Вінницький національний технічний університет
Факультет інформаційних технологій та комп’ютерної інженерії
Кафедра комп’ютерних наук

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему «Інформаційна технологія аналізу англомовного тексту на наявність сталих мовних конструкцій»

Виконала: студентка 2 курсу,
групи 1КН-18 м
спеціальності 122 «Комп'ютерні науки»
Миколюк І. О.
Керівник: к.т.н., проф. кафедри КН
Месюра В. І.
Рецензент: к.т.н., доц. кафедри ПЗ
Ракитянська Г. Б.

Вінниця - 2019 року

ЗАТВЕРДЖУЮ
Завідувач кафедри____КН____
д.т.н., проф.. Яровий А.А.

_____
(підпис)
“____” _____2019 року

ЗАВДАННЯ

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра зі спеціальності: 122 – «Комп'ютерні науки»

08-22.МКР.011.18.000.ПЗ

Магістранта групи 1КН-18м Миколюк Ірини Олександрівни

Тема магістерської кваліфікаційної роботи: «Інформаційна технологія аналізу англомовного тексту на наявність сталих мовних конструкцій»

Вхідні дані: мова програмування – об’єктно-орієнтована; кількість символів у тексті не менше 100; розмір тексту не більше 500 КБ; кількість класів – не менше 3; тестова вибірка.

Короткий зміст частин магістерської кваліфікаційної роботи:

1. Графічна: Структурна схема аналізу текстових документів; алгоритм аналізу логіко-лінгвістичних моделей текстових документів; модель функціонування інформаційної технології; структура інформаційної технології аналізу англомовного тексту на наявність сталих мовних конструкцій; схема загального алгоритму функціонування системи; загальна UML-діаграма класів; приклади роботи програми.
2. Текстова (пояснювальна записка): вступ, аналіз предметної області інформаційної технології аналізу англомовного тексту на наявність сталих мовних конструкцій, розробка моделі інформаційної технології аналізу англомовного тексту на наявність сталих мовних конструкцій, програмна реалізація інформаційної технології аналізу англомовного тексту на наявність сталих мовних конструкцій, економічна частина, висновки, перелік використаних джерел, додатки.

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз сучасного рівня розвитку інформаційних технологій аналізу тексту. Постановка задач дослідження			Аналітичний огляд літературних джерел, задачі досліджень, розділ 1 ПЗ
2	Розробка методу та інформаційної технології аналізу тексту.			Метод, інформаційна технологія, розділ 2
3	Програмна реалізація розробленої інформаційної технології, тестування та оцінка параметрів			Програмне забезпечення, розділ 3
4	Підготовка економічної частини			розділ 4
5	Апробація та/або впровадження результатів дослідження			тези доповідей/акт впровадження
6	Оформлення пояснювальної записки, графічного матеріалу та презентації			Пояснювальна записка, графічний матеріал, презентація

Консультанти з окремих розділів магістерської кваліфікаційної роботи

1. Науковий керівник _____ к. т. н, проф. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)
"____" _____20___ р. Месюра В. І.
ініціали та прізвище

2. Економічна частина _____ к. т. н, доц
(підпис) наук. ступінь, вчене звання (посада)
Бальзан М. В.
ініціали та прізвище
"____" _____20___ р.

Дата попереднього захисту роботи "____" _____20___ р.

Рецензент _____ к. т. н, доц., доц. кафедри ПЗ
(підпис) наук. ступінь, вчене звання (посада)
_____
ініціали та прізвище

Завдання видав
науковий керівник _____ к. т. н, проф., проф. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)
Месюра В. І.
ініціали та прізвище
"____" _____20___ р.

Завдання отримав магістрант _____ _____ І.О. Миколюк
(підпис) ініціали та прізвище
"____" _____20___ р.

АНОТАЦІЯ

В даній магістерській кваліфікаційній роботі реалізовано інформаційну технологію аналізу англомовного тексту на наявність сталих мовних конструкцій. Проведено аналіз сучасних програм-аналогів які використовуються у комп'ютерній лінгвістиці для автоматизації аналізу та обробки текстових документів, обрахунку частотності термів, класифікації текстів та наведено коротку порівняльну характеристику знайдених програм-аналогів.

