические шаблоны, кластеризация текстов.

ABSTRACT

Ikonnikov D.Yu. Research of models and methods of intellectual analysis of texts.

The aim of the work is to develop methods based on the use of Text Mining technology, which allows to improve the quality and speed of automatic clustering of documents.

As a result of implemented software implementation of the system analysis of the text array.

Key words: intelectual analiz tekstu, technology Text Mining, steming, stop-words, keywords, lexico statistical shabloni, clusterization text.

ЗМІСТ

Перелік умовних позначень..... 7
Вступ…..……..……. 8
1 Теоретичні засади проблеми дослідження моделей та методів аналізу текстової інформації.……. 10
1.1. Поняття моделі і моделювання…….. 10
1.2. Проблеми обробки текстової інформації…… 12
1.3 Технологія Text Mining як множина методів обробки тексту……. 14
1.4 Методологія обробки текстової інформації…… 18
1.5 Постановка задачі дослідження……..... 21
2 Математичні моделі аналізу текстової інформації …… 22
2.1 Метод автоматичного кластерного аналізу у групі методів аналізу текстової інформації…… 22
2.2 Алгоритм побудови лексичних ланцюжків……. 26
2.3 Методи кластеризації на зважених графах…….. 30
2.4 Семантико-синтаксичний та семантичний алгоритм списку……. 32
3 Вибір, розробка і тестування ефективності методів. Аналіз отриманих результатів……... 35
3.1 Видалення стоп слів……. 35
3.1.1 За законом Бредфорда без стемінгу…… 35
3.1.2 За законом Бредфорда зі стемінгом…… 37
3.1.3 Результати…… 39
3.1.4 Словниковий метод……... 41
3.1.5 Метод об ’ єднання…….. 42
3.1.6 Приклад тексту із проведеної попередньою обробкою……. 43
3.2 Виділення ключових слів…… 45
3.2.1 Міра TF-IDF…… 45
3.2.2 F-міра……. 47
3.2.3 Лінгво-статистичні шаблони……. 47
3.3 Комп’ютерна модель аналізу текстової інформації……. 49
3.3.1 Постановка задачі……..…… 50
3.3.2 Обґрунтування функцій програмного продукту…… 51
3.3.3 Обґрунтування системи параметрів ……... 53
3.3.4 Аналіз експертного оцінювання параметрів……. 54

3.3.5 Аналіз рівня якості варіантів реалізації функції……. 57
Перелік джерел посилань до розділів 1-3 ……..…. 58
4.Охорона праці та безпека в надзвичайних ситуаціях 58
4.1. Загальні питання з охорони праці…… 60
4.2. Аналіз стану умов праці
4.2.1. Вимоги до приміщень …… 61
4.2.2. Вимоги до організації місця праці …… 61
4.2.3. Навантаження та напруженість процесу праці …… 62
4.3. Виробнича санітарія
4.3.1. Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу……. 63
4.3.2. Пожежна безпека…… 64
4.3.3. Електробезпека……. 65
4.4 Гігієнічні вимоги до параметрів виробничого середовища……
4.4.1.Мікроклімат……. 65
4.4.2. Освітлення……. 66
4.4.3.Шум та вібрація, електромагнітне випромінювання…… 67
4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій……. 68
4.6 Висновки до розділу 4……. 71
Перелік джерел посилань до розділу 4……. 72
Висновки……. 73
Додаток А Програмний код…… 75
Додаток Б Комп’ютерна презентація…… 83

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ІС – інформаційна система

ЕОМ – електронна обчислювальна машина

ПМ - природна мова

ПП - програмний продукт

ФВА - функціонально-вартісний аналіз

NN - singular common noun

NNP - proper noun

AJ - general adjective

DT - general determiner

NLP - natural language processing

TF-IDT - term frequency - inverse document frequency

ВСТУП

Актуальність. Лінгвістична обробка природномовних текстів є однією з центральних проблем інтелектуалізації інформаційних технологій. Бурхливе зростання кількості електронних документів, що спостерігається в даний час, наочно показує, що традиційні механізми обробки електронних документів не спроможні впоратись з потребами користувачів. Ця тенденція помітна як в мережі Інтернет, так і у корпоративних мережах.

Засоби сучасних ЕОМ, що використовуються для обробки електронних текстів, дозволяють задавати різні обмеження на шукані комбінації слів в тексті, визначаючи обов'язковість або необов'язковість, допустиму відстань між словами та порядок їх знаходження в тексті. Це дає можливість проводити аналіз слова у всіх граматичних формах, точно і повно описуючи можливі способи представлення необхідного змісту в тексті. Для підвищення точності аналізу текстів розробляються методи попередньої лінгвістичної обробки, що вимагає, по-перше, значних обчислювальних витрат для лінгвістичного аналізу індексованої колекції текстів, по-друге, розробки спеціалізованої пошукової машини.

Автоматизоване вилучення знань з тексту є однією з основних задач штучного інтелекту і безпосередньо пов'язане з розумінням текстів на природній мові.

Ще з середини 50-х років минулого століття значні зусилля науковців були спрямовані на розробку математичних алгоритмів та комп'ютерних програм обробки текстів природною мовою. Для автоматизації аналізу та синтезу текстів створювалися різноманітні моделі процесів обробки тексту, а також відповідні алгоритми та структури представлення даних. Традиційно аналіз природномовних текстів представлявся як послідовність процесів - морфологічний аналіз, синтаксичний аналіз, семантичний аналіз. Для кожного з цих етапів було створено відповідні моделі та алгоритми. Для семантики тексту - класичні семантичні мережі та фреймові моделі Мінського, для синтаксису речення - граматики Хомського, системні граматики Холідея, дерева підпорядкування та системи складових Гладкого, розширенні мережі переходів; для морфологічного аналізу розроблено багато різних моделей, орієнтованих на конкретні групи мов.

Задачу автоматизованої аналітичної обробки текстової інформації намагаються вирішити багато іноземних та вітчизняних вчених. Зокрема, ще у 1979 році Кузін Н.Т. [1] описав методи частотної оброки текстової інформації, які згодом були удосконалені в роботах А. Бродера та Д.В. Ланде У своєму навчальному посібнику А.А. Барсегян та М.С.

Купріянов узагальнили дані щодо сучасних методів автоматичного аналізу Data Mining і Text Mining. Проте жоден з описаних методів не забезпечує вилучення з текстової інформації знань. У дослідженнях А.І. Авіленкової надано характеристику основних методів Data Mining, виділено їх переваги та недоліки, акцентуючи увагу на тому, що жоден з описаних методів не здатен вилучати знання з інформації. У роботі дослідницею продемонстровано роботу методу резолюцій Робінсона для порівняння двох простих речень; запропоновано алгоритм порівняння логіко-лінгвістичних моделей текстової інформації за змістом.

Аналіз сучасного стану проблеми, що досліджується. Можна виділити основні проблеми, пов'язані з необхідністю оптимізації моделювання та розробки методів аналізу текстової інформації:

- швидкий ріст обсягу інформації, що міститься в Інтернеті, є причиною все більш і більш зростаючих труднощів пошуку необхідних документів та організації їх у вигляді структурованих за змістом сховищ;
- більшість технологій роботи з текстовими документами орієнтовані на організацію зручної роботи з інформацією для людини, але практично відсутні можливості для передачі смислового змісту тексту, тобто відсутнє семантичне індексування;
- для ефективного вирішення завдання пошуку необхідно розширити поняття традиційного документа: з документом необхідно пов'язати знання, що дозволяють інтерпретувати й обробляти дані, які зберігаються в цьому документі;
- неструктурована інформація становить значну частину сучасних електронних текстових документів.

1 ТЕОРЕТИЧНІ ЗАСАДИ ПРОБЛЕМИ ДОСЛІДЖЕННЯ МОДЕЛЕЙ ТА МЕТОДІВ АНАЛІЗУ ТЕКСТОВОЇ ІНФОРМАЦІЇ

1.1 Поняття моделі і моделювання

Модель – об'єкт-замінник об'єкта-оригіналу, що забезпечує вивчення деяких властивостей останнього; спрощене подання системи для її аналізу і передбачення, а також отримання якісних та кількісних результатів, необхідних для прийняття правильного управлінського рішення [14].

При вирішенні конкретної задачі, коли необхідно виявити певну властивість досліджуваного об'єкта, модель виявляється не тільки корисним, але й часом єдиним інструментом дослідження. Один і той самий об'єкт може мати безліч моделей, а різні об'єкти можуть описуватися однією моделлю.

Єдина класифікація видів моделей відсутня через багатозначність поняття «модель» в науці і техніці. Її можна проводити за різними підставами: за характером моделей і модельованих об'єктів, за сферами додатків та ін.

Під терміном «моделювання» зазвичай розуміють процес створення точного опису системи; метод пізнання, що складається в створенні і дослідженні моделей [2].

Моделювання полегшує вивчення об'єкта з метою його створення, подальшого перетворення і розвитку. Воно використовується для дослідження існуючої системи, коли реальний експеримент проводити недоцільно через значні фінансові і трудові витрат, а також при необхідності проведення аналізу проектованої системи, тобто яка ще фізично не існує в даній організації.

Для формування моделі використовуються:

- структурна схема об'єкта;
- структурно-функціональна схема об'єкта;
- алгоритми функціонування системи;
- схема розташування технічних засобів на об'єкті;
- схема зв'язку та ін.

Всі моделі можна розбити на два великі класи: предметні (матеріальні) і знакові (інформаційні).

Для проектування ІС використовують інформаційні моделі, що представляють об'єкти і процеси у формі рисунків, схем, креслень, таблиць, формул, текстів і т.п.

Інформаційна модель – це модель об'єкта, процесу або явища, в якій представлені інформаційні аспекти модельованого об'єкту, процесу або явища. Вона є основою розробки моделей ІС.

Для створення описових текстових інформаційних моделей зазвичай використовують природні мови. Поряд з природними розроблені і використовуються формальні мови: системи числення, алгебра відношень, мови програмування та ін. Основна відмінність формальних мов від природних полягає в наявності у формальних мов не тільки жорстко зафіксованого алфавіту, але і строгих правил граматики та синтаксису.

За допомогою формальних мов будують інформаційні моделі певного типу – формально-логічні моделі.

При вивченні нового об'єкта спочатку зазвичай будується його описова модель, потім вона формалізується, тобто документується з використанням математичних формул, геометричних об'єктів тощо.

Процес побудови інформаційних моделей за допомогою формальних мов називають формалізацією.

Моделі, побудовані з використанням математичних понять і формул, називають математичними моделями.

Модель повинна враховувати якомога більше число факторів. Однак реалізувати таке положення складно особливо в слабкоструктурованих системах. Тому найчастіше прагнуть створювати моделі досить простих елементів, з урахуванням їх мікро- і макрозв’язків. Це дозволяє отримувати адекватні результати.

Часткова класифікація методів моделювання представлена на рисунку 1.1.

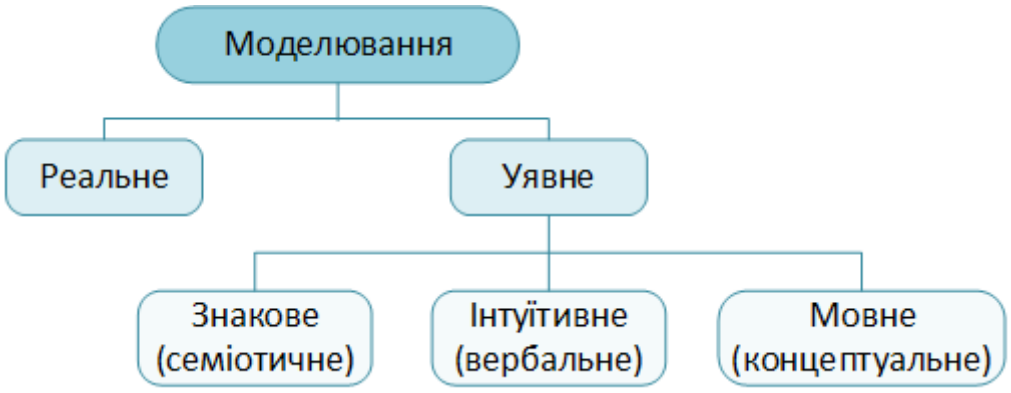

Рисунок 1.1 – Класифікація методів моделювання

Зазвичай розрізняють реальне (матеріальне, предметне) і уявне (ідеалізоване, концептуально-методологічне) моделювання.

Концептуально-методологічне моделювання є процесом встановлення відповідності реальному об'єкту деякої абстрактної конструкції, що дозволяє отримати характеристики об'єкта. Дана модель, як і всяка інша, описує реальний об'єкт лише з деяким ступенем наближення до дійсності.

Концептуальне моделювання є структурованим процесом створення систем, що складається з таких етапів [8]:

1. Аналіз.
2. Проектування.
3. Програмування.
4. Тестування.
5. Впровадження.

Найважливішою формою системного аналізу складних систем є імітаційне моделювання на ЕОМ, що описує процеси функціонування систем у вигляді алгоритмів. Його застосовують у випадках, коли необхідно врахувати велику різноманітність вихідних даних, вивчити протікання процесів в різних умовах. Процес імітації на будь-якому етапі може бути призупинений для проведення наукового експерименту на вербальному (описовому) рівні, результати якого після оцінки та обробки можуть бути використані на наступних етапах імітації.

1.2. Проблеми обробки текстової інформації

При сучасному темпі зростання обсягів інформаційних масивів неважко уявити, якими надмірно трудомісткими процесами будуть, як класифікація всього фонду електронних текстових документів вручну, так і його кластеризація. Допомогти у вирішенні даної проблеми здатні програмні засоби, які автоматично виконують інтелектуальну обробку даних. Останнім часом стало можливим втілення ідеї автоматичної класифікації або кластеризації документів по ряду причин. По-перше, мова йде про текстові документи, які можуть бути представлені у вигляді, придатному для автоматичного аналізу за допомогою програмних засобів. По-друге, на цей момент в науковому співтоваристві накопичився досить великий досвід дослідження і розробки таких систем. Причому інтерес до даної проблеми, не тільки не згасає, але в останні два десятиліття є підвищеним. Це в першу чергу викликано стрибком у розвитку програмно-

апаратної бази, яка стала придатною для тестування розроблених раніше математичних методів інтелектуальної обробки текстів.

Найбільш складні проблеми обробки природномовних текстів зумовлені явищами полісемії, омонімії тощо, які привносять у мову неоднозначність і значно ускладнюють задачу встановлення коректного відображення семантично-синтаксичної структури тексту в його формальне логічне представлення. Всі ці проблеми вирішуються на рівні семантичного аналізу.

З іншого боку, застосування ресурсномістких функцій логічно-семантичного аналізу робить програми обробки тексту занадто складними та повільними. Людина в процесі розуміння тексту не так часто застосовує логіку - лише по мірі виникнення логічних задач, а в решті випадків відбувається застосування інших механізмів, у першу чергу - пошук за асоціацією по за формою чи контекстом.

Пошук за асоціацією - це оцінювання поняття, що відповідає даному слову та є контекстно близьким до свого оточення. При цьому асоціативний пошук є швидким та економічним засобом розв’язання неоднозначності інтерпретації тексту. Тому частина методів пов’язана з визначенням асоціацій за контекстом. Для роботи цих методів необхідна онтологічно-словникова база (онтологія), яка містить інформацію про концепти (поняття) мови, зв’язки концептів зі словами мови та зв’язки між концептами (синонімія, антонімія, гіперо-, гіпонімія та інші). Разом з онтологією використовується ряд алгоритмів, а саме: алгоритм визначення концептів за словами в тексті, алгоритми відновлення значень мовних вказівників на основі онтології, алгоритм визначення тематичної належності слів та понять тексту та тексту в цілому, алгоритм визначення змістовної близькості слів та понять, алгоритм узагальнення.

Пошук за формою мовної конструкції - оцінювання фрази/речення через його форму (синтаксичну структуру, регулярний вираз, наявність визначених елементів), з створенням відповідної реакції на віднайдений шаблон. При цьому вчені намагаються збудувати такі шаблони або маркери, які б дозволяли охопити якомога більше мовних явищ.

Для аналізу складних ситуацій використовуються фрейми. Вони забезпечують найбільш зручний механізм для представлення жорстко структурованих знань про предметну область чи задачу.

Використання всіх цих методів у поєднанні з адаптованими методами статистки суттєво спрощує та прискорює створення систем інтелектуальної обробки текстів та їх використання.

1.3 Технологія Text Mining як множина методів обробки тексту

Інтелектуальний аналіз даних - область знань, яка відноситься до обробки даних, що вивчає пошук і опис прихованих, нетривіальних і практично корисних закономірностей у досліджуваних даних. До задач інтелектуального аналізу даних відноситься множина напрямків, такі як пошук документів в локальних і глобальних мережах, сортування, класифікація і кластеризація документів, автоматичне анотування та реферування, побудова тезаурусів і онтологій, системи автоматичного контролю, діалогові системи, системи, які навчаються, модифікація і поповнення баз знань, експертні системи і машинний переклад. Data Mining - дослідження і виявлення "машиною" (алгоритмами, засобами штучного інтелекту) в сирих даних прихованих знань, які раніше не були відомі, і є нетривіальними, практично корисними, доступними для інтерпретації людиною [1].

Виявлення знань в тексті - це нетривіальний процес виявлення дійсно нових, потенційно корисних і зрозумілих шаблонів в неструктурованих текстових даних (набір документів, що представляють собою логічно об'єднаний текст без будь-яких обмежень на його структуру: web-сторінки, електронна пошта, нормативні документи тощо).

Задачі Text Mining [11]:

– класифікація – визначення для кожного документа однієї та кількох наперед заданих категорій, до якої цей документ відноситься;

– кластеризація – автоматичне виявлення груп семантично схожих документів серед заданої фіксованої безлічі;

– автоматичне анотування – дозволяє скоротити текст, зберігаючи його смисл;

– витяг ключових понять – ідентифікація фактів і відносин в тексті;

– навігація по тексту – дозволяє переміщатися по документам щодо тем і значущих термінів;

– пошук ассоціацій – ідентифікація асоціативних відносин між ключовими поняттями.

Етапи Text Mining представлені на рисунку 1.2.

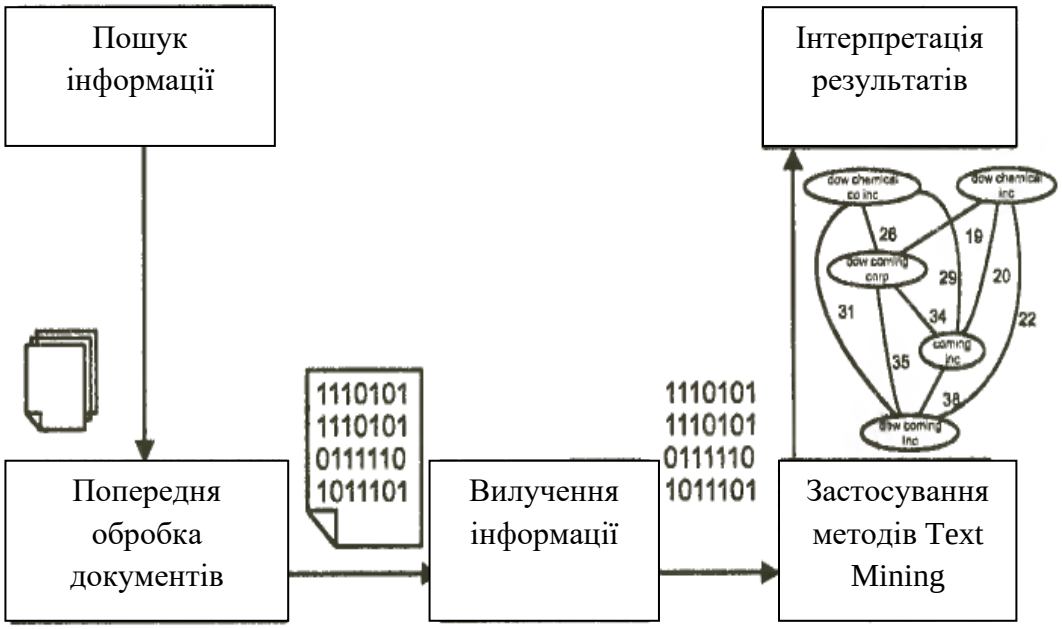

Рисунок 1.2 – Етапи Text Mining

Розглянемо визначені етапи Text Mining. Етап попередньої обробки документів включає:

1. Видалення стоп-слів. Стоп-слова - допоміжні слова, які несуть мало інформації про зміст документа («оскільки», «крім того»).
2. Стеммінг - морфологічний пошук: перетворення кожного слова до його нормальної форми («стиснення», «стислий» -> «стискати»).
3. Приведення регістра: «ТЕКСТ», «Текст» -> «текст»

На етапі вилучення інформації інтерес представляють деякі сутності, події, відносини. Витягнуті поняття аналізуються і використовуються для виведення нових. Витяг ключових понять – фільтрація великих обсягів інформації: відбір документів з колекції, позначка певних термінів у тексті.

Підходи до вилучення інформації з тексту:

– визначення частих наборів слів і об'єднання їх в ключові поняття (Apriori);

– ідентифікація фактів в текстах і витяг їх характеристик: факти – деякі події або відносини; ідентифікація проводиться за допомогою набору зразків; зразки – можливі лінгвістичні варіанти фактів;

– застосування шаблонів.

Вилучення ключових понять за допомогою шаблонів представлено на рисунку 1.3.

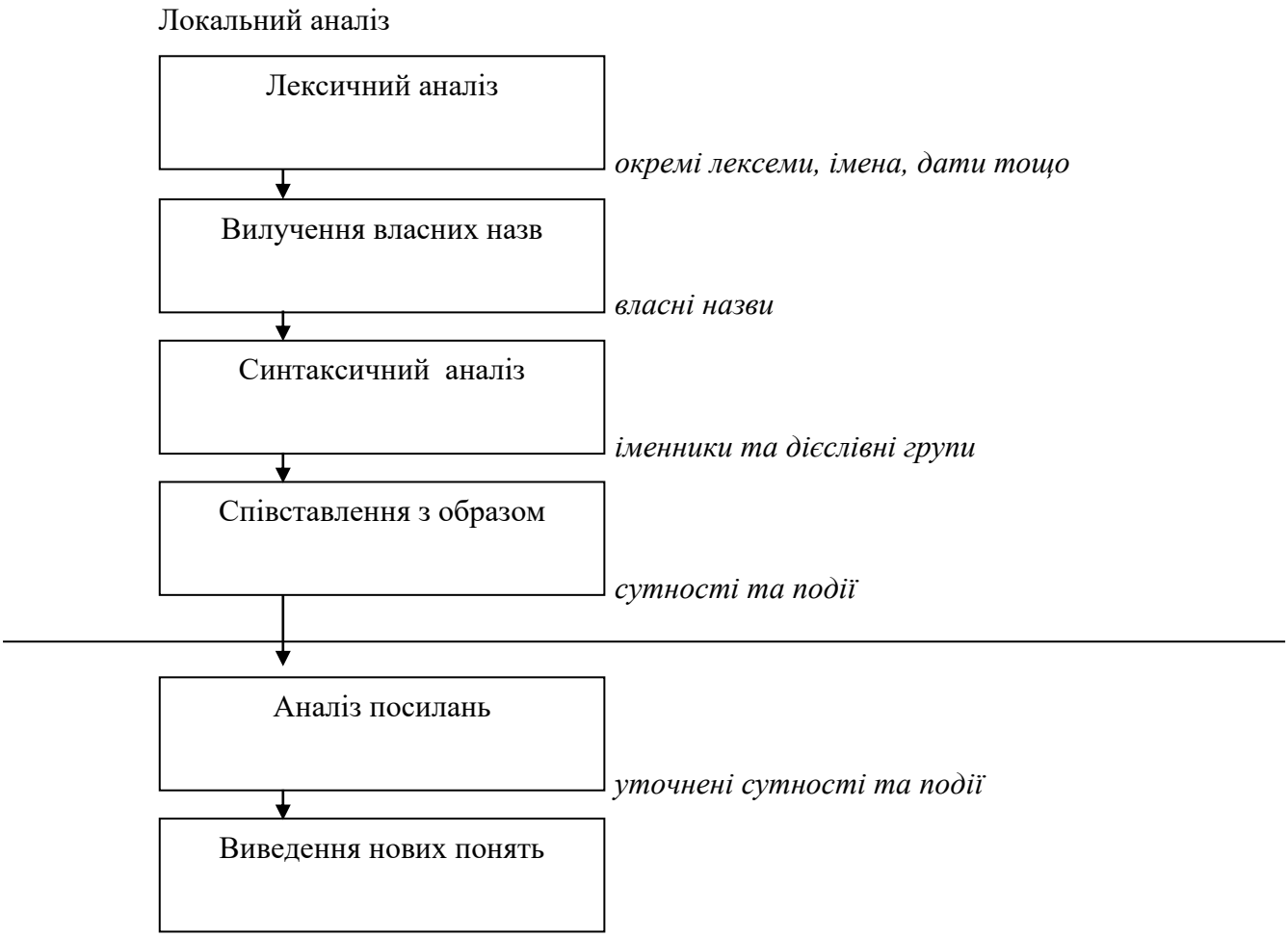

Рисунок 1.3 – Вилучення ключових понять за допомогою шаблонів

Розглянемо більш докладно властивості знань, що виявляються при застосуванні технології Data Mining. Знання повинні бути нові, оскільки зусилля витрачені на відкриття знань, які вже відомі користувачеві, не окупаються. Тому цінність представляють саме нові, раніше невідомі знання. Знання повинні бути нетривіальні, так як результати аналізу повинні відображати неочевидні, несподівані закономірності в даних, так звані приховані знання. Результати, які могли б бути отримані більш простими способами (наприклад, візуальним переглядом), не виправдовують застосування потужних методів Data Mining. Також знання повинні бути практично корисними, бо повинні застосовуватися, в тому числі і на нових даних з досить високим ступенем достовірності. Корисність полягає в тому, щоб ці знання могли принести певну вигоду при їх застосуванні і знання повинні бути доступні для розуміння людиною. Знайдені закономірності повинні бути логічно зрозумілі, в іншому випадку існує ймовірність, що вони є випадковими. Крім того, виявлені знання мають бути представлені в зрозумілому для людини вигляді [2].

Text Mining представляє собою множину методів обробки тексту, в результаті застосування яких з'являються нові, раніше не виявлені знання. Сьогодні це

міждисциплінарна область, у якій використовуються базові технології Data Mining в поєднанні з техніками інших дослідницьких областей, таких як пошук інформації - Information Retrieval, вилучення інформації - Information Extraction, математична лінгвістика, класифікація - Classification, кластеризація - Clustering, створення онтологій - Ontology engineering тощо.

Незважаючи на те, що в кожній з цих областей вирішуються свої специфічні прикладні задачі, часто буває досить складно провести чітку межу між Text Mining та іншою областю досліджень, оскільки всі вони мають справу з текстами, а, отже, - загальні проблеми і підходи до їх вирішення. Так з Information Retrieval в Text Mining запозичені деякі алгоритми та методи обробки тексту. Різниця між цими областями полягає в кінцевій меті. В інформаційному пошуку метою є знайти документи, які хоча б частково співпадали з пошуковим запитом і серед знайдених відібрати ті, для яких збіг є найбільш повним. Методи Text Mining спрямовані на виявлення невідомих «фактів» і прихованих взаємозв'язків, які можуть виявитися по семантичним, лексичним і статистичним ознакам в множині текстів.

Інша область, з якої запозичені методи Text Mining - це вилучення інформації. Information Extraction відрізняється від Text Mining тим, що в цій області розглядаються способи вилучення специфічної інформації, структурованих даних, такі як імена людей, географічні назви, заголовки книг по заздалегідь заданим відносинам. У Text Mining наперед невідомо, яка саме інформація може бути виявлена. Методи Text Mining можна ефективно застосовувати при створенні баз даних та знань.

За допомогою методів Text Mining можна багато чого зробити для вирішення задач інтеграції даних. Вони дозволяють виявити загальні властивості в текстах, узятих з різних джерел. Критерії подібності в Text Mining можуть враховувати синтаксичну та семантичну інформацію і застосовуватися не тільки до слів, але і до фраз, і до більш великих фрагментів тексту. Важливою проблемою в цій області залишається вибір найкращого критерію оцінки відстані між текстами - поняття, яке вперше було запроваджено в Text Mining Область досліджень, яка може значно виграти від використання Text Mining - це інформаційний пошук, тому що виконання запитів вимагає перевірки семантичних відношень між текстами. Застосування Text Mining покращує точність інформаційних пошукових систем і зменшує кількість документів, що повертаються по одному запиту. Класифікація текстів також є частиною Text Mining, тому що наявність структурованого підходу полегшує пошук, перегляд та маркування документів.

Таким чином, для вилучення знань з текстової інформації використовуються різноманітні методи автоматичного аналізу Data Mining. Такі методи використовують

алгоритми та засоби штучного інтелекту для дослідження і вилучення з великих об’ємів інформації знань, які будуть практично корисні та доступні для інтерпретації людиною [5].

1.4 Методологія обробки текстової інформації

До основних методів Data Mining належать класифікація, кластеризація, регресія, пошук асоціативних правил, анотування та автореферування.

Задача класифікації зводиться до визначення класу об’єкту за його характеристиками, при чому множина класів задається завчасно. Класифікація - використовує статистичні кореляції для побудови правил розміщення документів у наперед заданій категорії. Задача класифікації - це задача розпізнавання, коли система відносить новий об’єкт до тієї чи іншої категорії. В Data Mining задачу класифікації розглядають як визначення значення одного з параметрів об’єкту на основі значення інших параметрів [4].

Задача регресії подібна до задачі класифікації і дозволяє визначити за відомими характеристиками об’єкту значення деякого його параметру. Тут значенням параметру є не кінцева множина класів, а множина дійсних чисел.

Класифікація та регресія передбачають здійснення двох обов’язкових етапів. Перший етап - виділення набору об’єктів, для яких відомі значення залежних і незалежних змінних. На основі отриманого набору будується модель визначення значення залежної змінної (функція класифікації або регресії). На другому етапі побудовану модель застосовують до об’єктів, які аналізуються. Недоліком класифікації та регресії є те, що розробник системи повинен фіксувати кількість класів та характеристик, за якими буде проводитись дослідження. Це означає, що якщо система не виявить ознаки або класу, до якого можна віднести, наприклад, текстовий документ, він не буде коректно оброблений.

Анотування - це процес створення коротких повідомлень про електронний текст, які дозволяють робити висновки щодо доцільності його докладного вивчення [б]. Сучасні системи аналітичної обробки текстової інформації володіють засобами автоматичного складання анотацій, при цьому існує два підходи до вирішення цієї проблеми.

У першому підході програма-анотатор вилучає з першоджерела невелику кількість фрагментів, у яких найбільш повно представлено зміст документу. При другому підході анотація представляє собою синтезований документ у вигляді короткого змісту. Анотація, сформована відповідно з першим підходом, якісно поступається анотації, одержаній при

синтезі. Для підвищення якості анотування необхідно вирішити проблему орієнтування на вузьку предметну область. Тоді у такому процесі необхідна участь людини.

Метод анотування тексту довільної структури передбачає:

1. Формування множини анотованих фрагментів, які є цілими реченнями даного тексту, містять у своєму складі дієслово або короткий прикметник, і не є питальним чи окличним реченням.

2. Створення таблиці всіх можливих пар основних тематичних вузлів (тут використовується система продукцій для встановлення характеристик структурних одиниць тексту, описана раніше).

3. Відбір таких речень, які містять декілька різних тематичних вузлів, що не зустрічалися раніше у тексті.

Здійснення автоматичної анотації є прикладною задачею, що вирішується перед тим, як інформація із заданого тексту потрапить до пошукового серверу.

Автоматичне реферування - представляє собою створення коротких викладів матеріалів, анотацій, дайджестів, тобто вилучення найбільш важливих відомостей з одного або декількох документів і генерація на їх основі лаконічних та інформаційно-ємних звітів. На сьогодні існує два основних напрямки автореферування: квазіреферування (засноване на екстрагуванні фрагментів документів, тобто виділенні найбільш інформативних фраз і формування з них квазірефератів) і коротке викладення змісту первинних документів (дайджести) [3].

Автоматичне реферування та анотування використовуються в основному для економії часу користувачам, створення каталогів інформаційних ресурсів, використання словників-тезаурусів загального та спеціального призначення. Застосовується автоматичне реферування та анотування в корпоративних системах документообігу, пошукових машинах та каталогах ресурсів Інтернет, автоматизованих інформаційно - бібліотечних системах, каналах зв’язку, службах розсилки новин і т.д.

Пошук асоціативних правил представляє собою метод пошуку часткових залежностей між об’єктами та суб’єктами. Знайдені залежності представляються у вигляді правил та використовуються для кращого розуміння природи даних, що аналізуються. Тобто з великої кількості наборів об’єктів визначаються такі набори, що найбільш часто зустрічаються. При виявленні закономірностей можна з певною ймовірністю передбачити появу подій у майбутньому, що дозволяє приймати рішення. Така задача є різновидом задачі пошуку асоціативних правил і називається сиквенційним аналізом.

Кластеризація - це розбиття множини документів на кластери (групи документів зі спільними ознаками), які представляють собою підмножини, смислові параметри яких

заздалегідь невідомі. Числові методи кластеризації базуються на визначенні кластера як множини документів. Кластеризація може застосовуватися в довільній області, де необхідне дослідження експериментальних та статистичних даних [4].

Для задачі кластеризації характерний пошук груп найбільш схожих об’єктів. Після визначення кластерів використовуються інші методи Data Mining. Кластерний аналіз дозволяє розглядати великий об’єм інформації та скорочувати, стискати великі масиви інформації. Результат кластеризації залежить від природи даних об’єктів та від представлення кластерів. Кластеризація відрізняється від класифікації тим, що для проведення аналізу не потрібно мати виділену залежну змінну. Задача кластеризації вирішується на початкових етапах дослідження, а її розв’язок допомагає краще зрозуміти дані.

Всі описані вище методи автоматичного аналізу Data Mining забезпечують певну структуризацію текстової інформації, її узагальнення або анотування. Проте для вилучення знань з електронних текстів, зокрема, порівняння текстів та виявлення в них збігів, необхідні засоби автоматичного лінгвістичного аналізу.

Основний метод, що використовується сьогодні для логічного порівняння текстової інформації є метод резолюцій Робінсона. Наприклад, нехай є два простих речення, для кожного з яких побудовано логічну модель. За алгоритмом уніфікації шукаємо підстановку $Q = \{P / P'\}$, x_3 / x'_3. Якщо здійснити підстановку, будемо мати множини, що містять літерали з однаковими предикатами. Після цього, застосовуючи метод резолюцій, отримаємо резольвенту, що не дорівнює пустій множині, що свідчить про те, що речення однакові.

Аналізуючи зміст заданих речень, можна зробити висновки про те, що а даному випадку в алгоритмі уніфікації не можна було застосовувати підстановку P/P' , оскільки предикати різні за змістом і не є синонімами. Метод резолюцій не дає змогу визначити це в процесі заміни, через те, що не аналізує зміст слів, що входять до речення природної мови [7].

Це означає, що для здійснення коректного порівняння текстових документів за змістом необхідні нові алгоритми лінгвістичного аналізу, які забезпечать змістовну обробку текстової інформації. Одним із таких алгоритмів може бути алгоритм порівняння логіко-лінгвістичних моделей речень природної мови, що включає в себе наступні етапи.

1. Побудова логіко-лінгвістичних моделей. На цьому етапі кожному реченню природної мови ставиться у відповідність логічна формула, що представляє собою одновимірний масив слів, з яких складаються речення, упорядковані у відповідності до того, яку синтаксичну роль вони виконують.

2. Ідентифікація. Відбувається почерговий перегляд елементів всіх логіко-лінгвістичних моделей: предикатів, предикатних змінних (суб'єктів), предикатних змінних (аргументів), предикатних констант. Серед складових логіко-лінгвістичних моделей шукаються спільнокореневі слова, синоніми, активні та пасивні форми спільнокореневих дієслів.

3. Заміна тотожних змінних. Якщо на етапі ідентифікації знайдено тотожні змінні, у всіх логіко-лінгвістичних моделях відбувається їх перепозначення, завдяки чому одні й ті самі слова (навіть якщо вони мають різні граматичні рамки) будуть позначатися однаково, і, відповідно, мати ідентичний зміст.

4. Логічний вивід. Після ідентифікації та заміни тотожних змінних застосовується система продукцій, що містить правила порівняння логіко-лінгвістичних моделей. Такі правила дозволяють через встановлені зв'язки між словами переходити до представлення значень слів у вигляді комбінацій елементарних компонентів змісту.

1.5 Постановка завдання дослідження

Таким чином, дослідження засобів та методів інтелектуального аналізу текстів є актуальним завданням для обробки і розпізнавання текстів. Тому ставиться завдання розробки алгоритму аналізу текстової інформації, що базуються на використанні технології Text Mining.

Для цього необхідно вирішити такі завдання:

– провести аналіз існуючих реалізацій як окремих алгоритмів, так і програмних систем, які були створені для структурування інформації, а також практичне порівняння роботи алгоритмів інтелектуального аналізу тексту (Text Mining);

– розробити алгоритм семантичного аналізу тексту;

– реалізувати алгоритм застосування методів структурування даних, здійснити порівняльну статистику цих методів на наборах текстів;

– реалізувати комп'ютерну модель для аналізу текстової інформації.

2. МАТЕМАТИЧНІ МОДЕЛІ АНАЛІЗУ ТЕКСТОВОЇ ІНФОРМАЦІЇ

2.1 Метод автоматичного кластерного аналізу у групі методів аналізу текстової інформації

Кластерний аналіз займає одне з центральних місць серед методів аналізу даних і являє собою сукупність методів, підходів і процедур, розроблених для вирішення проблеми формування однорідних класів (кластерів) у довільній проблемній області.

Під автоматичною кластеризацією текстових документів розуміють процес класифікації колекції текстових документів, який базується тільки на аналізі та виявленні внутрішньої тематичної структури колекції без наявності апріорної інформації про неї, тобто при відсутності визначеного рубрикатора і множини документів-зразків. Класифікація документів з використанням алгоритмів кластеризації призводить до розбиття множини документів на однорідні, у відповідному розумінні, групи або кластери шляхом автоматичного аналізу тематичної близькості між ними. Кластеризація текстів базується на гіпотезі: тісно пов'язані за змістом документи намагаються бути релевантними одним і тим же запитам, тобто документи, релевантні запиту, віддалені від тих, які не релевантні цьому запиту.

Задачу автоматичної кластеризації текстових документів у загальному вигляді можна сформулювати наступним чином: дано множину текстів на природній мові - колекція текстових документів. Передбачається, що існує множина тематичних груп (кластерів), на які можна розбити дану колекцію документів. Тоді задача автоматичної кластеризації колекції текстових документів зводиться до пошуку невідомої множини таким чином, щоб підсумкова множина була оптимальною у відповідності з деяким критерієм якості розбиття документів.

Слід зазначити, що вихідними даними задачі кластеризації є не самі тексти на природній мові, а їх інформаційно-пошукові образи.

Інформаційно-пошуковий образ документа представляє собою багатовимірний вектор в евклідовому просторі ознак документа, що характеризує смисловий зміст вихідного документа. Ознаки документів автоматично витягуються з текстів відповідно до обраного способу представлення тексту, і в самому поширеному випадку є словами. Ознаки документів всієї колекції об'єднуються в загальну множину. Вектор ознак кожного документа має розмірність N_p. Процес кластеризації базується на аналізі тематичної близькості документів, визначення якої полягає в припущенні, що геометрична близькість

векторів документів в просторі ознак документів всієї колекції означає дійсну подібність предметних областей даних документів [6].