Проведено обґрунтування вибору методів, які можуть бути використані для розв'язання даної задачі, розроблено математичну модель аналізу тексту.

Розроблено алгоритм аналізу англомовного тексту на наявність сталих мовних конструкцій та відповідне програмне забезпечення мовою Java в середовищі Eclipse. Аналіз роботи програмного забезпечення показав достатньо високу швидкість аналізу англомовного тексту.

ABSTRACT

In this master's qualification work is implemented information technology of analysis of English-language text for the presence of permanent linguistic structures. An analysis of modern analog programs used in computer linguistics to automate the analysis and processing of text documents, calculating the frequency of terms, classification of texts, and gives a brief comparative description of the found analog programs.

The justification of the choice of methods that can be used for solving this problem is made, the mathematical model of text analysis is developed.

An algorithm for analyzing English-language text for the presence of persistent language constructs and corresponding Java language software in Eclipse have been developed. Analysis of the software showed a sufficiently high speed of analysis of English-language text.

ЗМІСТ

ВСТУП

Актуальність теми дослідження. Ідіоми, фразові дієслова та інші сталі конструкції широко використовуються в повсякденній мові. Вони несуть метафоричний сенс, який ускладнює їх осмислення, оскільки їх значення не може бути виведене зі значення їх складових частин. Вони ставлять виклик для таких додатків обробки природних мов, як машинний переклад, пошук інформації та відповіді на запитання, оскільки їх переклад та значення потрібно виводити логічно, а не буквально.

Усі мови світу часто використовують ідіоматичні вирази, більшість з яких мають соціокультурне, історичне чи політичне походження. Хоча багато подібних виразів можна знайти в різних мовах, багато інших не збігаються точно за своїм мовним чи семантичним значенням та вживанням. У той же час ідіоми часто є каменем спотикання для студентів другої / іноземної мови та студентів із загальноосвітніх шкіл.

Присутні у мовній свідомості носіїв англійської мови, сталі вирази використовуються ними у відповідних мовленнєвих ситуаціях автоматично. Для розуміння словникового складу мов дуже важливо усвідомити головну відмінність різних видів лексики; перш за все відокремлюються два полюси: це терміни та ідіоми. Існує багато тлумачних, а також двомовних фразеологічних словників. У більшості з них фіксуються переважно одиниці, які обслуговують номінативно-класифікаційну діяльність людини і мають нефразеологічні еквіваленти.

Необхідність володіння англійською мовою на сьогодні вважається великою, адже найбільше друкованої продукції видають саме англійською мовою. Англійська мова – це офіційна мова міжнародного бізнесу та торгівлі, Інтернету і техніки, науки і мистецтв. 80% ділового мовного простору займає саме вона. Англійська мова використовується при заповненні анкет, складання резюме, у діловому та приватному листуванні.

Для української і російської мови не властиво використання таких конструкцій, як фразові дієслова. На сьогоднішній день вживання фразових дієслів як в усній, так і в письмовій англійській мові стає все частішим явищем. Фразові дієслова є невід'ємною частиною активної комунікативної сфери спілкування англійською мовою і широко використовується не тільки в розмовному стилі мови, а й в інших видах вербальної комунікації.

Аналіз контенту зараз використовується у багатьох сферах, починаючи від маркетингових та медіа-досліджень до літератури та риторики, етнографії та культурології, гендерних та вікових питань, соціології та політології, психології та когнітивної науки та багатьох інших галузей дослідження.

Труднощі семантичного аналізу природно-мовних конструкцій дають поштовх для дослідників звертати увагу на екстралінгвістичні знання у вигляді різного роду когнітивних моделей.

Задачі обробки текстів виникли практично відразу після появи обчислювальної техніки. Незважаючи на півстолітню історію досліджень в області штучного інтелекту, накопичений досвід обчислювальної лінгвістики, величезний стрибок у розвитку IT і суміжних дисциплін, задовільного вирішення більшості практичних завдань обробки тексту поки не знайдено.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп’ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою дослідження магістерської кваліфікаційної роботи є підвищення швидкості аналізу текстової інформації.