Оцінка тематичної близькості документів заснована на обчисленні деякої міри близькості. Часто використовуваними мірами близькості між векторами текстових документів у просторі їх ознак є косинусна міра, яка обчислює значення косинуса між двома векторами документів і міри близькості, засновані на вимірюванні відстані між векторами документів в багатовимірному просторі ознак документів [3].

Вихідними даними задачі автоматичної кластеризації документів є отриманий в результаті її вирішення набір кластерів, структура яких залежить від вибору алгоритму кластеризації та може належати до одного з наступних типів [5]:

– плоский набір кластерів - множина кластерів, що відображає деяке число незалежних один від одного груп документів;

– ієрархічний набір кластерів - множина кластерів, елементам якої співставлені зв'язки ієрархічного типу, які відображають деревоподібну структуру груп документів, при якій кожен вузол дерева представляє групу, що містить всі документи її груп-нащадків;

– набір кластерів у вигляді графа - множина **кла**стерів, елементам якого співставлені зв'язки **довільного типу, що відображають деякі відносини між групами документів.**

Кластеризація масиву авторефератів дисертацій буде складатися з наступних основних етапів обробки даних:

– формування інформаційно-пошукових образів текстових документів;

– формування множини кластерів інформаційно-пошукових образів.

Розроблюваний алгоритм формування образів документів заснований на статистичному підході до аналізу текстів на природній мові. Образ кожного документа пропонується формувати у вигляді багатовимірного вектора нормалізованих і зважених одиночних слів (ознак), що зустрічаються в тексті даного документа. Розмірність такого вектора буде дорівнювати кількості унікальних ознак у колекції документів.

Запропонований спосіб формування образів $Ф_р$ складається з таких основних етапів: $Ф_р$ - спосіб вилучення ознак із текстів документів; $Ф_{ОР}$ - спосіб відображення документів у простір їх ознак; $Ф_к$ - алгоритм редукції простору ознак документів. Спосіб вилучення ознак $Ф_р$ полягає в послідовному виконанні наступних операцій: лексичний аналіз тексту (видалення розмітки, пунктуації, цифр, перетворення всіх букв у прописні тощо), видалення стапелів, тобто широковживаних слів, які не несуть самостійного сенсу, наприклад, прийменників, сполучників, часток і займенників; морфологічний аналіз.

Простежимо використання такого підходу до морфологічного аналізу, як виділення псевдооснов слів. В результаті даного аналізу слова з тексту приводяться до спеціального виду, і в подальшому, слова, що мають однаковий спеціальний вид (псевдооснову) розглядаються як одна ознака. В результаті вилучення ознак способом $Ф_P$ вдається отримати Np - розмірну множину ознак (псевдооснов слів) колекції документів P, яку також називають загальним словником ознак колекції документів.

Спосіб відображення документів у простір їх ознак $Ф_{OP}$ заснований на процедурі зважування ознак. Зважування ознак документів пропонується виконувати за допомогою традиційної техніки tf*idf, яка є незалежною від наявності навчальної множини, враховує частоту входження терму, як в окремий документ, так і в усю колекцію в цілому.

Необхідність розробки алгоритму редукції простору ознак документів обумовлена тим, що високорозмірні та розріджені вектори документів мають у просторі ознак недостатньо виразну орієнтацію для того, щоб автоматичні методи шляхом обчислення відстані між ними могли б зробити однозначний висновок про їхню спорідненість або відмінність. Для вирішення даної проблеми в сучасних інформаційно-пошукових системах застосовується примусове скорочення простору ознак за критерієм DF. Алгоритм примусової редукції за критерієм DF видаляє із загального словника ознак P колекції документів всі ті ознаки, документа частота яких вище порогового DF значення $т_{max}$ та нижче порогового значення $x_{m}j_{n}$.

Кластеризацію інформаційно-пошукових образів документів (авторефератів дисертацій) запропоновано виконувати у відповідності з підходом, заснованим на самоорганізуючих картах Кохонена [4]. Штучна нейронна мережа Кохонена або самоорганізуюча карта ознак (SOM) представляє собою двослойну мережу. Кожний нейрон першого (розподільного) слою з'єднаний з усіма нейронами другого (вихідного) слою, які розташовані у вигляді двовимірної решітки. Нейрони вихідного слою називаються кластерними елементами, їх кількість визначає максимальну кількість сегментів, на які система може розділити вихідні дані. Збільшуючи кількість нейронів другого слою можна збільшувати деталізацію результатів процесу кластеризації.

Система працює за принципом змагання - нейрони другого слою змагаються один з одним, перемагає той елемент-нейрон, чий вектор ваги ближче всього до вихідного вектора сигналів. За міру близькості двох векторів зазвичай береться евклідова відстань між ними. Таким чином, кожен вихідний вектор відноситься до деякого кластерного елементу.

Перш ніж мережа почне працювати її необхідно навчити на множині даних, яка буде піддана кластеризації. На кожному кроці навчання з вихідного набору даних

випадково вибирається один вектор. Потім проводиться пошук нейрона вихідного слою, для якого відстань між його вектором ваги і вихідним вектором – мінімальна. За певним правилом здійснюється корегування ваги для нейрона-переможця і нейронів з його околу, яка задається відповідною функцією околу h(t, j, m), де m - нейрон-переможець; j - нейрон вихідного слою, для якого обчислюється значення функції околу; t - параметр часу. Радіус околу повинен зменшуватися із збільшенням параметра часу. Цю проблему можна вирішити використанням функції Гаусса.

Кластером буде група векторів, відстань між якими всередині цієї групи менше, ніж відстань до сусідніх груп. Структура кластерів при використанні алгоритму самоорганізуючих карт Кохонена може бути відображена шляхом візуалізації відстані між опорними векторами (ваговими коефіцієнтами нейронів). При використанні цього методу найчастіше використовується уніфікована матриця відстаней, тобто обчислюється відстань між вектором ваги нейрона в сітці і його найближчими сусідами. Потім ці значення використовуються для визначення кольору, яким цей вузол буде відображатися. Зазвичай використовують градації сірого, причому, чим більше відстань, тим темніше відображається вузол. При такому використанні, вузлам з найбільшою відстанню між ними та сусідами відповідає чорний колір, а навколишнім вузлам - білий. Таким чином, розташовані поблизу кластери зі схожими кольорами утворюють більш глобальні кластери. Зазвичай в них розташовані близькі за ознаками документи.

На вхід нейронної мережі подається множина векторів документів у вигляді матриці розміром N на M, де N - кількість документів, що кластеризуються, а M - кількість унікальних термів у колекції документів, які кластеризуються. На перетині стовпчиків і рядків розташовуються ваги j-того терму в і-тому документі, обчислені за методом tf * idf.

Базовий алгоритм навчання мережі Кохонена виглядає наступним чином:

Крок 1. Ініціалізувати матрицю ваги малими випадковими значеннями (на відрізку [0, 1]).

Крок 2. Випадковим чином вибрати вектор з вихідної множини.

Крок 3. Для кожного вихідного нейрона j обчислити відстань між його вектором ваги Wi та вихідним вектором x.

Крок 4. Знайти вихідний нейрон-переможець j_{min} з мінімальною відстанню min(dj).

Крок 5. Для вихідного нейрона-переможця j_{min} та для його сусідів з околу оновлюються вектори ваги за правилом:

$$Wy(t+1)=Wy(t)+e(t)*h(t,j,m)*(x_j-Wy(t)),$$

де Wy(t) – значення вагового коефіцієнта зв'язку вхідного нейрона і та вихідного нейрона j

у момент часу t; h (t,j,m) - значення функції околу з центральним нейроном вихідного слою m для нейрона вихідного слою j у момент часу t; e(t) - коефіцієнт швидкості навчання в момент часу t; x_j - вихід нейрона першого слою з номером і.

Крок 6. Повторити кроки з кроку 2 для всіх елементів вихідної множини.

Цикл навчання триває до досягнення системою потрібного стану. В якості критеріїв зупинки процесу навчання можна використовувати наступні:

- топологічну впорядкованість карти ознак (матриці ваги);
- зміни ваги стають незначними;
- вихід мережі стабілізується, тобто вихідні вектори не переходять між кластерними елементами;
- досягнуто граничне значення помилки на карті;
- пройдено задану кількість кроків.

Представлений метод автоматичної кластеризації текстового масиву містить наступні кроки:

- формування інформаційно-пошукових образів документів, що дозволяє будувати образи текстових документів у вигляді векторів у просторі їх ознак;
- редукцію простору ознак документів, яка дозволяє підвищити представницьку здатність сформованих на попередньому кроці образів документів;
- алгоритм кластеризації заснований на самоорганізуючих картах Кохонена.

2.2 Алгоритм побудови лексичних ланцюжків

Лексичні ланцюжки представляють змістовні єдності серед довільного числа зв’язаних слів.

По-перше, це лексичний ланцюжок тотожних об’єктів. До нього заносяться ті елементи тексту, що вказують на один і той самий об’єкт або одне і те саме поняття. Задачу знаходження всіх згадувань одного і того самого об’єкту називають задачею розв’язання кореференції [12].

По-друге, це лексичний ланцюжок семантично зв'язаних об’єктів. Такий ланцюжок не обмежує типи елементів та види зв’язків між ними, поки вони пов’язані між собою з точки зору автора. Надалі вживається саме таке значення терміну, оскільки це дозволяє визначати не лише тотожність/відмінність між об’єктами, а і визначати тематичну належність. Для побудови таких лексичних ланцюжків в якості джерела знань вживається WordNet.

Лексичні ланцюжки обчислюються шляхом групування послідовних наборів семантично зв'язаних слів. Тотожні слова, синоніми, гіперніми і гіпоніми, мероніми, голоніми - ознаки, що дозволяють групувати слова в один ланцюжок. Гіпернім - поняття, що є узагальнюючим для даного у онтології (WordNet[8]). Тіпонім - поняття, що є уточненням даного у онтології (WordNet). Меронім - поняття, що позначає «ціле» у відношенні «ціле-частина». Голонім - поняття, що позначає «частина» у відношенні «ціле-частина».

Необхідно зауважити, що у WordNet представлені не слова, а сенси (концепти) - поняття, і кожен сенс має свій комплект слів, які його позначають („синсет")

Умови групування:

1. Два входження повнозначного слова ідентичні і використовуються в тому ж самому сенсі. *(Великий корабель на рейді. Цей корабель - вітрильник.)*

2. Два входження повнозначних слів використовуються в одному і тому ж самому сенсі, але текстуально різні тобто, є синонімами. *(Той аероплан летить швидко. Проте, мій літак швидше.)*

3. Змісти двох входжень повнозначних слів мають гіпернім/гіпонім відношення між ними. *(Я маю автомобіль. Це - вантажівка.)*

4. Змісти двох входжень повнозначних слів - елементи одного рівня в гіпернім/гіпонім дереві і мають спільного предка. (*Той аеробус летить швидко. Проте, мій винищувач швидше.)*

5. Два входження повнозначних слів означають відповідно частину та ціле – є зв'язок меронімії. *(Дмитро відчинив двері. Замок голосно клацнув*.)

В обчисленні лексичних ланцюжків входження повнозначних слів повинні бути згруповані згідно з вищезгаданими правилами, але кожне входження повнозначного слова повинно належати точно одному лексичному ланцюжку.

Розглянемо загальний метод побудови колекції лексичних ланцюжків для тексту:

1. Вибрати слово або словосполучення, яке взято з тексту (надалі об'єкт) і яке має представлення у WordNet.

2. Для кожного об'єкту знайти відповідний ланцюжок і вставити об'єкт туди.

3. Якщо ланцюжок не існує, то створити новий на основі заданого об'єкту.

Як видно з опису методу, немає однозначного способу для відповіді на певні питання. Як визначити відповідний ланцюжок? Наприклад, входження іменника може відповідати декільком різним сенсам слова, і система повинна визначити, яке саме входження має місце. Наприклад, «коса» як інструмент і «коса» як зачіска.

Як забезпечити однозначність? Навіть якщо сенс слова може бути визначений, може трапитись, що слово можна занести до декількох різних лексичних ланцюжків, тому що це слово може бути зв’язане зі словами в різних ланцюжках.

Для коректності вводяться параметри об’єднання об’єкту з ланцюжком. До цих параметрів входять відстані та напрямки для зазначених у попередньому розділі умов групування за зв’язками. Необхідно враховувати, що комбінуючи умови групування, можна будувати як завгодно складні зв’язки.

1. Для умови групування 3 накладається обмеження на довжину такого зв’язку при обчисленні його від більш деталізованого поняття до більш загального, але не навпаки.
2. Для умови групування 4 накладається обмеження на відстань від узагальнюючого поняття. У нашому випадку – 2.
3. Для умови групування 5 накладається обмеження на довжину такого зв’язку, якщо в шляху є різнотипні переходи. У нашому випадку - 2.

Необхідно зауважити, що два слова будуть зв’язані разом з більшою вірогідністю, якщо в тексті, який аналізується, вони стоять поблизу. З цього випливає ще один параметр об’єднання: об’єкти зв’язані, якщо вони знаходяться у сусідніх реченнях.

Для різних умов групування варто задати різні допустимі відстані для визначення сусідства:

1. Для умов 1-3 відстань може сягати 7 речень.
2. Для умов 4-5 відстань не повина перевищувати 4 речення.

Для ефективного обчислення лексичних ланцюжків створюється структура, яка неявно зберігає кожну інтерпретацію кожного слова. А потім з цього неявного представлення обчислюється оптимальна конфігурація. Обробка документа починається зі створення великого масиву мета- ланцюжків, розмір якого дорівнює числу сенсів слів тексту, знайдених у WordNet, плюс число слів у документі, оскільки можливо, що слова не будуть знайдені у WordNet. Довжина кожного такого мета-ланцюжка дорівнює кількості повнозначних слів у тексті.

Коли алгоритм знаходить повнозначне слово, збільшується лічильник у відповідному мета-ланцюжку, який містить сенс цього слова, та у кожному ланцюжку, куди це слово входить за однією з вище визначених ознак.

Коли переший прохід закінчено, текст проглядається ще раз, і для кожного повнозначного слова визначається ланцюжок, до якого воно вносить якнайбільшу вагу. З решти ланцюжків слово вилучається.

Позначення: w_f - повнозначне слово з тексту, *cfwjj* - сенс слова w, визначений за WordNet, *{c(Wi)j}k* - мета-ланцюжок.

Алгоритм побудови лексичних ланцюжків.

Для кожного ***Wj є T***

Для кожного *c(Wj)j*

Для кожного *{c(Wj)j}k* обновити значення

Для кожного *W i є T*

Для кожного $\{c(wi)_m\}k$, що має *c(w,)j*

Визначити *k* мета-ланцюжа, до якого *c(w,)*; належить найбільше.

Обновити таблицю, видаливши зайві елементи.

Оцінка складності роботи алгоритму:

$$0\ (N*s_{max}*M^2) \qquad (2.1)$$

де *N* – довжина тексту в словах, s_{max} - найбільша кількість сенсів слова, *M* – довжина мета-ланцюжка. Таким чином, отримані ланцюжки представляють собою динамічно сформований комплект тем документу.

За відсутності даних про конкретне слово чи поняття у локалізації WordNet можна застосувати евристичне об'єднання елементів тексту у тематичні ланцюжки. Таке об'єднання спирається на припущення, що автор тексту писав його як осмислений текст, і користувався однією термінологією у межах викладення однієї думки або ідеї.

Алгоритм TextTilling розділяє текст на сукупність фрагментів, які мають внутрішні зв'язки. Тоді два об'єкти будуть зв'язані з більшою достовірністю, якщо вони знаходяться в одному фрагменті.

Відстань між вікнами:

$$\cos © = (A,B)/(\ |A||B|) \qquad (2.2)$$

де *A,B* - частотні вектори відповідних вікон.

Межа розриву – параметр *Th.*

Алгоритм TextTilling

Задати розмір *n* вікна в словах – w_1 ,...w_n

Пересуваючи вікно по тексту, створити з $c(w_i)_j$ вектори, що відповідають двом сусіднім вікнам;

Порахувати відстані між кожними двома сусідніми вікнами

$w_i....w_{n+i}$; $w_{n+i+1}....w_{i+2n}$;

Якщо cos © > *Th* – розрив (2.3)

Встановити маркер «Розрив»

Оцінка складності роботи алгоритму: $O(N*w^2)$, де N - довжина тексту в словах, w -

довжина вікна у словах.

Результати індексації, а саме побудовані лексичні ланцюжки, необхідні для двох задач:

– побудови оцінки важливості тих чи інших елементів у рефераті;

– побудови оцінки близькості фрагментів тексту з метою вилучення дублів.

Ані частотний підхід, ані зважування окремих елементів семантичного представлення (концептів) не працює коректно, якщо в тексті(текстах), що реферуються, спостерігається велика кількість повторів або запозичень.

Також треба враховувати, що задача оцінювання імовірних запозичень має високу обчислювальну складність, якщо її виконувати на всьому наборі вхідних текстів. Можливість розбиття тексту на фрагменти та вилучення очевидних дублів значно прискорює роботу системи реферування.

2.3 Методи кластеризації на зважених графах

Важливим етапом мультиреферування є кластеризація фрагментів текстів, для того щоб прибрати дублі та зібрати подібні тексти в групи. Якщо замість набору текстів є огляд, то зникає потреба в першому етапі кластеризації, а саме в кластеризації за результатами індексації.

Важливою проблемою кластерного аналізу є вибір методології визначення якості одержаних кластерів та визначення цільової функції кластерного аналізу. Найбільш ефективні результати досліджень одержані при використанні так званих зовнішніх мір, тобто при порівнянні результатів автоматичного аналізу з ручним.

Критерії якості кластеризації, як правило, ґрунтуються на наступних вимогах:

– усередині груп об’єкти повинні бути тісно пов’язані між собою;

– значення відстані між об’єктами різних груп повинне бути достатньо великим;

– при інших рівних умовах розподіл об’єктів по групам повинен бути рівномірним.

Міри якості кластерного розбиття поділяються на два класи. До першого класу відносяться міри, які базуються на оцінках експертів, так звані зовнішні міри якості, наприклад *F*-міра. До другого класу, відповідно, відносяться міри, які не використовують додаткову інформацію - внутрішні міри якості(загальна внутрішня подібність).

У задачі мультиреферування порушуються всі такі вимоги. У той же час існує

хороша оцінка загальної внутрішньої подібності за рахунок можливості обчислити ступінь запозичення одного тексту відносно іншого. Це, у свою чергу, підштовхує до застосування кластеризації на графах.

Міра близькості, заснована на визначенні запозичень, добре спрацьовує, коли тексти документів є запозиченнями одне з одного. Це, зазвичай є нормою в текстах новин, опублікованих в Інтернеті, коли новину передруковують, лише додавши посилання на джерело. Проте для повноцінного аналізу такої міри близькості не досить. Альтернативою виступає метод оцінки за словниками текстів. Два тексти будуть тематично близькими, якщо набори сутностей, згаданих у них, збігаються. Що більша збіжність, то більша подібність тематичного наповнення.

Для оцінювання такої близькості використовується косинусна міра.

$$\cos © = (A,B)/(|A||B|) \quad (2.4)$$

де A,B - частотні словники текстів, представлені як вектори.

Подібність, обчислена таким чином, досить добре описує тематичне наповнення текстів, за умови, що коректно виділені основні термінологічні одиниці.

Оцінка складності швидкість порівняння текстів за такою мірою:

$$O \text{ (оцінки подібності)} = O(N) \quad (2.5)$$

Методи кластеризації, які базуються на використанні зважених графів, об’єктам вибірки ставлять у відповідність деякий набір вершин V. Дві вершини v_a та v_b, що відповідають векторам ознак $\boldsymbol{x_a}$ та x_b об'єктів A та B, можуть бути з’єднані ненаправленим ребром E_{ab} з додатковою вагою $S(x_a;x_b)$, що являє собою міру близькості між векторами. Кількість ребер графа $|E|$ дорівнює кількості ненульових значень близькості між усіма парами точок. Набір ребер, видалення яких розбиває граф $G = (V,E)$ на $к$ підграфів, що не перетинаються, називається роздільником ребер. Отже, метод кластерного аналізу з використанням зважених графів полягає в знаходженні роздільника ребер з мінімальною сумою належних йому ребер. При цьому часто накладається додаткова умова - отримання приблизно рівної кількості об’єктів (вузлів) у кожному кластері (підграфі).

Відповідно до розглянутого методу загальна складність алгоритму кластеризації складе:

$$O (N^2)* O \text{ (оцінки запозичень)} \quad (2.6)$$

Треба врахувати, що сама по собі формула вказує на складність принаймні

O(оцінки запозичень) = $O(N^4)$, де N - кількість слів у тексті.

Таким чином, оптимальною виглядає кластеризація за результатами індексації або якимось іншим способом з подальшою додатковою кластеризацією за мірою запозичення.

2.4 Семантико-синтаксичний та семантичний алгоритм списку

Цей алгоритм належить до групи алгоритмів, які виконують абстрагування, з опорою на зовнішні джерела інформації. У наведеному вигляді він не здатний прореферувати весь текст, або стиснути його до малого об'єму. Проте він є корисним, оскільки дозволяє отримати для ряду випадків стиск там, де простий вибір буде змушений втратити інформацію.

Передбачається, що є ряд додаткових механізмів, а саме: синтаксичний та семантичні аналізатори, синтаксичний синтезатор для речень. У межах областей між двома маркерами «Тема почалася» і «Тема скінчилася», які належать одній темі, застосовується перший алгоритм узагальнення. Він працює переважно з онтологією, використовуючи зв'язок „бути" (is_a). У процесі узагальнення цей алгоритм пробігає по онтології від понять нижчого рівня до понять вищого рівня у пошуках поняття, яке є водночас допустимими та досить абстрактними, для можливості здійснення узагальнення. Для нього є обов'язковою синтаксична передобробка, оскільки він інтенсивно використовує синтаксичні дані.

Позначимо тематичну область як *To.*

Алгоритм стиску 1 для кожного s_i є To

Скласти предикатну структуру відповідно до підмета і присудка.

Назвати її базовою.

Для кожного $s_{i+j} \epsilon$ *To* (від даного і до кінця області) скласти предикатну структуру відповідно до підмета і присудка.

Порівняти предикатну структуру з базовою.

Якщо для підметів присудків або і підметів, і присудків виконуються «Умови групування» з розділу Індексація на відстань 2

З s_i та s_{i+j} будується одне s_i, більш поширене і, використовуючи більш загальні поняття, s_{i+j} вилучається.

Оцінка складності роботи алгоритму:

$$0(N_T^2 * l_{max} * \kappa_{max}) \qquad (2.7)$$

де N_T- довжина блоку тексту в реченнях, κ_{max} - найбільша кількість сенсів слова, l_{max} - найбільша довжина речення в тексті.

Недоліком цього алгоритму є його чутливість до синтаксичних неоднорідностей та «короткозорість», оскільки він не реагує на зв'язки між поняттями у WordNet, що мають

довжину більшу за 2. Проте, якщо збільшити відстань до 3х або 4х, часто відбувається надлишкове узагальнення, що негативно впливає на якість реферату. Попри очевидні переваги такого алгоритму, а саме здатність до складання висновків, хоч би і обмежену, необхідно зауважити його малу частоту очікуваного використання.

Семантичний алгоритм списку належить до групи алгоритмів, які виконують абстрагування, з опорою на зовнішні джерела інформації. У наведеному вигляді він не здатний прореферувати весь текст, або стиснути його до малого об'єму. Проте він є корисним, оскільки дозволяє отримати для ряду випадків стиск там, де простий вибір буде змушений втратити інформацію.

Передбачається, що є ряд додаткових механізмів, а саме: синтаксичний та семантичні аналізатори, синтаксичний синтезатор для речень.

У межах областей між двома маркерами «Тема почалася» і «Тема скінчилася», які належать одній темі, також застосовується другий алгоритм узагальнення. Його основою є пошук у ширину в орієнтованому графі онтології. Умови зупинки:

1. Як тільки зустрічається вершина (концепт), що є забороненою (зайвою), алгоритм припиняє обчислювати ваги для цієї вершини та всіх вершин, для яких вона є нащадком в ієрархії WordNet. До заборонених сенсів належать загальні поняття, якщо вони не представлені явно в лексичному ланцюжку.
2. Якщо вершина поза бажаною тематикою, алгоритм припиняє обчислювати ваги для цієї вершини та всіх вершин, для яких вона є нащадком в ієрархії WordNet;
3. Якщо досягнуто довжину шляху, рівну 5.

Позначимо тематичну область як *To.* Повнозначне слово – w_i, $c(w_i)_j$ - сенс слова.

Алгоритм узагальнення

Створити список *L*

Для кожного $s_i \in To$

Для кожного $\boldsymbol{w \in s_i}$

Для кожного $c(w_i)_j$

Додати $c(w_i)_j$ у *L*

///

Для кожної ітерації

Створити список L_n

Якщо не виконуються умови зупинки

Для $c(w) \in L$

Визначити кількість шляхів, що проходять через $c(w)$ в напрямку більш загального $c(w)_n$ у WordNet

Занести $c(w)_n$ у L_n

Встановити $c(w)_n$ вагу, рівну сумі всіх шляхів від нижніх $c(w)$ синсетів через нього $c(w)_n$

Замінити $L = L_n$

///

Для кожного $c(w) \in L$

Для кожного $s_i \in To$

Для кожного $\boldsymbol{w_i \in s_i}$

Для кожного $c(w_i)_j$

Виконати розмітку за WordNet.

///

Для кожного $s_i \in To$

Створити список L (речень кандидатів)

Для кожного $\boldsymbol{s_{i+j} \in To}$

Порівняти маркери s_i та s_{i+j},

Якщо маркери співпадають,

Додати s_{i+j} у L

Інакше:

Для кожного $s_k \in L$ (речення зі списку)

Порівняти предикатну структуру s_i зі s_k.

Якщо немає відповідностей – вилучити s_k

Опрацювати L, створивши більш загальне речення s_i.

Вилучити використані s_k з тектсу.

Зробити список порожнім.

Оцінка складності роботи алгоритму на одному проході:

$$0(N_T^2 * l_{max} * \kappa_{max} + SS^3 + SS * N_T * 1_{max} * \kappa_{max} + N_T^2 * 1_{max} * \kappa_{max}) \quad (2.8)$$

де N_T - довжина блоку тексту в реченнях, κ_{max} - найбільша кількість сенсів слова, 1_{max}- найбільша довжина речення в тексті, SS - кількість сенсів у списку сенсів. Очевидно, що чим більше різних змістовно наповнених елементів є в межах теми, тим повільніше працює алгоритм.

Таким чином, можна визначити ті концепти з WordNet, що не представлені в тексті явно, проте сильно пов’язані з його змістом. Це дозволяє, наприклад, узагальнити «стіл, стілець, ліжко» до «меблі» але не до «об’єкт». Проблема полягає в тому, що як і попередній алгоритм, цей також чутливий до синтаксичних структур. Так само необхідно зауважити малу ефективність (частоту очікуваного вживання) даного алгоритму.

3 ВИБІР, РОЗРОБКА І ТЕСТУВАННЯ ЕФЕКТИВНОСТІ МЕТОДІВ. АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Для виконання порівняльного аналізу алгоритмів була розроблена програма, у якій реалізовані деякі методи, описані у першому розділі. Мовою програмування було обрано Java, оскільки саме для неї розроблено багато бібліотек, що полегшують створення даної системи. Код наведений у Додатку А.

3.1 Видалення стоп слів

Першим набором алгоритмів, які було вирішено порівняти є алгоритми пошуку стоп-слів, що є невід'ємною частиною попередньої обробки текстів. Для порівняння обрано алгоритм, побудований на Y-інтерпретації закону Бредфорда, причому в двох варіаціях: з попереднім стемінгом та виконанням стемінгу після видалення стоп-слів. Другий алгоритм - словниковий. Словник представлений на сайті http://www.ranks.nl/stopwords. Третій - на основі об'єднання слів, що зустрічаються найчастіше у текстах набору.

3.1.1 За законом Бредфорда без стемінгу

Слід зазначити, що у даній роботі використовується стемер Портера, що працює на принципі N-грам, тому результати можуть сильно різнитись від порядку застосування алгоритмів, що і буде продемонстровано нижче.

Для демонстрації результатів було проведено аналіз на прикладах з 4 тестів наукової тематики. Як показало дослідження, для отримання достовірних даних достатньо 4 текстів. Результати наведено в таблицях 3.1 та 3.2.

Біля назви тексту в дужках наведено загальну кількість слів у тексті. Для кожного тексту розраховано кількість та частоту входжень слова у текст. Після цього вручну було перевірено точність методу. В таблиці виділено слова, які алгоритм відмітив, як стоп-слова, проте вони несуть інформативне навантаження у тексті.

Таблиця 3.1 – Результати виявлення стоп-слів за законом Бредфорда без стемінгу, частина 1

Machine Learning (1784)			Sentiment Analysys (1725)		
Word	Amount	Percentage	Word	Amount	Percentage
the	92	5.156951	the	99	5.73913
learning	60	3.363229	of	72	4.173913
of	60	3.363229	a	59	3.42029
a	55	3.08296	sentiment	47	2.724638
and	53	2.970852	to	45	2.608696
to	51	2.858744	and	44	2.550725
in	42	2.35426	is	33	1.913043
is	35	1.961883	anal y si	29	1.681159
machine	29	1.625561	in	27	1.565217
data	22	1.233184	or	25	1.449275
from	20	1.121076	that	21	1.217391
as	20	1.121076	on	20	1.15942
with	18	1.008969	as	19	1.101449
on	17	0.952915	text	17	0.985507
are	17	0.952915	it	17	0.985507
that	16	0.896861	this	16	0.927536
by	11	0.616592	can	15	0.869565
it	11	0.616592	be	14	0.811594
be	11	0.616592	for	13	0.753623
training	11	0.616592	featur	12	0.695652
theory	10	0.560538	classifi	11	0.637681
Field	10	0.560538	are	11	0.637681
can	10	0.560538			
Всього	677	37.94843		666	38.6087
Всього після перевірки	557	31.22222		557	32.2889

Таблиця 3.2 – Результати виявлення стоп-слів за законом Бредфорда без стемінгу, частина 2

Text Mining (1291)			Big Data (1721)		
Word	Amount	Percentage	Word	Amount	Percentage
data	82	6.351665	data	88	5.113306
the	74	5.731991	and	82	4.764672
and	46	3.563129	the	74	4.299826
of	37	2.865995	of	62	3.602557

Продовження таблиці 3.2

Text Mining (1291)			Big Data (1721)		
mining	32	2.478699	to	43	2.498547
in	32	2.478699	big	38	2.208019
to	29	2.246321	a	37	2.149913
is	22	1.704105	in	31	1.801278
a	19	1.471727	is	21	1.220221
as	17	1.316809	that	16	0.929692
for	16	1.239349	for	14	0.813481
be	14	1.084431	are	14	0.813481
or	13	1.006971	as	14	0.813481
learning	12	0.929512	with	13	0.755375
analysis	12	0.929512	information	12	0.697269
process	11	0.852053	can	12	0.697269
patterns	11	0.852053	or	11	0.639163
used	11	0.852053	on	11	0.639163
knowledge	10	0.774593	it	11	0.639163
discovery	10	0.774593	storage	10	0.581058
			analytics	10	0.581058
			parallel	10	0.581058
			processing	9	0.522952
			from	9	0.522952
Всього	510	39.50426		652	37.88495
Всього після перевірки	319	24.70952		513	29.80825

3.1.2 За законом Бредфорда зі стемінгом

Аналогічно попередньому розділу проведемо аналіз для тих самих текстів, але з попереднім стемінгом.

Таблиця 3.3 – Результати виявлення стоп-слів за законом Бредфорда зі стемінгом, частина 1

Machine Learning (1784)			Sentiment Analysys (1725)		
Word	Amount	Percentage	Word	Amount	Percentage
the	92	5.156951	the	99	5.73913
learning	60	3.363229	of	72	4.173913
of	60	3.363229	a	59	3.42029
a	55	3.08296	sentiment	47	2.724638
and	53	2.970852	to	45	2.608696

Продовження таблиці 3.3

Machine Learning (1784)			Sentiment Analysys (1725)		
to	51	2.858744	and	44	2.550725
in	42	2.35426	is	33	1.913043
is	35	1.961883	analysi	29	1.681159
machine	29	1.625561	in	27	1.565217
data	22	1.233184	or	25	1.449275
from	20	1.121076	that	21	1.217391
as	20	1.121076	on	20	1.15942
with	18	1.008969	as	19	1.101449
on	17	0.952915	text	17	0.985507
are	17	0.952915	it	17	0.985507
that	16	0.896861	this	16	0.927536
by	11	0.616592	can	15	0.869565
it	11	0.616592	be	14	0.811594
be	11	0.616592	for	13	0.753623
training	11	0.616592	featur	12	0.695652
theory	10	0.560538	classifi	11	0.637681
Field	10	0.560538	are	11	0.637681
can	10	0.560538			
Всього	677	37.94843		666	38.6087
Всього після перевірки	557	31.22222		557	32.2889

Таблиця 3.4 – Результати виявлення стоп-слів за законом Бредфорда зі стемінгом, частина 1

Text mining (1291)			Big Data (1721)		
Word	Amount	Percentage	Word	Amount	Percentage
data	82	6.351665	data	88	5.113306
the	74	5.731991	and	82	4.764672
and	46	3.563129	the	74	4.299826
of	37	2.865995	of	62	3.602557
mine	33	2.556158	to	43	2.498547
in	32	2.478699	big	38	2.208019
to	29	2.246321	a	37	2.149913
is	22	1.704105	in	31	1.801278
a	19	1.471727	is	21	1.220221
use	18	1.394268	process	18	1.045904
as	17	1.316809	that	16	0.929692
for	16	1.239349	it	16	0.929692
be	14	1.084431	for	14	0.813481
process	13	1.006971	are	14	0.813481

Продовження таблиці 3.4

Text mining (1291)			Big Data (1721)		
or	13	1.006971	use	14	0.813481
set	12	0.929512	as	14	0.813481
learn	12	0.929512	with	13	0.755375
databas	12	0.929512	can	12	0.697269
			set	11	0.639163
			or	11	0.639163
			inform	11	0.639163
Всього	501	38.80713		651	37.82684
Після перевір.	349	27.03333		507	29.45969

3.1.3 Результати

Зведемо результати двох попередніх підрозділів у таблиці та порівняємо з результатами комерційного продукту advego.ru/

Таблиця 3.5 – Результати для Y-інтерпретації закону Бредфорда без стемінгу

Назва тексту	Всього слів	advego		Бредфорд		Перевірка		Похибка відносно перевірки		Похибка відносно advego	
		Всього	Процент	Всього	Процент	Всього	Процент	Всього	Процент	Всього	Процент
Text mining	1291	377	29,20	510	39.50	319	24.70	191	37.45	133	35.27
Big Data	1721	608	35,32	652	37.88	513	29.80	139	21.31	44	7.23
Machine learning	1784	644	36,09	677	37.94	557	31.22	120	17.72	33	5.12
Sentiment Analysys	1725	608	35,24	666	38.60	557	32.28	109	16.36	58	9.54

Таблиця 3.6 – Результати для Y-інтерпретації закону Бредфорда зі стемінгом

Назва тексту	Всього слів	advego		Бредфорд		Перевірка		Похибка відносно перевірки		Похибка відносно advego	
		Всього	Процент	Всього	Процент	Всього	Процент	Всього	Процент	Всього	Процент
Text mining	1291	377	29,20	501	38.80	349	27.03	152	30.33	124	32.8912
Big Data	1721	608	35,32	651	37.82	507	29.45	144	22.11	43	7.07
Machine learning	1784	644	36,09	677	37.94	557	31.22	120	17.72	33	5.12
Sentiment Analysys	1725	608	35,24	666	38.60	557	32.28	109	16.36	58	9.53

Складемо порівняльну характеристику з урахуванням не лише точності, а й повноти виділення стоп-слів.

Таблиця 3.7 – Зведені результати для Y-інтерпретації закону Бредфорда

Назва тексту	Всього слів	Без стемінгу				Зі стемінгом			
		Всього	Процент	Точність	Повнота	Всього	Процент	Точність	Повнота
Text mining	1291	510	39.50	64.73	81.82%	501	38.80	67.1088	91.98%
Big Data	1721	652	37.88	92.77	81.48%	651	37.82	92.93	80.08%
Machine learning	1784	677	37.94	94.88	84.38%	677	37.94	94.88	84.38%
Sentiment Analysys	1725	666	38.60	90.46	90.84%	666	38.60	90.47	90.84%

Як видно з таблиці, метод після стемінгу дає дещо кращі результати. Також можна помітити залежність між кількістю слів та ефективністю роботи алгоритму.

3.1.4 Словниковий метод

Як вже було зазначено раніше, стоп-слова задані заздалегідь вручну. В даному випадку їх список взятий із сайту http://www.ranks.nl/stopwords, цей перелік налічує 665 слів.

Було проаналізовано 15 текстів наукової тематики. Результати наведено у таблиці нижче.

Таблиця 3.8 – Результати для словникового методу

Назва тексту	Загальна кількість слів	Кількість видалених стоп-слів		Advego	
		Кількість	Відсоток	Кількість	Відсоток
Getting Started in Text Mining Part Two	1505	826	54,88%	501	33,29%
Text Mining Introductory Overview	1709	964	56,41%	576	33,70%
Machine Learning 2	1784	961	53,87%	494	27,69%
Big Data	1721	776	45,09%	608	35,33%
Machine Learning	1784	961	53,87%	644	36,10%
How Companies Can Use Sentiment Analysis to Improve their business	1693	970	57,29%	622	36,74%
Getting Started in Text Mining	1891	1074	56,80%	691	36,54%
Sentiment Analysys	1725	906	52,52%	608	35,25%
Sentiment analysis using product review data	3868	1942	50,21%	1409	36,43%
Where to start with text mining	3102	1740	56,09%	1203	38,78%
Text mining is a burgeoning new field that attempt	1677	909	54,20%	574	34,23%
Text Mining	1291	605	46,86%	377	29,20%
Data Mining What is data mining	1636	739	45,17%	457	27,93%
Text categorization	1632	908	55,64%	560	34,31%
Sentiment Analysis of news comments	917	503	54,85%	314	34,24%

З таблиці видно, що словниковий метод видаляє набагато більше слів, це зумовлено словником, що підходить для багатьох типів і тематик текстів, а також для SEO.

3.1.5 Метод об’єднання

Як вже було описано вище, даний метод полягає у знаходженні та об’єднанні слів, що зустрічаються найчастіше у текстах із набору.

Розглянемо підхід до вибору порогу входження слова до переліку стоп-слів з інтерпретацією закону Бредфорда.

Із набору текстів була виділена зона J0 (зона стоп-слів). Ці слова наведені у таблиці нижче.

Таблиця 3.9 – Результати для словникового методу

Word	Amount	Percentage
the	1331	4.775402
of	953	3.419202
to	746	2.676521
and	719	2.57965
a	681	2.443312
in	504	1.808266
is	496	1.779564
data	393	1.410017
that	347	1.244977
for	336	1.205511
are	259	0.929248
as	254	0.911309
be	245	0.879018
it	233	0.835964
use	226	0.81085
or	224	0.803674
on	201	0.721154
Sentiment	192	0.688863
text	189	0.6781
with	186	0.667336
mine	186	0.667336
Learn	185	0.663749
can	175	0.62787
this	174	0.624282
word	138	0.495121
from	135	0.484357
by	132	0.473594

Продовження таблиці 3.9

Word	Amount	Percentage
analysi	122	0.437715
machin	112	0.401837
document	109	0.391073
process	106	0.38031
an	101	0.362371
have	100	0.358783

Всього помилкових слів - 1594 із 10589 слів у словнику, тобто похибка 15.053%, а точність 84.947%. З одного боку, за числами це дуже непогана точність, а з іншого на цьому прикладі видно, що цей метод видалить основні ключові слова, які характеризують цей набір тестів.