Для досягнення наведеної мети були поставлені та вирішені наступні задачі:

1) Розглянути та проаналізувати методи та технології розв’язання задачі аналізу тексту;

2) дослідити перелік необхідних функцій, які повинна містити інформаційна технологія;
3) запропонувати математичну модель інформаційної технології аналізу англомовного тексту на наявність сталих мовних конструкцій;
4) виконати програмну реалізацію запропонованої інформаційної технології;
5) провести тестування програмного продукту та виконати аналіз отриманих результатів.

Об'єкт дослідження - процес аналізу англомовного тексту на наявність сталих мовних конструкцій.

Предмет дослідження - методи аналізу тексту.

Методи дослідження. У роботі використані наступні методи наукових досліджень: системного аналізу для аналізу структури інформаційної системи, метод аналізу логіко-лінгвістичних моделей речень природної мови, метод пошуку текстових збігів у реченнях природної мови довільної складності, алгоритм Байєсівського класифікатора та алгоритм підрахунку TF-індексу для реалізації модуля аналізу англомовного тексту на наявність сталих мовних конструкцій, об'єктно-орієнтованого програмування для автоматизації розрахунків.

Наукова новизна одержаних результатів полягає в наступному:

− вперше запропоновано інформаційну технологію аналізу англомовного тексту на наявність сталих мовних конструкцій, яку засновано на сумісному використанні методу Байєсівського класифікатора та алгоритм підрахунку TF-індексу в поєднанні з алгоритмами логіко-лінгвістичного моделювання для аналізу текстового документу, що забезпечмло підвищення швидкості аналізу англомовного тексту.

− вдосконалено модель аналізу англомовного тексту на наявність сталих мовних конструкцій шляхом симбіозу механізмів розробки новітніх інформаційних технологій та точних методів комп'ютерної лінгвістики, що забезпечує підвищення швидкості отримання вихідних даних.

Практичне значення одержаних результатів полягає у наступному:

1. Розроблено алгоритм аналізу англомовного тексту, який використовує поєднання алгоритмів логіко-лінгвістичного моделювання та методів Data Mining.

2. Розроблено програмне забезпечення аналізу англомовного тексту на наявність сталих мовних конструкцій.

Розроблені алгоритми можуть бути впроваджені в начальний процес як основа лекцій на тему «Методи обробки текстів природною мовою» дисципліни «Інтелектуальний аналіз даних».

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими, та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок магістранта. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно.

Апробація результатів роботи. Результати роботи були апробовані на всеукраїнській науково-практичній інтернет-конференції студентів, аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН 2019)» (м. Вінниця, Україна, 2019 р.) [1] та XLVII Науково-технічній конференції факультету інформаційних технологій та комп’ютерної інженерії (2018) [2].

Публікації. За результатами магістерської кваліфікаційної роботи опубліковано тези доповіді у збірнику праць всеукраїнської науково-практичної інтернет-конференції [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ АНГЛОМОВНОГО ТЕКСТУ НА НАЯВНІСТЬ СТАЛИХ МОВНИХ КОНСТРУКЦІЙ

1.1 Постановка задачі

Аналіз англомовного тексту на наявность сталих мовних конструкцій є важливою проблемою для обробки тексту. Існує багато конструкцій, які використовуються в тій чи іншій ситуації і можуть показати, наскільки добре ними володіє автор, чи правильно він використовує їх в тексті, і може визначати стиль тексту, використовуваний автором. У цій роботі необхідно реалізувати інформаційну технологію аналізу англомовного тексту на наявність сталих мовних конструкцій. Тобто необхідно вирішити проблему аналізу тексту програмно, що визначить насиченість тексту стійкими конструкціями англійської мови.

Розробляючи програму, необхідно звернути увагу на функції, які вона повинна виконувати, а саме: точність та швидкість аналізу.