Цей метод може показати кращі результати, якщо взяти більше текстів абсолютно різної направленості, але така обробка за часом підготовки словника програє ручній підготовці словника.

3.1.6 Приклад тексту із проведеної попередньою обробкою

Маємо фрагмент тексту про Data Mining:

Data mining is an interdisciplinary subfield of computer science. It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Aside from the raw analysis step, it involves database and data management aspects, data preprocessing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating. Data mining is the analysis step of the "knowledge discovery in databases" process, or KDD.

The term is a misnomer, because the goal is the extraction of patterns and knowledge from large amounts of data, not the extraction (mining) of data itself. It also is a buzzword and is frequently applied to any form of large-scale data or information processing (collection, extraction, warehousing, analysis, and statistics) as well as any application of computer decision support system, including artificial intelligence, machine learning, and business intelligence.

The book Data mining: Practical machine learning tools and techniques with Java (which covers mostly machine learning material) was originally to be named just Practical machine learning, and the term data mining was only added for marketing reasons. Often the more general terms (large scale) data analysis and analytics - or, when referring to actual methods, artificial intelligence and machine learning - are more appropriate.

Після стемінгу та видалення стоп-слів зі словнику:

data mine interdisciplinari subfield comput scienc . comput process discov pattern data involv method intersect artifici intellig machin learn statist databas system . goal data mine process extract data set transform understand structur. raw analysi step involv databas data manag aspect data pre-process model infer consider interesting metric complex consider post-process discov structur visual onlin updat. data mine analysi knowledg discoveri databas process KDD.

term misnom goal extract pattern knowledg larg amount data extract mine data. buzzword frequent appli form large-seal data process collect extract wareh analysi statist well applic comput decis system includ artifici intellig machin learn busi intellig . book data mine practic machin learn tool techniqu Java cover machin learn materi origin practic machin learn term data mine market reason . term larg scale data analysi analyt - refer actual method artifici intellig machin learn - appropri.

Після стемінгу та видалення стоп-слів за Бредфордом:

interdisciplinari subfield comput scienc . comput discov pattern involv method intersect artifici intellig machin learn statist system . goal mine process extract data set transform understand structur . raw analysi step involv manag aspect data premodel infer consider interesting metric complex consider post- discov structur visual onlin updat. analysi knowledg discoveri KDD .

term misnom goal extract pattern knowledg larg amount data extract, buzzword frequent appli form large-seal data collect extract wareh analysi statist well applic comput decis system includ artifici intellig machin busi intellig . book practic machin learn tool techniqu Java cover machin learn materi origin practic machin term market reason . term larg scale analysi analyt - refer actual method artifici intellig machin- appropri.

Після стемінгу та видалення стоп-слів за методом об’єднання:

interdisciplinari subfield comput scienc . comput discov pattern involv method intersect artifici intellig machin statist databas system . goal extract transform understand structur. raw step involv databas manag aspect pre- model infer consider interesting metric complex consider post- discov structur visual onlin updat. analysi knowledg discoveri KDD.

term misnom goal extract pattern knowledg larg amount extract, buzzword frequent appli form large-seal collect extract wareh statist well applic comput decis system includ artifici intellig machin bust intellig . book mine practic machin tool techniqu Java cover machin learn materi origin practic machin term data mine market reason . term larg scale- refer actual method artifici intellig machin- appropri.

Розглянувши результати очевидно, що найкраще себе показує саме словниковий метод. Так як обидва інші видаляють основні ключові слова. В даному прикладі такі як: data, mining, process, learning, set, database.

Ці слова якнайкраще характеризують текст, є ключовими, а за даним методом вони мають бути виключеними з аналізу на етапі попередньої обробки. Тому можна зробити висновок, що дані методи не варто використовувати у системі.

3.2 Виділення ключових слів

Продемонструємо роботу алгоритмів виділення ключових слів. Для аналізу взято 3 алгоритми: 2 статистичних та один гібридний.

3.2.1 Міра TF-IDF

Наведемо приклади виділення ключових слів за мірою TF-IDF у таблиці 3.10.

Таблиця 3.10 – Результати для міри TF-IDF

Big Data		Text Mining		Text categorization		Text Mining inductor overview	
Word	TF-IDF	Word	TF-IDF	Word	TF-IDF	Word	TF-IDF
datum	61	datum	15	text	12	word	12
information	10	Data	12	document	11	text	11
set	8	mining	11	categorization	10	document	10
analytic	7	process	7	category	6	example	10
processing	7	pattern	6	word	6	mining	10
storage	7	analysis	5	language	5	term	8
architecture	6	learning	5	method	5	application	6
system	6	pre	5	technique	5	input	6

Продовження таблиці 3.10

Big Data		Text Mining		Text categorization		Text Mining inductor overview	
Word	TF-IDF	Word	TF-IDF	Word	TF-IDF	Word	TF-IDF
technology	6	database	4	authorship	4	number	6
application	5	machine	4	classification	4	algorithm	4
framework	5	method	4	example	4	datum	4
process	5	set	4	machine	4	domain	4
size	5	task	4	model	4	indexing	4
velocity	5	analytic	3	retrieval	4	information	4
volume	5	business	3	scheme	4	type	4
characteristic	4			set	4	approach	3
database	4			test	4	cluster	3
definition	4			topic	4	form	3
insight	4			training	4	language	3
management	4			application	3	list	3
model	4			approach	3	method	3
organization	4			author	3	project	3
server	4			datum	3	service	3
time	4			learning	3	time	3
type	4			metada	3	variable	3
value	4			mining	3		
3v	3			n-gram	3		
ability	3			news	3		
algorithm	3						
analysis	3						
business	3						
component	3						
computing	3						
concept	3						
connection	3						
cost	3						
difficulty	3						
factory	3						

З таблиці видно, що ці слова є ключовими, бо вони відповідають ознакам ключових слів для даний текстів і характеризують їх.

3.2.2 F-міра

Продемонструємо результати роботи іншого алгоритму, навівши приклади з використанням тих самих текстів, але з використанням F-міри.

Таблиця 3.11– Результати для F-міри

Big Data		Text Mining		Text categorization		Text Mining inductor overview	
Word	Weight	Word	Weight	Word	Weight	Word	Weight
analysis	0.04842	Discovering	0.09087	Word	0.08044	Mining	0.0649
varies	0.0443	Learning	0.0587	Sample	0.06145	Numbers	0.0342
Traditional	0.03554	Analysis	0.04284	Learning	0.0602	Variables	0.0341
capture	0.031887	Community	0.2112	Constrain	0.0531	appropriate	0.0319
Information	0.02235	Inferences	0.02004	Parti cul ary	0.04192	Summarise	0.031
Greater	0.021312	Machine	0.0198	Retrieval	0.03791	Automatic	0.0271
Exabytes	0.0193	Databases	0.0161	Traditional	0.0364	algorythms	0.0269
Other	0.01835	statistic	0.01608	Counts	0.0264	Terms	0.0236
Store	0.01805	Phrase	0.01607	Identify	0.02173	department	0.0234
Technological	0.017093	Intersection	0.01445	Training	0.02145	Process	0.0204
Predictive	0.0167			Estimated	0.02054	Design	0.0196
Result	0.01547			Predict	0.01864		
Systems	0.0142						
Requires	0.1405						

Із даних прикладів видно, що цей метод менш ефективний, ніж попередній.

3.2.3 Лінгво-статистичні шаблони

Метод лінгво-статистичних шаблонів значно важчий та набагато більш трудоємкий для реалізації. Тому було прийнято рішення використати готовий продукт KH Coder, де вже реалізований цей метод. Результати роботи наведені таблиці 3.12.

Список словосполучень, виділених програмою, був наведений не повністю, але цього достатньо, щоб зробити висновки щодо роботи алгоритму.

Таблиця 3.12 – Результати для лінгво-статистичних шаблонів частина 1

Big Data		Text Mining	
Word	Weight	Word	Weight
big data	439.161	data mining	428.794
data sets	40.818	of data	40.15
big data analytics	36.523	data analysis	29.659
data lake	17.907	knowledge discovery	24.973
stored data	17.907	data mining step	23.421
data storage	16.181	term data mining	21.51
2.5 exabytes of data	10.647	machine learning	21.22
data size	10.647	data set	19.834
petabytes of data	10.647	data sets	19.834
data set	9.621	data fishing	16.678
process data	8.953	data preparation	16.678
analysis of data sets	8.22	data mining process	13.903
business intelligence	8.207	large data sets	12.444
amount of data	8.09	target data set	11.631

Таблиця 3.13 – Результати для лінгво-статистичних шаблонів частина 2

Word	Weight	Word	Weight
text categorization	66.73	text mining	121.413
document text	21.393	input documents	14.651
document categorization	18.248	text mining	14.28
new document	14.455	data mining project	10.593
text classification	13.441	large numbers of	7.74
language identification	9.391	large numbers	7.737
document profile	8.859	results of text mining	7.14
document retrieval	8.379	text mining algorithms	7.14
new words	7.045	clusters of words	6.387
document's language	6.62	purpose of text mining	6.361
1990s text categorization	5.593	text mining program	6.361
automatic text categorization	5.593	meaning of text	6.16
text categorization methods	5.593	various data mining	5.557
text categorization problem	5.593	text mining	5.464

Таким чином, можемо зробити висновок, що гібридні методи виділення ключових слів є найефективнішими.

У даному розділі було проведено безпосередній аналіз методів, описаних у першому розділі. При аналізі методів виділення стоп-слів було виявлено, що метод, який працює на основі Y-інтерпретації закону Бредфорда чисельно продемонстрував доволі високу точність (близько 85%) проте при більш детальному аналізі було виявлено, що цей метод видаляє найбільш значущі ключові із текстів, що були проаналізовані, що робить

даний метод непридатним для використання у такого роду системах у даному вигляді. Словниковий метод не є універсальним, оскільки універсальний словник робить цей метод менш точним (погіршує результат подальшого виявлення колокацій) та менш повним. Тому для використання даного методу варто використовувати словник стоп-слів, розроблений саме для даної предметної області. Було запропоновано скомбінувати ці методи при аналізі великої кількості текстів різної тематики та направленості. Аналізуються слова, що зустрічаються у цих текстах найчастіше та відбираються за Бредфордом. Чим більша варіативність тематики текстів, тим більша якість словника. При аналізі методів виявлення ключових слів було використано 3 алгоритми: 2 статистичні (міра TF-IDF та F-міра) та гібридний (метод лінгво-статистичних шаблонів). При аналізі результатів було виявлено, що F-міра показала значно гірші результати за TF-IFD. Значно складнішим у реалізації та більш ресурсозатратним в плані виконання є алгоритм лінгво-статистичних шаблонів, проте саме цей алгоритм показав найкращий результат, що було показано у даному розділі.

3.3 Комп’ютерна модель автоматичного аналізу текстової інформації

У даному розділі проводиться оцінка основних характеристик програмного продукту, призначеного для аналізу методів інтелектуального аналізу текстів, що використовуються для структурування знань. Інтерфейс користувача був розроблений за допомогою мови програмування Java у середовищі розробки NetBeans 8.1.

Програмний продукт призначено для використання на персональних комп’ютерах під управлінням операційної системи Windows, Unix, Mac.

Нижче наведено аналіз різних варіантів реалізації модулю з метою вибору оптимальної, з огляду при цьому як на економічні фактори, так і на характеристики продукту, що впливають на продуктивність роботи і на його сумісність з апаратним забезпеченням. Для цього було використано апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) - це технологія, яка дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. Як прямі, так і побічні витрати розподіляються по продуктам та послугам у залежності від потрібних на кожному етапі виробництва обсягів ресурсів. Виконані на цих етапах дії у контексті метода ФВА називаються функціями.

Мета ФВА полягає у забезпеченні правильного розподілу ресурсів, виділених на виробництво продукції або надання послуг, на прямі та непрямі витрати. У даному

випадку - аналізу функцій програмного продукту й виявлення усіх витрат на реалізацію цих функцій.

Фактично цей метод працює за таким алгоритмом:

– Визначається послідовність функцій, необхідних для виробництва продукту. Спочатку - всі можливі, потім вони розподіляються по двом групам: ті, що впливають на вартість продукту і ті, що не впливають. На цьому ж етапі оптимізується сама послідовність скороченням кроків, що не впливають на цінність, і відповідно витрат.

– Для кожної функції визначаються повні річні витрати й кількість робочих годин.

– Для кожної функції на основі оцінок попереднього пункту визначається кількісна характеристика джерел витрат.

– Після того, як для кожної функції будуть визначені їх джерела витрат, проводиться кінцевий розрахунок витрат на виробництво продукту.

3.3.1 Постановка задачі

У роботі застосовується метод ФВА для проведення техніко-економічний аналізу розробки програмного продукту, призначеного для аналізу методів інтелектуального аналізу текстів, що використовуються для структурування знань. Оскільки основні проектні рішення стосуються всієї системи, кожна окрема підсистема має їм задовольняти. Тому фактичний аналіз представляє собою аналіз функцій програмного продукту, призначеного для збору, обробки та проведення аналізу гетероскедастичних процесів в економіці та фінансах.

Відповідно цьому варто обирати і систему показників якості програмного продукту.

Технічні вимоги до продукту наступні:

– програмний продукт повинен функціонувати на персональних комп’ютерах із стандартним набором компонент;

– забезпечувати високу швидкість обробки великих об’ємів даних;

– забезпечувати зручність і простоту взаємодії з користувачем або з розробником програмного забезпечення у випадку використовування його як модуля;

– передбачати мінімальні витрати на впровадження програмного продукту.

3.3.2 Обґрунтування функцій програмного продукту

Головна функція Fo - розробка програмного продукту, який проводить аналіз текстів та виводить результати, а також необхідні статистичні дані. Виходячи з конкретної мети, можна виділити наступні основні функції:

F1 - вибір мови програмування;

F2 - збереження вихідних даних;

F3 - інтерфейс користувача.

Кожна з основних функцій може мати декілька варіантів реалізації. Функція F1.

а) мова програмування Java;

б) мова програмування PHP;

Функція F2.

а) система управління базами даних;

б) виведення у файли Excel.

Функція F3.

а) веб-інтерфейс користувача;

б) інтерфейс користувача, створений за технологією JavaFX;

в) консольний інтерфейс.

Варіанти реалізації основних функцій

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис.3.1). Морфологічна карта відображує всі можливі комбінації варіантів реалізації функцій.

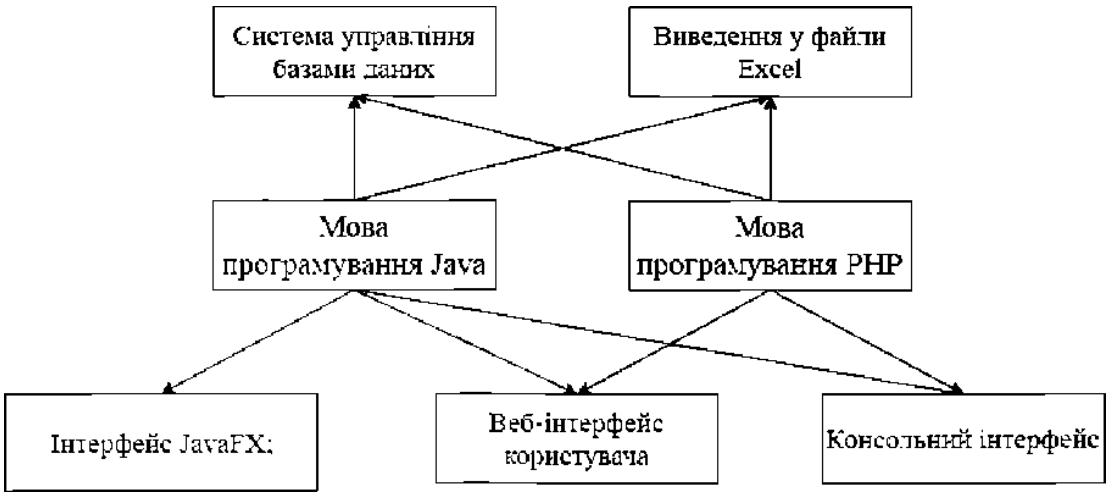

Рисунок 3.1 – Морфологічна карта

На основі цієї карти побудовано позитивно-негативну матрицю варіантів основних функцій (табл.. 3.14).

Таблиця 3.14 – Позитивно-негативна матриця

Основні функції	Варіанти реалізації	Переваги	Недоліки
F1	А	Велика кількість бібліотек з Text Mining у вільному доступі	Займає більше часу при написанні коду
	Б	Займає менше часу при написанні коду	Немає бібліотек з Text Mining у вільному доступі
F2	А	Швидкодія, гнучкість	Необхідно багато додаткового програмного коду, необхідна обробка великої кількості виключень
	Б	Менше програмного коду, зручне подальше використання даних у звітах	Нижча швидкодія, менша розширюваність продукту
F3	А	Можливе розгортання на сервері для подальшого загального користування, зручність користування	Час розробки, велика кількість додаткового коду, що не стосується Text Mining Велика кількість додаткових технологій, що необхідно використати
	Б	Зручність користування	Більше коду
	В	Найшвидший метод розробки, що не вимагає додаткового коду	Незручний для користувача

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція F1:

Оскільки для покращення якості вихідного продукту та зменшення часу розробки деякі функції варто використати з готових бібліотек, що можливо у разі використання Java, варіант б) має бути відкинутий.

Функція F2:

Оскільки в рамках даного дослідження швидкість розробки та презентація даних є пріоритетною, то варіант б) має бути відкинутий

Функція F3:

Інтерфейс користувача не відіграє велику роль у даному програмному продукту, а

варіант а) вимагає набагато більшого часу розробки, тому варіант а) відкидаємо, а вважаємо варіанти б) та в) гідними розгляду.

Таким чином, будемо розглядати такі варіанти реалізації ПП:

1. Fla-F2a-F3б
2. Fla-F2a-F3в

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

3.3.3 Обгрунтування системи параметрів

На підставі даних про основні функції, що повинен реалізувати програмний продукт, вимог до нього, визначаються основні параметри виробу, що будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

XI - Об’єм оперативної пам’яті, що використовується;

X2 - об’єм пам’яті для збереження даних;

XЗ - час обробки даних;

X4 - потенційний об’єм програмного коду.

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію ПП як показано у табл. 3.15.

Таблиця 3.15 – Основні параметри ПП

Назва параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Об'єм оперативної пам’яті	XI	МБ	2000	800	200
Об’єм пам’яті для збереження даних	X2	Мб	128	64	8
Час обробки даних алгоритмом	XЗ	мс	5500	4000	600
Потенційний об’єм програмного коду	X4	кількість строк коду	4500	3000	2000

3.3.4 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі - розробка програмного продукту, який дає найбільш точні результати при знаходженні параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

– визначення рівня значимості параметра шляхом присвоєння різних рангів;

– перевірку придатності експертних оцінок для подальшого використання;

– визначення оцінки попарного пріоритету параметрів;

– обробку результатів та визначення коефіцієнту значимості.

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^{N} r_{ij} R_{ij} = \frac{Nn(n+1)}{2}; \tag{3.1}$$

де N - число експертів, n - кількість параметрів;

б) середня сума рангів

$$T = \frac{1}{n} R_{ij} \tag{3.2}$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T \tag{3.2}$$

Сума відхилень по всім параметрам повинна дорівнювати 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^{N} \Delta i^2 \tag{3.4}$$

Результати експертного ранжування наведені у таблиці 3.16.

Таблиця 3.16 – Результати ранжування параметрів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів Ri	Відхилення Ai	Ai^2
			1	2	3	4	5	6	7			
XI	Об'єм оперативної пам’яті	МБ	3	1	2	2	2	1	2	13	-4.5	20.25
X2	Об’єм пам’яті для збереженн я даних	МБ	4	4	4	4	3	4	4	27	9.5	90.25
X3	Час обробки даних алгоритмом	Мс	1	3	3	3	4	3	3	20	2.5	6.25
X4	Потенційний об’єм програмно го коду	кількість строк коду	2	2	1	1	1	2	1	10	-7.5	56.25
	Разом		10	10	10	10	10	10	10	70	0	173

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3 - n)} \quad (3.4)$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 3.17.

Таблиця 3.17 – Попарне порівняння параметрів

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	>	>	>	>	>	>	>	>	1,5
X1 і X3	<	>	>	>	>	>	>	>	1,5
X1 і X4	<	>	<	<	<	>	<	<	0,5
X2 і X3	<	<	<	<	>	<	<	<	0,5
X2 і X4	<	<	<	<	<	<	<	<	0,5
X3 і X4	>	<	<	<	<	<	<	<	0,5

Числове значення, що визначає ступінь переваги і-го параметра над j-тим, a_{ij} визначається по формулі:

$$a_{ij} = \begin{cases} 1.5 \text{ при } X_i > X_j \\ 1.0 \text{ при } X_i = X_j \\ 0.5 \text{ при } X_i < X_j \end{cases} \quad (3.5)$$

З отриманих числових оцінок переваги складемо матрицю A= || a_{ij} ||.

Для кожного параметра зробимо розрахунок вагомості Кві за наступними формулами:

$$K_{\text{ві}} = \frac{b_i}{\sum_{i=1}^{n} b_i}, \text{де } b_i = \sum_{i=1}^{N} a_{ij}. \quad (3.6)$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{\text{ві}} = \frac{b_i'}{\sum_{i=1}^{n} b_i'}, \text{де } b_i' = \sum_{i=1}^{N} a_{ij} b_j. \quad (3.7)$$

Як видно з таблиці 3.18, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 3.18 – Розрахунок вагомості параметрів

Параметри X_i	Параметри X_J				Перша ітер.		Друга ітер.		Третя ітер	
	XI	X2	X3	X4	b_i	*Кві*	b_i^1	*Кві*^{1}	b_i^2	*Кві*^{2}
XI	1.0	1.5	1.5	0.5	4.5	0.281	20.3	0.29	91.1	0.29
X2	00.5	1.0	0.5	0.5	2.5	0.156	6.25	0.09	15.6	0.05
X3	0.5	1.5	1.0	1.5	3.5	0.219	12.3	0.18	42.9	0.14
X4	1.5	1.5	1.5	1.0	5.5	0.344	30.3	0.44	166	0.53
Всього:					16	1	69	1	316	1

3.3.5 Аналіз рівня якості варіантів реалізації функції

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів Х2 (об'єм пам'яті для збереження даних) та Х1 (об'єм оперативної пам'яті) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра Х4 (кількість строк коду) обрано не найгіршим (не максимальним), тобто це значення відповідає або варіанту а) 4000 або варіанту б) 2000 мс.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 3.19):

$$K_K(j) = \sum_{i=1}^{n} K_{вi,j} B_{i,j}\,, \tag{3.8}$$

де n - кількість параметрів;

$K_{вi}$ коефіцієнт вагомості і-го параметра;

B_i - оцінка і-го параметра в балах.

Таблиця 3.19 – Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1(X1)	А	800	3.25	0.2884	0.9373
F2(X2)	А	64	6.4	0.0494	0.9373
F3(X3,X4)	А	4000	5	0.1357	0.6785
	Б	2000	2.5	0.5265	1.31625

За даними з таблиці 3.19 за формулою (3.9) визначаємо рівень якості кожного з варіантів:

$$K_K = K_{\text{ТУ}}[F_{1k}] + K_{\text{ТУ}}[F_{2k}] + \ldots + K_{\text{ТУ}}[F_{zk}], \tag{3.9}$$

K_{K1} = 0.9373 + 0.9373 + 0.6785 = 1.932

K_{K2} = 0.9373 + 0.9373 + 1.31625= 2.5697

Як видно з розрахунків, кращим є другий варіант, для якого коефіцієнт технічного рівня має найбільше значення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ ДО РОЗДІЛІВ 1-3

1. Drott M. C., Griffith B. C. An Empirical Examination of Bradford's Law and the Scattering of Scientific Literature // Journal of the American Society for Information Science. 1978. Vol. 29, Iss. 5. P. 238-246.

2. Jurafsky D. Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 2nd edition, / Jurafsky D., Martin J.// 2009 [Електронний ресурс]. - Режим доступу: http://www.cse.iitk.ac.in/users/mohit/Speech-and-Language-Processing.pdf.

3. Liu Z., Huang W., Zheng Y., Sun M. Automatic keyphrase extraction via topic decomposition. Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Cambridge, Massachusetts, 2010, pp. 366-376.

4. Sato S., Sasaki Y. Automatic Collection of Related Terms from the Web // The Companion Volume to the Proceedings of 41st Annual Meeting of the ACL, Sapporo, Japan, 2003. – P. 121–124.

5. Барсегян А.А. Технологии анализа данных: Data Mining, Viual Mining, Text Mining, OLAP / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод. - Спб.: БХВ-Петербург, 2007. – 245 с.

6. Большакова Е.И., Кльїшинский З.С., Ландз Д.В., Носков А.А., Пескова О.В., Ягунова Е.В. Автоматическая обработка текстов на естественном язьіке и компьютерная лингвистика: учеб. пособие / Большакова Е.И., Клышинский З.С., Ландз Д.В., Носков А.А., Пескова О.В., Ягунова Е.В. - М.: МИЗМ, 2011. - 272 с.

7. Вавіленкова А.І. Аналіз методів обробки текстової інформації / А.І. Вавіленкова / Вісник НТУ ХПІ. Серія «Інформатика та моделювання». – Х.: НТУ ХПІ. - 2013. - N° 39 (1012). - С. 35 - 40.

8. Вавіленкова А.І. Логіко-лінгвістична модель як засіб відображення синтаксичних особливостей текстової інформації / А.І. Вавіленкова // Математичні машини та системи. - 2010. - № 2. - С. 134-137.

9. Верес О. М., Оливко Р. М. Класифікація методів аналізу великих даних Електронний ресурс]. - Режим доступу: http://webcache.googleusercontent.com/search?q=cache:sF7cXDQu9hgJ:science.lpnu.ua/sites/default/files/journal-paper/2018/jun/13005/ilovepdfcom-84-92.pdf+&cd=14&hl=ru&ct=clnk&gl=ua

10. Волошин В.Г. Комп'ютерна лінгвістика: Навчальний посібник / В.Г. Волошин. - Суми: Університетська книга, 2004. - 382 с.

11. Губин М.В. Модели и методы представлення текстового документа в системах информационного поиска / М.В. Губин // Научно-техническая информация. Сер. 1. - 2004. С. 12-24.

12. Нога Р. Аналітичний огляд методів та засобів опрацювання текстової інформації / Р. Нога, Н. Б. Шаховська // Вісн. Нац. ун-ту "Львів. політехніка". - 2011. - № 715. - С. 323-332.

13. Оксанич І.Г., Піскунов Д.М., Черниш Д.П. Інтелектуальний аналіз масиву текстових документів на основі технології Text Mining / І.Г. Оксанич, Д.М. Піскунов, Д.П. Черниш // Системи обробки інформації. – 2013. – № 2(109). – С. 139-143.

14. Партико З.В. Прикладна і комп'ютерна лінгвістика / З.В. Партико. – Львів: «Афіша», 2008. - 221 с.

15. Ситник В. Ф. Інтелектуальний аналіз даних (дейтамайнінг): навч. посіб. / В. Ф. Ситник, М. Т. Краснюк. – К.: КНЕУ, 2007. – 376 с.

16. Тарануха В.Ю. Інтелектуальна обробка текстів: [навчальний посібник] / В. Ю.Тарануха. - Київ: електронна публікація на сайті факультету, 2014. - 80 с.

17. Чубукова И. А. Data Mining: учеб. пособ. /И. А. Чубукова. – М. : Интернет-университет информационных технологий: БИНОМ: Лаборатория знаний, 2006. – 382 с.

18. Шаховська Н. Б. Організація великих даних у розподіленому середовищі / Н. Б. Шаховська, Ю. Я. Болюбаш, О. М. Верес // Обчислювальна техніка та автоматизація: [зб. наук. пр. ДонНТУ]. – Донецьк, 2014. – С. 147–155. – (Вісник / ДонНТУ ; № 2 (27). – С. 13-19.

19. Яцко В.А., Стариков М.С., Ларченко Е.В. Алгоритмы предварительной обработки текста: декомпозиция, аннотирование, морфологический анализ [Текст] / В.А. Яцко, М.С. Стариков, Е.В. Ларченко [и др.] // Научно-техническая информация. Сер. 2. – 2009. – № 11. – С. 8-18.

РОЗДІЛ 4
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

Метою роботи є розробка методів, що базуються на використанні технології Text Mining, яка дозволяє підвищити якість і швидкість виконання автоматичної кластеризації документів, та програмна реалізація системи аналізу текстового масиву Так як в процесі проектування використовувалося комп’ютерне обладнання, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для умов праці з використанням персонального комп'ютера.

В даному розділі проведено аналіз потенційних небезпечних та шкідливих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

4.1 Загальні питання з охорони праці

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» [1] визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави і особистої відповідальності.

4.2 Аналіз стану умов праці

Робота над створенням системи та дослідження методів інтелектуального аналізу тексту проходитиме в приміщенні багатоквартирного будинку. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп’ютером.

4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 4.1/

Таблиця 4.1 – Розміри приміщення

Найменування	**Значення**
Довжина, м	5
Ширина, м	5
Висота, м	2.8
Площа, $м^2$	25
Об’єм, $м^3$	70

Згідно з [2] розмір площі для одного робочого місця оператора персонального комп’ютера має бути не менше 6 кв. м, а об’єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за [3] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Приміщення знаходиться на другому поверсі трьох поверхової будівлі і має об’єм 70 $м^3$, площу – 25 $м^2$. Обладнано одне місце праці укомплектоване ПК.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	770	680 ÷800
Висота простору для ніг, мм	750	не менше 600
Ширина простору для ніг, мм	550	не менше 500
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	450	400 ÷500
Ширина сидіння, мм	450	не менше 400

Продовження таблиці 4.2 – Характеристики робочого місця

Глибина сидіння, мм	470	не менше 400
Висота поверхні спинки, мм	400	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	750	700 ÷800

Температура в приміщенні протягом року коливається у межах 18–24°С, відносна вологість — близько 50%. Швидкість руху повітря не перевищує 0,2 м/с. Шум знаходиться на рівні 50 дБА. Система вентилювання приміщення — природна неорганізована, а опалення — централізоване.

4.2.3 Навантаження та напруженість процесу праці

За фізичним навантаженням робота відноситься до категорії легкі роботи (Іа), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав’язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту.

Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви програм тривалістю 15 хв. через кожну годину роботи.

4.3 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Аналіз небезпечних та шкідливих факторів виконується у табличній формі (табл. 4.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00-7.15-18 [6] «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга U=+220В +-5%;
- робочий струм I=2А;
- споживана потужність P=350 Вт.

Робоче місце має відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [3].

Таблиця 4.3 – Аналіз небезпечних і шкідливих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
Фізичні			
- підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	система організаційних і технічних заходів і засобів, що забезпечують захист від шкідливого і небезпечного впливу електричного струму, електричної дуги, електромагнітного поля і статичної електрики.	4	[4]
- недостатність природного світла	порушення умов праці (вимог до приміщень)	2	[5]
- недостатнє освітлення робочої зони	порушення гігієнічних параметрів	3	[5]

Продовження таблиці 4.3 – Аналіз небезпечних і шкідливих факторів

психофізіологічні:			
- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи	4	[6] [3]
- фізичні (статичне – сидіння)	порушення умов праці (організації місця праці- сидіння користувача,) та організації робочого часу - безпервна робота)	2	[6] [3]

4.3.2 Пожежна безпека

Небезпека розвитку пожежі обумовлюється застосуванням розгалужених систем електроживлення ЕОМ, вентиляції і кондиціювання.

Запобігти утворенню горючого середовища (замінити горючі речовини і матеріали на негорючі і важкогорючі) не надається технічно можливим. Тому проектом передбачаються засоби запобігання утворення (або внесення) в горюче середовище джерел запалювання.

Згідно ДБН В.2.5-28:2018 [5] таке приміщення, площею 25 м^2, відноситься до категорії "В" (пожежонебезпечної) та для протипожежного захисту в ньому можливо встановлення автоматичної пожежної сигналізації із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота,

аміак, фенол, ацетон, стирол.

4.3.3 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

4.4 Гігієнічні вимоги до параметрів виробничого середовища

4.4.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. Оптимальні значення мікроклімату для робочого місця відповідають ДСН 3.3.6.042-99 [2] (табл. 4.4):

Таблиця 4.4 – Норми мікроклімату робочої зони об’єкту

Період року	**Категорія робіт**	**Температура C^0**	**Відносна вологість %**	**Швидкість руху повітря, м/с**
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату. Дане приміщення обладнане системою опалення, кондиціонування повітря. Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

Рівні позитивних і негативних іонів у повітрі мають відповідати ДСН 3.3.6.042-99 [2].

4.4.2 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає ДБН В.2.5-28:2018 [5]. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє ДБН і для даного приміщення в світлий час доби достатньо природного освітлення.

Розрахунок освітлення.

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше -1/8, в побутових – 1/10:

$$S_b = \left(\frac{1}{5} \div \frac{1}{10}\right) \cdot S_n, \qquad (4.1)$$

де S_b – площа віконних прорізів, м^2;

S_n – площа підлоги, м^2.

$S_n = a \cdot b = 5 \cdot 5 = 25$ м^2,

$S = 1/8 \cdot 25 = 3{,}125$ м^2.

Приймаємо 2 вікна площею S=1,6 м^2 кожне.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні.

Розрахунок кількості світильників n виробляється по формулі (4.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \qquad (4.2)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, м^2; S = 25 м^2;

Z – поправочний коефіцієнт світильника (Z=1,15 для ламп розжарювання та ДРЛ; Z = 1,1 для люмінесцентних ламп) приймаємо рівним 1,1;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

M – число люмінесцентних ламп в світильнику – 2;

F – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (А.2), отримуємо:

$$n = \frac{300 \cdot 25 \cdot 1{,}1 \cdot 1{,}5}{5400 \cdot 0{,}575 \cdot 2} \approx 2{,}0$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220 В.

4.4.3 Шум та вібрація, електромагнітне випромінювання

Рівень шуму, зумовлений як роботою системного блоку, клавіатури, так і друкуванням на принтері, а також зовнішніми чинниками, коливається у межах 50–65 дБА ДСН 3.3.6.042-99 [2].

Віброізоляцію можливо здійснювати за допомогою спеціальної прокладки під системний блок, яка послаблює передачу вібрацій робочого столу. Вібрація на робочому місці в приміщенні, що розглядається, відповідає нормам ДСН 3.3.6.042-99 [2].

4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання передбачено наступні заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціювання повітря.

Крім того, потрібно дотримуватися правил безпеки під час експлуатації інших електричних приладів та вимоги безпеки при надзвичайних ситуаціях.

Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом, приміщення в якому проводиться робота відноситься до першого класу (без підвищеної небезпеки). Коефіцієнт використання вертикальних заземлювачів $\eta_в$ в залежності від розміщення заземлювачів та їх кількості знаходиться в межах 0,4…0,99. Взаємну екрануючу дію горизонтального заземлювача (з'єднувальної смуги) враховують за допомогою коефіцієнта використання горизонтального заземлювача η_c.

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів $R_{шт.з.}$:

$$R_{шт.з.} = \frac{R_д \cdot R_{пр.з.}}{R_{пр.з.} - R_д}, \qquad (4.3)$$

де $R_{пр.з.}$ – опір природних заземлювачів; $R_д$ – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то $R_{шт.з.}=R_д$.

Підставивши числові значення у формулу (А.3), отримуємо:

$$R_{шт.з.} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом}$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ, Ом·м. Приблизне значення питомого опору глини приймаємо ρ=40 Ом•м (табличне значення).

3) Розрахунковий питомий опір ґрунту, $\rho_{розр}$, Ом·м, визначається відповідно для вертикальних заземлювачів $\rho_{розр.в}$, і горизонтальних $\rho_{розр.г}$, Ом·м за формулою:

$$\rho_{розр..} = \psi \cdot \rho, \tag{4.4}$$

де ψ – коефіцієнт сезонності для вертикальних заземлювачів I кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $\rho_{розр.в}$=1,7 і горизонтальних $\rho_{розр.г}$=5,5 Ом·м.

$$\rho_{розр.в} = 1,7 \cdot 40 = 68 \text{ Ом•м}$$

$$\rho_{розр.г} = 5,5 \cdot 40 = 220 \text{ Ом•м}$$

4) Розраховується опір розтікання струму вертикального заземлювача $R_в$, Ом, за формулою (4.5).

$$R_в = \frac{\rho_{розр.в}}{2 \cdot \pi \cdot l_в} \cdot \left(\ln \frac{2 \cdot l_в}{d_{ст}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_в}{4 \cdot t - l_в} \right), \tag{4.5}$$

де $l_в$ – довжина вертикального заземлювача (для труб - 2−3 м; $l_в$=3 м);

$d_{ст}$ – діаметр стержня (для труб - 0,03−0,05 м; $d_{ст}$=0,05 м);

t – відстань від поверхні землі до середини заземлювача, яка визначається за формулою (4.6):

$$t = h_в + \frac{l_в}{2}, \tag{4.6}$$

де $h_в$ – глибина закладання вертикальних заземлювачів (0,8 м); тоді $t = 0,8 + \frac{3}{2} = 2,3$ м

$$R_в = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left(\ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}$$

5) Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання $\eta_в$:

$$n = \frac{2 \cdot R_в}{R_д} = \frac{2 \cdot 18,5}{4} = 9,25 \tag{4.7}$$

І визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки $\eta_в$ =0,57 (табличне значення).

6) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання $n_в$, шт:

$$n_в = \frac{2 \cdot R_в}{R_д \cdot \eta_в} = \frac{2 \cdot 18,5}{4 \cdot 0,57} = 16,2 \approx 16 \qquad (4.8)$$

7) Визначається довжина з'єднувальної стрічки горизонтального заземлювача $l_с$, м:

$$l_с = 1,05 \cdot L_в \cdot (n_в - 1), \qquad (4.9)$$

де $L_в$ – відстань між вертикальними заземлювачами, (прийняти за $L_в$ = 3м);

$n_в$ – необхідна кількість вертикальних заземлювачів.

$$l_с = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м}$$

8) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки) $R_г$, Ом:

$$R_г = \frac{\rho_{розр.г}}{2 \cdot \pi \cdot l_с} \cdot \ln \frac{2 \cdot l_с^{\,2}}{d_{см} \cdot h_г}, \qquad (4.10)$$

де $d_{см}$ – еквівалентний діаметр смуги шириною b, $d_{см}$ =0,95b, b = 0,15 м;

$h_г$ – глибина закладання горизантальних заземлювачів (0,5 м);

$l_с$ - довжина з'єднувальної стрічки горизонтального заземлювача $l_с$, м

$$R_г = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом}$$

9) Визначається коефіцієнт використання горизонтального заземлювача $\eta_{с.}$ відповідно до необхідної кількості вертикальних заземлювачів $n_в$. Коефіцієнт використання з'єднувальної смуги $\eta_с$=0,3 (табличне значення).

10) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{заг} = \frac{R_в \cdot R_г}{R_в \cdot \eta_с + R_г \cdot n_в \cdot \eta_в} \le R_д. \qquad (4.11)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{заг}$ <4 Ом, а саме:

$$R_{заг} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \le R_д$$

4.6 Висновки до розділу 4

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в кваліфікаційній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника.

Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Було наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп’ютері.

У двадцять першому столітті актуальною проблемою є забруднення навколишнього середовища. Проблеми з екологією зустрічаються в повсякденному житті і в будь-який діяльності людини, і не є винятком сфери пов'язаний інформаційними технологіями. Важливо розуміти що використання несправного обладнання або неправильної експлуатації, впливає не тільки на здоров'я людини, а також на навколишнє середовище. Так само потрібно сортувати і утилізувати відходи в процесі роботи. Виходячи з вищесказаного можна зробити висновок що дотримуватися нормативних документів охорони праці є обов'язковим.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ ДО РОЗДІЛУ 4

1. Закон України «Про охорону праці». Режим доступу: https://zakon.rada.gov.ua/laws/show/2694-12 - 10.14.1992 р.