Інформаційна технологія аналізу англомовного тексту на наявність сталих мовних конструкцій повинна отримувати на вході англомовний текст, а також аналізувати наявність ідіом, фразових дієслів, термінів на інших стійких конструкцій у тексті, перетворюючи текст у набір векторів за допомогою TF - індексів, щоб класифікувати текст за його тематикою. Після аналізу інформаційна технологія повинна вивести рівень насиченості тексту сталими конструкціями, кількість конструкцій кожного типу та стиль тексту.

Це має полегшити процес вивчення англійської мови для студентів, які займаються іноземними мовами та вивчення сталих мовних конструкцій, філологів та викладачів англійської мови або просто людей, що бажають самостійно заглибитися в іноземну мову. Інформаційна технологія - це інструмент, який забезпечить зручний та ефективний аналіз англомовного тексту

на наявність сталих мовних конструкцій на відміну від неякісних або важко зрозумілих аналогів.

Проблема полягає у відсутності простого, безкоштовного і водночас багатофункціонального та продуктивного інструменту для обробки та аналізу тексту. Це питання стосується людей, які працюють у галузі філології або просто тих, хто хоче дізнатися більше про англійську мову. Вирішення проблеми підвищить популярність аналізу тексту, створить ефективний та зручний інструмент для аналізу та класифікації текстів.

Тому для вирішення проблеми аналізу англомовного тексту на наявність сталих мовних конструкцій необхідно вирішити такі основні завдання:

1. Обґрунтувати вибір методу аналізу тексту.
2. Розробити математичну модель.
4. Спроектувати структуру інформаційної технології.
5. Розробити базу даних англомовних сталих конструкцій.
6. Програмно реалізувати інформаційну технологію.
7. Провести тестування інформаційної технології.

Отже, потрібно розробити інформаційну технологію, в якій буде реалізовано алгоритм аналізу тексту конструкціями, які зберігаються в базі даних, а також визначити стиль аналізованого тексту.

1.2 Аналіз сталих мовних конструкцій в англомовних текстах

У мові поширено використовуються словосполучення, які існують як готові словесні блоки, а не утворюються у процесі розмови. Це так звані фразеологізми та сталі конструкції. Більшість з них стилістично значимі, надають висловлюванню емоційно-експресивного відтінку. Фразеологічні одиниці вирізняються різноманітністю граматичних моделей та ознак, складною лексикограматичною і смисловою структурою. Цей прошарок лексики займає важливе місце в загальнонаціональному словникові будьякої мови, швидко розвивається, створюючи для перекладача низку труднощів. Тому проблема

відтворення фразеологізмів на мові перекладу залишається досі в полі зору лінгвістів та вчених з різних сфер науки, а саме істориків мови, філософів, логіків та країнознавців.

Науково-технічні тексти розкривають низку граматичних нюансів. Найбільш типовою лексичною ознакою науково-технічної літератури є багатство тексту термінами та термінологічними словосполученнями, а також наявність лексичних конструкцій та скорочень. Щодо синтаксичної структури, англійські тексти науково-технічного змісту відрізняються своєю складністю, оскільки використовують певні конструкції. Вони містять багато дієприкметникових, інфінітивних та германських конструкцій, які інколи ускладнюють розуміння тексту та змушують перекладачів виконувати додаткові завдання. Різноманітність термінів є однією з визначальних характеристик науково-технічного тексту. У науково-технічному тексті частка термінологічної лексики становить не більше 25%, а основна частина лексики - загальнонаукові, загальнотехнічні та загальноприйняті слова.

В англомовних текстах важливе місце займають різноманітні види скорочень. Так як вони функціонують незалежно, вони фіксуються в лексикографічних джерелах і часто стають більш відомими, ніж їх джерела (радіолокатор, лазер), їх можна вважати лексичними одиницями науково-технічної мови. У науково-технічному мовленні використовуються ті самі синтаксичні структури та морфологічні форми, як і в інших функціональних стилях. Однак ряд граматичних явищ відзначається в цьому стилі частіше, ніж в інших, одні явища, навпаки, зустрічаються в ньому порівняно рідко, інші використовуються лише з характерною лексичною "сферою".