2. Державні санітарні норми. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» - Режим доступу: https://zakon.rada.gov.ua/rada/show/va042282-99 - 01.02.1999 р.

3. Державні санітарні правила і норми. ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» - Режим доступу: https://zakon.rada.gov.ua/rada/show/v0007282-98 - 10.12.1998 р.

4. Державний стандарт України. ДСТУ Б В.2.5-82:2016 «Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом» - Режим доступу: http://epicentre.co.ua/dstu/doc28522.html - 01.07.2016 р.

5. Державні будівельні норми. ДБН В.2.5-28:2018 «Природне і штучне освітлення» - Режим доступу: http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf - 03.10.2018

6. Нормативно-правовий акт з охорони праці. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» - Режим доступу: https://zakon.rada.gov.ua/laws/show/z0508-18 - 14.02.2018 р.

ВИСНОВКИ

У рамках магістерської роботи був розроблений і реалізований метод інтелектуального аналізу текстового масиву за допомогою технології Text Mining. У результаті роботи здійснена програмна реалізація системи аналізу текстового масиву.

Підґрунтя технології Text Mining – статистичний та лінгвістичний аналіз, методи штучного інтелекту. Ця технологія застосовується для проведення аналізу, забезпечення навігації та пошуку в неструктурованих текстах. Застосування інформаційних систем класу Text Mining дає змогу користувачам набувати нових знань.

Технології Text Mining – набір методів, які призначені для видобування відомостей з текстів на основі сучасних ІКТ, що дає змогу виявити закономірності, які забезпечують користувачам отримання корисних даних та нових знань.

Основними методами технології Text Mining є: класифікація (classification); кластеризація (clustering); побудова семантичних мереж або аналіз зв'язків (Relationship, Event andFact Extraction); здобуття феноменів, фактів, понять (feature extraction); автоматичне реферування, створення анотацій (summarization); відповідь на запити (question answering); тематичне індексування (thematic indexing); пошук за ключовими словами (keyword searching); засоби підтримки та створення таксономії (oftaxonomies) і тезаурусів (thesauri).

Прикладом ефективного застосовування технологій Text Mining є проведення контент-аналізу. Контент-аналіз – це якісно-кількісне, систематичне опрацювання, оцінювання та інтерпретація форми і змісту тексту.

У третьому розділі було проведено безпосередній аналіз методів, описаних у першому розділі. При аналізі методів виділення стоп-слів було виявлено, що метод, який працює на основі Y-інтерпретації закону Бредфорда чисельно продемонстрував доволі високу точність (близько 85%) проте при більш детальному аналізі було виявлено, що цей метод видаляє найбільш значущі ключові із текстів, що були проаналізовані, що робить даний метод непридатним для використання у такого роду системах у даному вигляді. Словниковий метод не є універсальним, оскільки універсальний словник робить цей метод менш точним (погіршує результат подальшого виявлення колокацій) та менш повним. Тому для використання даного методу варто використовувати словник стоп-слів, розроблений саме для даної предметної області. Було запропоновано скомбінувати ці методи при аналізі великої кількості текстів різної тематики та направленості. Аналізуються слова, що зустрічаються у цих текстах найчастіше та відбираються за

Бредфордом. Чим більша варіативність тематики текстів, тим більша якість словника. При аналізі методів виявлення ключових слів було використано 3 алгоритми: міра TF-IDF, F-міра та метод лінгво-статистичних шаблонів. При ручному аналізі результатів було виявлено, що F-міра показала значно гірші результати за TF-IFD. Значно складнішим у реалізації та більш ресурсозатратним в плані виконання є алгоритм лінгво-статистичних шаблонів, проте саме цей алгоритм показав найкращий результат, що було показано у 3 розділі. Для фінального структурування знань існує 2 найпоширеніших методи - Naive Bayes та метод Роше.

Перспективою подальших досліджень є розробка повноцінної системи структурування знань з використанням інтелектуального аналізу текстової інформації з використанням модифікованих алгоритмів та словників її розгортання на веб-сервері.

ДОДАТОК А

Програмний код

```
1       package model.entity;
2       import java.io.FileOutputStream;
3       import java.io.IOException;
4       import java.util.ArrayList;
5       import java.util.Arrays;
6       import java.util.Collections;
7       import java.util.Comparator;
8       import java.util.HashSet;
9       import java.util.Random;
10      import java.util. Set;
11      import java.util.logging.Level;
12      import java.util.logging.Logger;
13      import org.apache.poi.hssf.usermodel.HSSFCell;
14      import org.apache.poi.hssf.usermodel.HSSFRow;
15      import org.apache.poi.hssf.usermodel.HSSFSheet;
16      import org.apache.poi.hssf.usermodel.HSSFWorkbook;
17      import service.prepare.TextPrepare;
18      import service.util.WordComparator;
19      *
20      * @author PRIEST
21      */
22      public class Text {
23      private int id; private String name; private String preperedText;
24  private String lingMatkUp; private int setOfTextsId; private int
25  wordsAmount;
26      private ArrayList<Paragraph> paragraphs = new
27  ArrayList<Paragraph>(); private ArrayList<Word> words = new
28  ArrayList<Word>(); private Set<String> stopWordsBradford = new
29  HashSet<String>();
30      // private Set<String> stopWordsDict = new HashSet<String>();
31  private ArrayList<Word> stopWordsTotal = new ArrayList<Word>();
32      public void uniteWords(Word word){ for (Word stopWord:
33  stopWordsTotal){
```

```
34        if(stopWord.getWoid().equalsIgnoreCase(word.getWord())){
35  stopWord.setAmount(word.getAmountO+stopWord.getAmountO);
36  System.out.println("summ" + word.getWordO);
37        } else{
38        stopWordsTotal.add(word);
39        System.out.println("add" + word.getWordO);
40        }
41        }
42        }
43        public void sortWordsO {
44        Comparator<Word> wc = new WordComparator();
45        Collections.sort( words, wc);
46        }
47        public void getStopWordByDictToExcel(){ int totalAmount = 0;
48        for(Word word: words){ for(String diet: StopWordsList.getList())
49        {
50        if(word. getW ord().equalsIgnoreCase(dict)) {
51        // System.out.println(dict + " " + word.getAmount());
52        totalAmount+=word. getAmount();
53        break;
54        }
55        }
56        }
57        System.out.println(this.wordsAmount + " " + totalAmount);
58        }
59        public void processWordsStemming() {
60        ArrayList<Paragraph>  paragraphs  =  this.getParagraphs();  for
61  (Paragraph paragraph : paragraphs) {
62        paragraph.setParagraphString(TextPrepare.deletePunctuation(paragr
63  aph.getParagraphString()));    paragraph.    setW    ords(TextPrepare.
64  splitWords(paragraph.getParagraphString()));
65        }
66        for  (Paragraph  paragraph  :  paragraphs)  {  ArrayList<String>
67  strTest = new ArrayList<String>();
68        strTest.addAll(paragraph.getWords()); for (String str : strTest)
69  {
70        String stemmedWord = TextPrepare.stemm(str); //String stemmedWord
71  = str; paragraph.getStemmedT ext(). add(stemmedWord);
```

```
72        Word word = this.findWord(stemmedWord); if (word.hasForm(str)) {
73  word. incAmount();
74        } else {
75        word. addForm(str); word. incAmount();
76        }
77        this.incW ordsAmount();
78        }
79        }
80        }
81        public void deteleStopWords(){
82        }
83        public void toExcel() { try {
84        FileOutputStream fileOut = null; if (this.name != "") {
85        fileOut = new FileOutputStream(name + ".xls");
86        } else {
87        Random rand = new Random();
88        fileOut = new FileOutputStream(Integer.toString(rand.nextInt()) +
89  ".xls");
90        }
91        HSSFWorkbook workbook = new HSSFWorkbook();
92        HSSFSheet worksheet = workbook.createSheet("stat");
93        HSSFRow row1 = worksheet.createRow(0);
94        HSSFCell          cell01          =          row1.createCell(0);
95  cell01.setCellValue("Word");
96        HSSFCell cell02 = row1.createCell(1);
97        cell02.setCellValue(" Amount");
98        HSSFCell      ceIIO3      =      row      1      .createCell(2);
99  cell03.setCellValue("Percentage");
100       int i = 1;
101       for (Word word : this.getWords()) {
102       HSSFRow row = worksheet. createRow(i++);
103       HSSFCell    cellO    =    row.createCell(O);    cellO.    setCell
104 Value(word.getWord());
105       HSSFCell    celll    =    row.createCell(l);    cell    1.    setCell
106 Value(word.get Amount());
107       HSSFCell cell2 = row.createCell(2);
108       cell2.setCellValue( 100 * (double) word.getAmount() / (double)
109 this.getWordsAmount());
110       workbook. write(fileOut);
```

```
111         fileOut.flush();
112         fileOut.close();
113         } catch (TOException ex) {
114         Logger.getLogger(Text.class.getName()).log(Level.SEVERE,     null,
115   ex);
116         }
117         stopWoidsBradfordtoExcelO;
118         }
119         public void stopWordsBradfordtoExcel() {
120         try {
121         FileOutputStream fileOut = null; if (this.name != "") {
122         fileOut = new FileOutputStream(name + "Brad.xls");
123         } else {
124         Random rand = new Random();
125         fileOut = new FileOutputStream(Integer.toString(rand.nextInt()) +
126   "Brad.xls");
127         }
128         HSSFWorkbook workbook = new HSSFWorkbook();
129         HSSFSheet worksheet2 = workbook.createSheet("Bradford");
130         HSSFRow rowll = worksheet2.createRow(0);
131         HSSFCell  celll  1  =  rowll.createCell(O);  celll  l.setCellValue("
132   Word");
133         HSSFCell  celll2  =  rowll.createCell(l);  cell  12.  setCell  Value("
134   Amount");
135         HSSFCell   ceIII3   =   rowll.createCell(2);   cell   13.   setCell
136   Value("Percentage"); int i=0;
137         for (intj = 0;j < words Amount/3; j+=words.get(i).getAmount()) {
138         HSSFRow row2 = worksheet2.createRow(i+l);
139         HSSFCell  cell20  =  row2.createCell(0);  cell20.  setCell  Value
140   (words, get(i). getWord());
141         HSSFCell cell21 = row2.createCell(l);
142         cell21. setCell Value (words, get(i). getAmount());
143         HSSFCell cell22 = row2.createCell(2);
144         //   System.out.println(words.get(i).getAmount()   +   "   "   +
145   words.get(i).getWord());
146         cell22.setCellValue(100  *  (double)  words.get(i).getAmount()  /
147   (double) this.getWordsAmount());
148         i++;
```

```
149       } for (intj = words Amount/3; j < 2*wordsAmount/3;
150   j+=words.get(i).getAmount()) {
151       HSSFRow row2 = worksheet2.createRow(i+l);
152       HSSFCell          cell20          =          row2.createCell(4);
153   cell20.setCellValue(words.get(i).getWord());
154       HSSFCell          cell21          =          row2.createCell(5);
155   cell21.setCellValue(words.get(i).getAmoimtO);
156       HSSFCell cell22 = row2.createCell(6);
157       //   System.out.println(words.get(i).getAmount()   +   "   "   +
158   words.get(i).getWord());     cell22.setCellValue(100     *     (double)
159   words.get(i).getAmount() / (double) this.getWordsAmountO);
160       i++;
161       }
162       for     (intj     =     2*wordsAmount/3;     j     <     wordsAmount;
163   j+=words.get(i).getAmountO) {
164       HSSFRow row2 = worksheet2.createRow(i+l);
165       HSSFCell          cell20          =          row2.createCell(8);
166   cell20.setCellValue(words.get(i).getWord());
167       HSSFCell cell21 = row2.createCell(9);
168       cell21 .setCellValue(words.get(i).getAmount());
169       HSSFCell cell22 = row2.createCell(10);
170       //   System.out.println(words.get(i).getAmount()   +   "   "   +
171   words.get(i).getWord());     cell22.setCellValue(100     *     (double)
172   words.get(i).getAmount() / (double) this.getWordsAmountO);
173       i++;
174       }
175       workbook. write(fileOut); fileOut flush(); fileOut.close();
176       } catch (IOException ex) {
177       Logger.getLogger(Text.class.getName()).log(Level.SEVERE,     null,
178   ex);
179       }
180       }
181       public void printAllWordsO {
182       for (Word word : words) { word.printWord();
183       System.out.print("    "    +    Double.toString(100    *    (double)
184   word.getAmount() / (double) wordsAmount) +
185       }
186       System.out.print("/nTotal: " + wordsAmount + " words");
187       }
```

```
188       public Word findWord(String stemmed) { for (Word word : words) {
189   if (word.getWord().equalsIgnoreCase(stemmed)) { return word;
190       }
191       }
192       this.words.add(new Word(stemmed)); return findWord(stemmed);
193       }
194       public void incWordsAmount() { this.wordsAmount++;
195       }
196       public void setWords(ArrayList<Word> words) { this.words = words;
197       }
198       public ArrayList<Word> getWords() { return words;
199       }
200       public int getId() { return id;
201       }
202       public String getName() { return name;
203       }
204       public String getPreperedText() { return preperedText;
205       }
206       public String getLingMatkUp() { return lingMatkUp;
207       }
208       public int getSetOfTextsId() { return setOfTextsId;
209       }
210       public int getWordsAmount() { return wordsAmount;
211       }
212       public ArrayList<Paragraph> getParagraphs() { return paragraphs;
213       }
214       public void setId(int id) { this.id = id;
215       }
216       public void setName(String name) { this.name = name;
217       }
218       public     void     setPreperedText(String     preperedText)     {
219   this.preperedText = preperedText;
220       }
221       public void setLingMatkUp(String lingMatkUp) { this.lingMatkUp =
222   lingMatkUp;
223       }
224       public void setSetOfTextsId(int setOfTextsId) { this.setOfTextsId
225   = setOfTextsId;
226       }
```

```
227       public void setWordsAmount(int wordsAmount) { this.wordsAmount =
228   wordsAmount;
229       }
230       public  void  setParagraphs(ArrayList<Paragraph>  paragraphs)  {
231   this.paragraphs = paragraphs;
232       }
233       }
234       /*
235       To change this license header, choose License Headers in Project
236   Properties.
237       To change this template file, choose Tools | Templates
238       and open the template in the editor.
239       */
240       package service.prepare;
241       import config.SplitterConfig;
242       import java.util. ArrayList;
243       import java.util.logging.Level;
244       import java.util.logging.Logger;
245       import model.entity .Paragraph;
246       import model.entity.Word;
247       import org.tartarus. snowball. SnowballStemmer;
248       import org.tartarus.snowball.ext.englishStemmer;
249       *
250       * @author PRIEST
251       */
252       public class TextPrepare {
253       public                  static                  ArrayList<Paragraph>
254   uniteParagraphs(ArrayList<Paragraph>  list)  {  for  (int  i  =  0;  i  <
255   list.size(); i++) {
256       }
257       return list;
258       }
259       public static String deletePunctuation(String text) {
260       text = text.replaceAll(SplitterConfig.punctuationRegExp,"");
261       text = text.replaceAll("\V","");
262       text = text.replaceAll("\\!"," !");
263       text = text.replaceAll("\\.",".");
264       text = text.replaceAll("\\?"," ?");
265       return text;
```

```
266         }
267         public static ArrayList<String> splitWords(String text) {
268         ArrayList<String> words = new ArrayList<String>();
269         String[] strWords = text.split(" "); boolean toLowerCase = true;
270   for   (int   i   =   0;   i   <   strWords.length;   i++)   {   if
271   (strWords[i].equals(".")       ||       strWords[i].equals("!")       ||
272   strWords[i].equals("?")) { toLowerCase = true;
273         } else if (toLowerCase) { words.add(strWords[i].toLowerCase());
274   toLowerCase = false;
275         } else {
276         words.add(strWords[i]);
277         }
278         }
279         return words;
280         }
281         public static String stemm( String wordStr) {
282         String word = "";
283         Class stemClass;
284         try {
285         stemClass                                                    =
286   Class.forName("org.tartarus.snowball.ext.englishStemmer");
287   SnowballStemmer stemmer = (SnowballStemmer) stemClass.newInstance();
288   stemmer.    setCurrent(wordStr);    stemmer.    stem();    word    =
289   stemmer.getCurrentO;
290         } catch (ClassNotFoundException ex) {
291         Logger.getLogger(TextPrepare.class.getName()).log(Level.   SEVERE,
292   null, ex); } catch (InstantiationException ex) {
293         Logger.getLogger(TextPrepare.class.getName()).log(Level.   SEVERE,
294   null, ex); } catch (IllegalAccessException ex) {
295         Logger.getLogger(TextPrepare.class.getName()).log(Level.   SEVERE,
296   null, ex);
297         }
298         return word;
299         }
```

Додаток Б

Комп’ютерна презентація

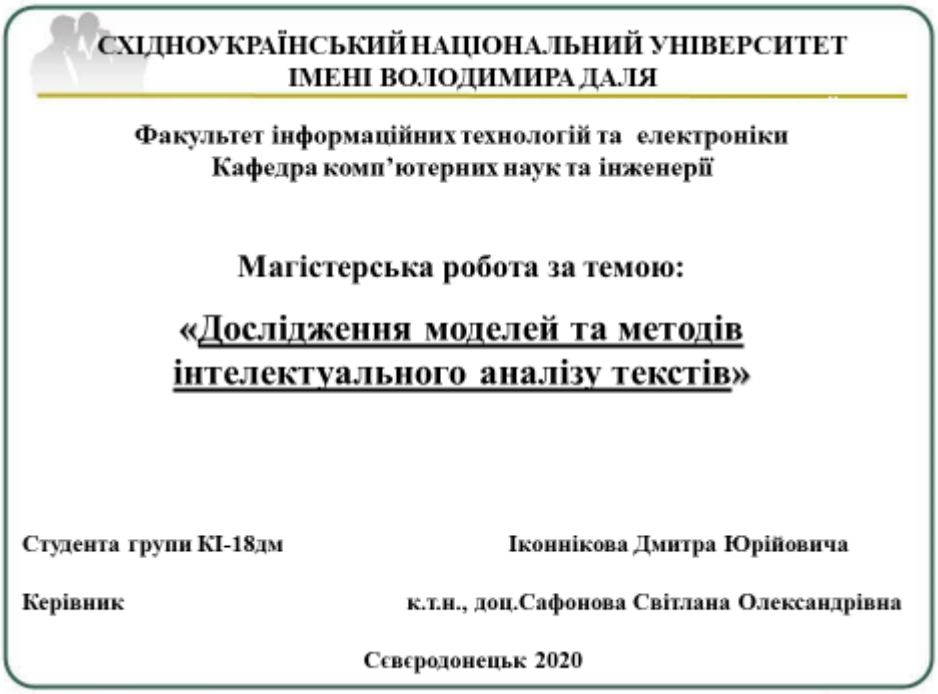

Рисунок Б.1- Слайд №1

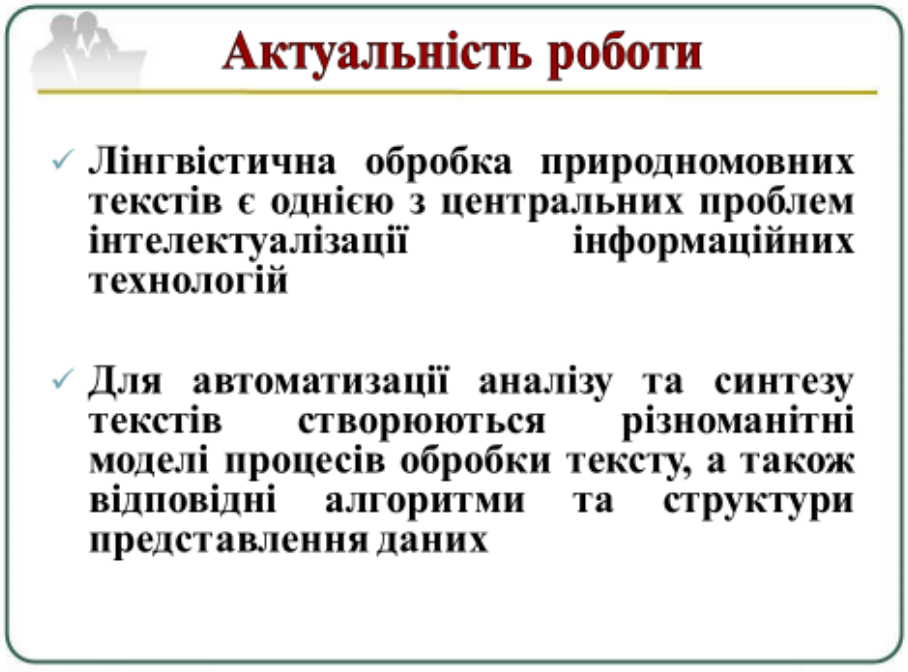

Рисунок Б.2- Слайд №2

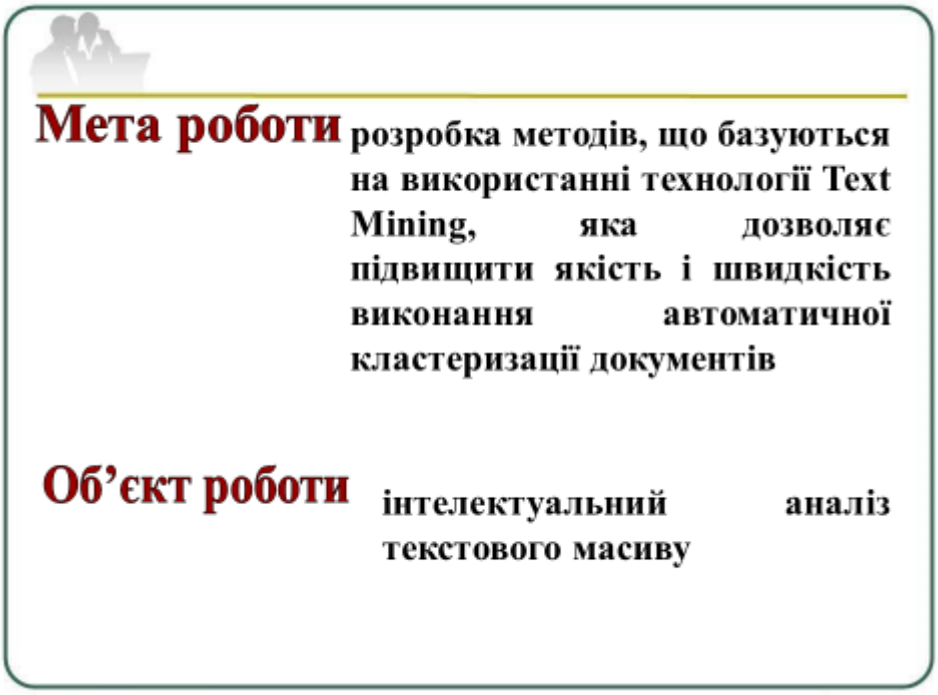

Рисунок Б.3- Слайд №3

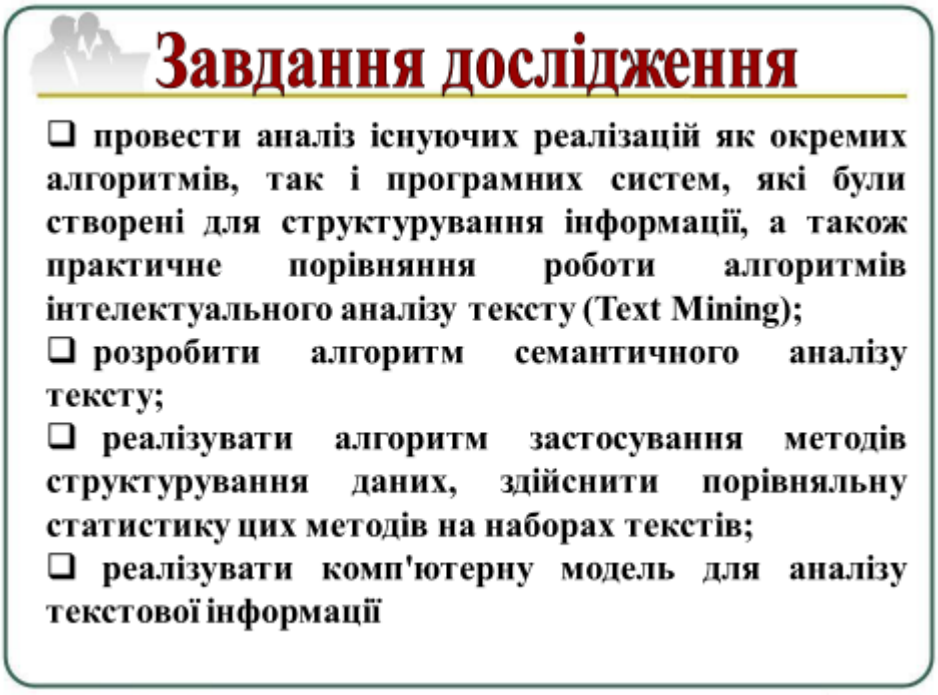

Рисунок Б.4- Слайд №4

Рисунок Б.5- Слайд №5

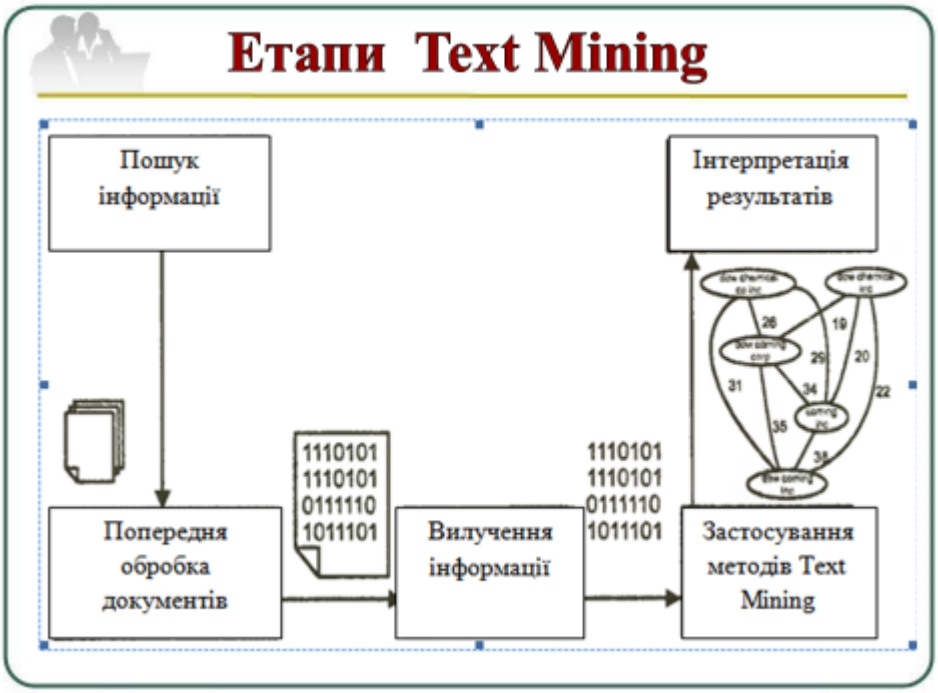

Рисунок Б.6- Слайд №6

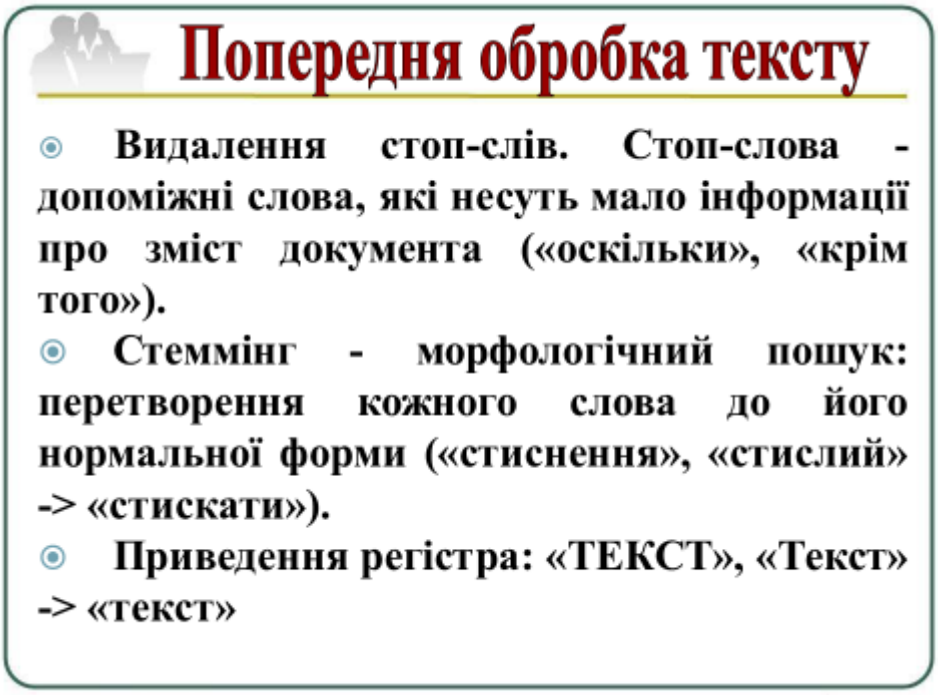

Рисунок Б.7- Слайд №7

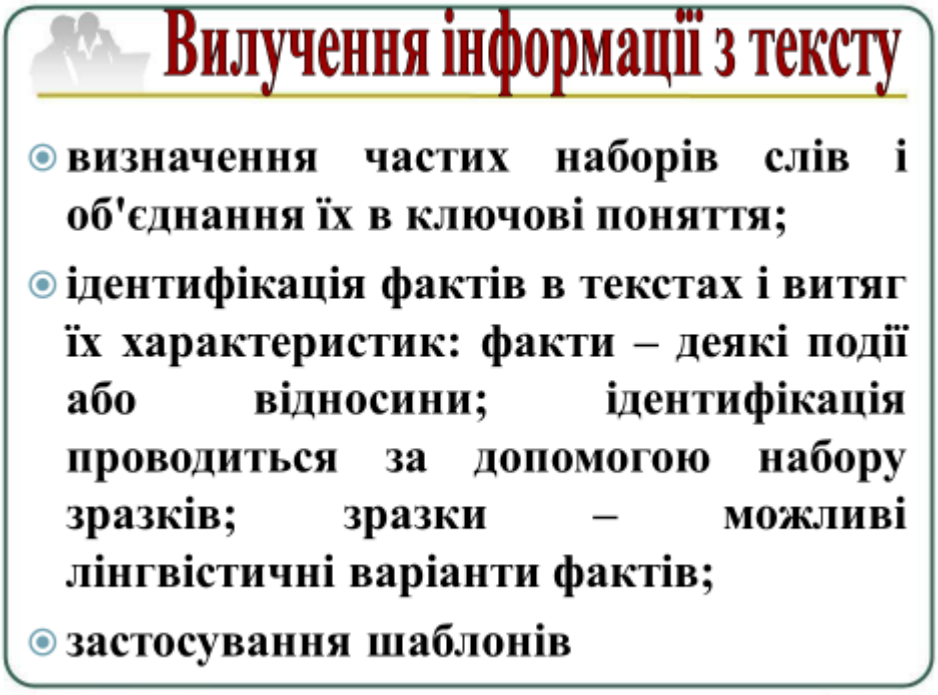

Рисунок Б.8- Слайд №8

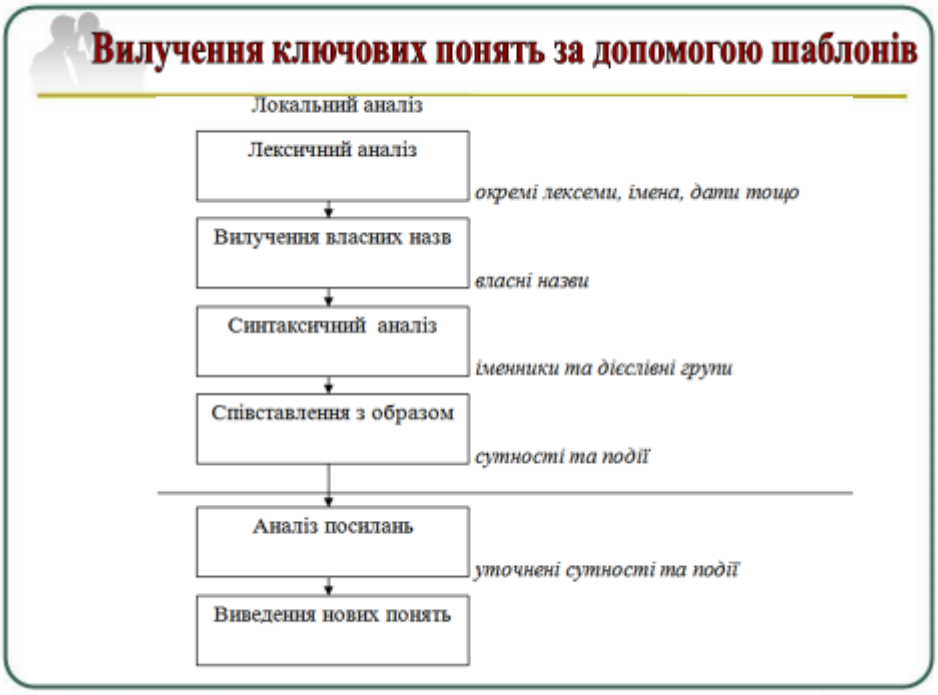

Рисунок Б.9- Слайд №9

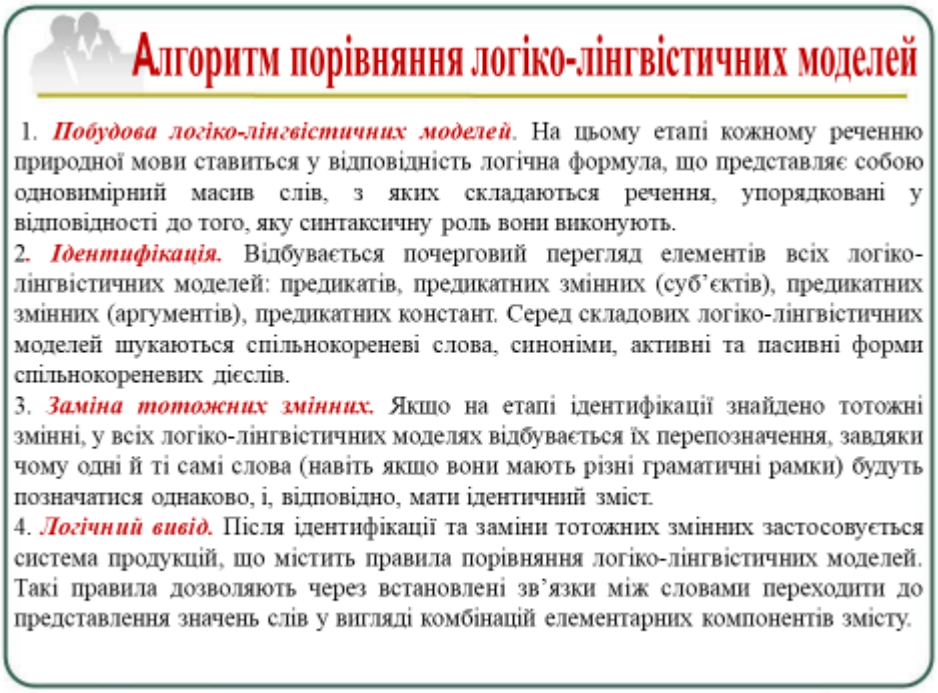

Рисунок Б.10- Слайд №10

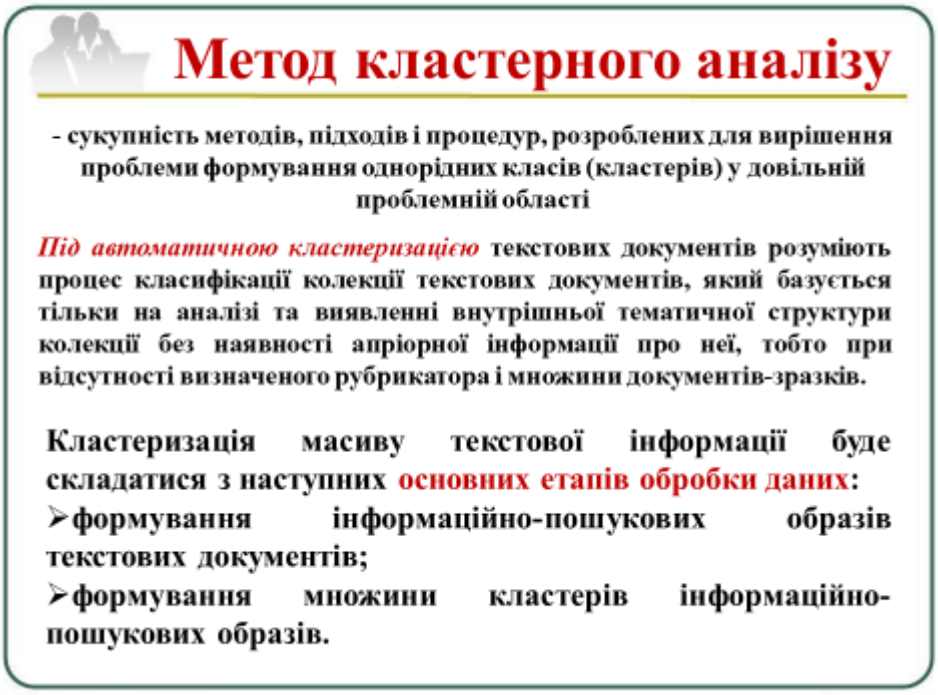

Рисунок Б.11- Слайд №11

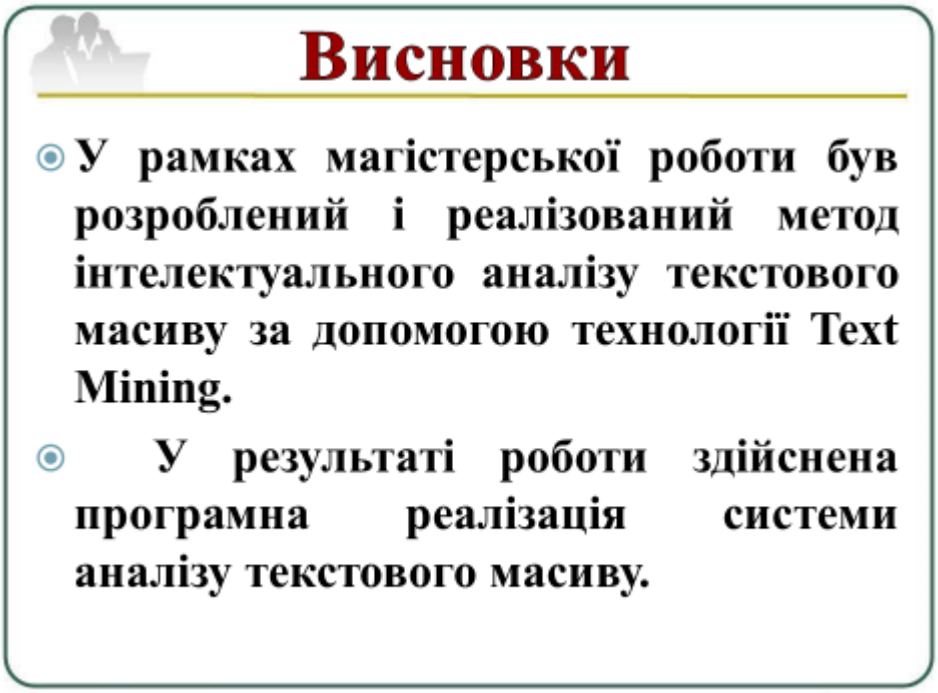

Рисунок Б.12- Слайд №12