У науковій літературі майже немає метафор, метонімічних транспозицій та інших стилістичних фігур, широко використовуваних у художній літературі. Кожна мовна конструкція, в тому числі фразеологічна одиниця, ідіома, найчастіше вживається в художньому тексті. Саме в цьому функціональному стилі порівняно з науковим, діловим, розмовним, публіцистичним проявляються усі потенційні можливості фразеологізму. Слід зазначити, що частотність ідіом

та особливості їхнього використання значною мірою залежать від художнього тексту. Тож основне завдання – вміти розпізнати ідіому в тексті, відрізнити стійку сполуку від нестійкої. Маючи справу з фразеологічними одиницями, потрібно вміти аналізувати стилістичні та культурно-історичні аспекти певної мови.

1.3 Дослідження понять та принципів аналізу тексту

Аналіз тексту – процес отримання високоякісної інформації з тексту природною мовою. Як правило, використовується шаблонне статистичне навчання: вхідний текст розділяється на шаблони, після чого дані обробляються.

Текст – одна з найбільш «насичених» форм вихідних даних, оскільки не має недоліків штучно створеної та формалізованої «реальності» опитувальників з обмеженим вибором відповідей. Водночас текстові дані – одні найбільш незручних для аналізу, оскільки потребують великих затрат часу, якщо йдеться про репрезентативні дослідження [4].

Ручний аналіз неефективний для обробки великих обсягів тексту, тому що він обмежений швидкістю, похибками та помилками, зумовленими людським фактором. Статистичні методи потребують вирішення цих проблем, тому що текстові документи містять лінгвістичні та семантичні взаємозв'язки, які потрібно приймати до уваги. Вилучення знань з неструктурованих даних є головною технічною проблемою.

Одна з найбільш вивчених задач аналізу тексту – контекстно-вільний пошук текстової інформації. Її сенс полягає в наступному: є великий набір файлів, що містять тексти природною мовою, і заданий деякий рядок. Необхідно знайти всі файли, в яких він зустрічається або наявна схожа текстова інформація. У переважній більшості випадків необхідний саме «нечіткий» пошук (тобто пошук за змістом слова з урахуванням специфіки природної мови). Більшість існуючих систем ґрунтуються виключно на морфологічному аналізі слів і не задіють більш складних схем аналізу.

Найбільш важливе завдання – вилучення інформації з текстів і представлення її у вигляді формальної системи знань (зокрема, у вигляді семантичної мережі). Виконано ряд експериментальних розробок у даному напрямку, орієнтованих на конкретні предметні області, проте комерційних програмних продуктів немає [5].

Комп'ютерний аналіз тексту природною мовою активно розвивається в останні роки багатьма дослідниками. Наявні на сьогоднішній день обчислювальні потужності дозволяють використовувати для обробки великих масивів документів широкий діапазон математичних методів, які доцільно використовувати у вирішенні завдань пошуку, кластерного аналізу, класифікації та виявлення прихованих закономірностей в даних.

Математичні методи були введені в обробку тексту, що відбувається в той момент, коли недостатньо представлена лінгвістична складова алгоритмів, що не дозволяє досягти достатньої якості роботи прикладних систем [6]. Складність практичної реалізації прийнятного аналізатора тексту обумовлена наявністю тісного зв'язку між синтаксисом і мовною семантикою. Тому для вирішення проблем (званих синтаксичної омонімією) необхідно створення спеціального тлумачно-комбінаторного словника, що включає в себе синтаксичну і семантичну інформацію про сполучуваність слів [5].

Обробка природної мови – загальний напрямок штучного інтелекту і математичної лінгвістики. Він вивчає проблеми комп'ютерного аналізу і синтезу природних мов. Стосовно штучного інтелекту аналіз означає розуміння мови, а синтез – генерацію грамотного тексту. Вирішення цих проблем буде означати створення більш зручної форми взаємодії комп'ютера і людини [6].

Розуміння тексту трактується як процес співвіднесення мовного опису з позамовною дійсністю. Основна наукова проблема, з якою стикаються розробники при створенні блоку знань про предметну область це проблема характеру подання (репрезентації) знань про розуміння світу в пам'яті.

Існує два підходи до вирішення завдання розуміння тексту [7]:

- моделювання людських механізмів розуміння природно мовних текстів;
- створення технічних систем, що реалізують функцію розуміння відмінним від людини способом при порівнянності результатів розуміння системи і людини для заданої мети розуміння.

В якості робочого обраний другий підхід.

Модель машинного розуміння тексту заснована на реалізації наступних процедур:

- формування базису розуміння (зразка) шляхом виокремлення з знань про предметну область фрагмента з урахуванням поточної мети розуміння;

- виділення словникових виразів (слів, словосполучень, пропозицій), що позначають виділені елементи предметної області;

- співвіднесення виділених словникових виразів з іменами елементів предметної області.

(Результат розуміння може бути представлений або у вигляді фрагмента тексту, що містить інформацію, відповідну мети кінцевого користувача, або у вигляді пар: ім'я елемента предметної області словниковий вираз).

Для того, щоб спростити процедуру аналізу взаємозв’язку термінів, пропонується виділяти один термін як «домінанту», а інший, який буде траплятися разом з ним, умовно називати «супровідним» (рис. 1.1).

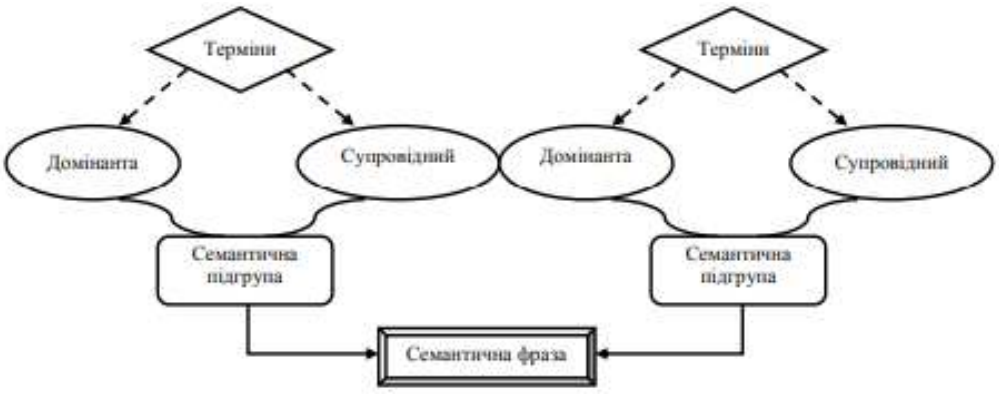

Рисунок 1.1 – Схематична модель термінологічного процесу аналізу тексту

При цьому не слід забувати, що один і той самий термін може виступати як у ролі «домінанти», так і «супровідним». Після того, як зв’язки між окремими парами термінів будуть установлені, слова, які тісно пов’язані один з одним за сенсом, можна об’єднати у семантичні підгрупи тощо доки у підгрупі не з’являться цілі змістові фрази.

Завдяки контент-аналізу можна отримати достатньо об’єктивний результат та зробити зміст тексту вимірюваним і придатним для точного обчислення.

1.4 Аналіз функціонування систем обробки текстової інформації

Однією з основних особливостей розвитку суспільства в кінці XX - на початку XX століття є різке зростання соціальної значущості інформації, внаслідок чого збільшуються інформаційні потоки [6]. Динамічний характер попиту на інформацію призводить до необхідності постійно отримувати оперативну інформацію для своєчасного прийняття рішень [7].

Тому першочерговим завданням обробки великої кількості інформації є аналіз даних, а тому системи управління базами даних повинні підтримувати повнотекстові методи пошуку, які дозволять швидко знайти потрібну інформацію. Аналіз даних передбачає побудову найбільш поширених математичних методів та алгоритмів вилучення знань; фільтрація, перетворення та моделювання даних для отримання корисної інформації для прийняття рішень [8]. Здебільшого дані подаються як текстова інформація, яка в різних формах складає значну кількість інформаційних ресурсів. Тому створення та розвиток систем обробки тексту представляє інтерес на всіх етапах розвитку інформаційних технологій [9].

Системи аналітичної обробки текстової інформації використовуються в різних галузях людської діяльності: торгівлі, банківській справі, у галузі прогнозування, безпеки програмного забезпечення, сертифікації тощо [10]. Кожна з цих систем розробляється на основі специфічної технології аналізу

природних мов, яка передбачає перехід від деякого вираження природної мови до певного внутрішнього представлення [11].

Обробка природних мов лежить в основі сучасного програмного забезпечення, яке розпізнає та інтерпретує природну мову. Задовільні результати вже отримані при здійсненні морфологічного, синтаксичного та частково семантичного аналізу, але розуміння змісту текстової інформації залишається проблемою [12]. Однією з причин автоматизації обробки текстів є час, який потрібен для написання програм. Це може бути значно зменшено, якби комп'ютери змогли зрозуміти прості твердження на природній мові [13].Основний недолік систем обробки текстової інфомрації – робота зі змістом окремих слів та словосполучень, рідше – із зв'язками між ключовими словами, але структура речення, а тим більше електронного документу в цілому не аналізується.

Сучасні автоматизовані системи, що аналізують смислову структуру тексту на рівні, вищому за речення, знаходяться на експериментальній стадії [14]. На сьогодні жодна з теорій не може претендувати на повну, хоча найсучасніші з них досягли задовільних теоретичних результатів [15].

Проблеми аналізу тексту вирішуються сучасними системами аналітичної обробки текстової інформації за допомогою різних підходів залежно від сфери застосування [16]. Тому при дослідженні існуючих інформаційних систем для обробки текстової інформації враховуються загальні показники систем, способи обміну даними, обсяг та завдання, що вирішуються.

1.5 Подання тексту та відбір ознак

Хоча вибір функції часто використовується в інших завданнях класифікації, він особливо важливий у проблемі класифікації тексту через його високу розмірність (велика кількість ознак) та наявність невідповідних (шумових) особливостей. У більшості випадків подання тексту здійснюється одним із двох способів. Перший - це документ як сукупність слів, у якому

документ порівнюється зі словами та частотою їх виникнення в ньому. Тобто подання не залежить від порядку слів, у якому вони фігурують у тексті. Другий спосіб - подати текст, власне, як набір рядків, у яких документ є послідовним словом. Більшість алгоритмів класифікації тексту використовують перше представлення через його простоту та зручність для завдань класифікації.

Найпопулярніші способи вибору функцій - це видалення стоп-слів і фраз. Коли ми видаляємо стоп-слова, ми визначаємо загальні слова для документів, які не є конкретними або загальними для різних класів.

Утворення різних форм одного слова об’єднуються в одне слово (термін). Наприклад, скарб, тому слова різного роду / форми / часу / відмінка поєднуються. Методи вибору декількох функцій були переглянуті та експериментально перевірені на ефективність [5].

Існує кілька підходів до вибору найкращих інформаційних функцій. У загальній класифікації [17] є три основні категорії методів вибору найкращих інформаційних ознак: методи фільтра (Фільтри), Методи обертання та Вбудовані.

1.6 Дослідження методів та інформаційних технологій для аналізу тексту

Вибираючи конкретний алгоритм класифікації, слід враховувати особливості кожного з них. Питання визначення набору класифікаційних ознак, їх кількості та методів обчислення ваг залишається невирішеним. В алгоритмах глибокого навчання точність класифікації сильно залежить від наявності навчальної вибірки відповідного розміру. Підготовка такого зразка - дуже трудомісткий процес. Проблема вибору параметрів деяких алгоритмів на етапі навчання залишається відкритою.

Далі докладно описується кожен з етапів, описуються різні алгоритми побудови класифікаторів, їх експерименти та результати цих експериментів.

Опис методів класифікації

На малюнку 1.2 представлена загальна схема процесу класифікації. Розглянемо кожен з його етапів.

Переробка та індексація документів. Попередня обробка включає токенізацію, видалення функціональних слів (семантично нейтральних слів, таких як союзи, прийменники, статті тощо). Далі йде морфологічний аналіз (розмітка за частинами мови та стемматизація). Це може значно зменшити розмірність простору. Як результат, усі значущі слова, що містяться в документі, виглядають як функції документа.

Індексація документів - це побудова деякої числової моделі тексту, яка переводить текст у зручну для подальшої обробки презентації.

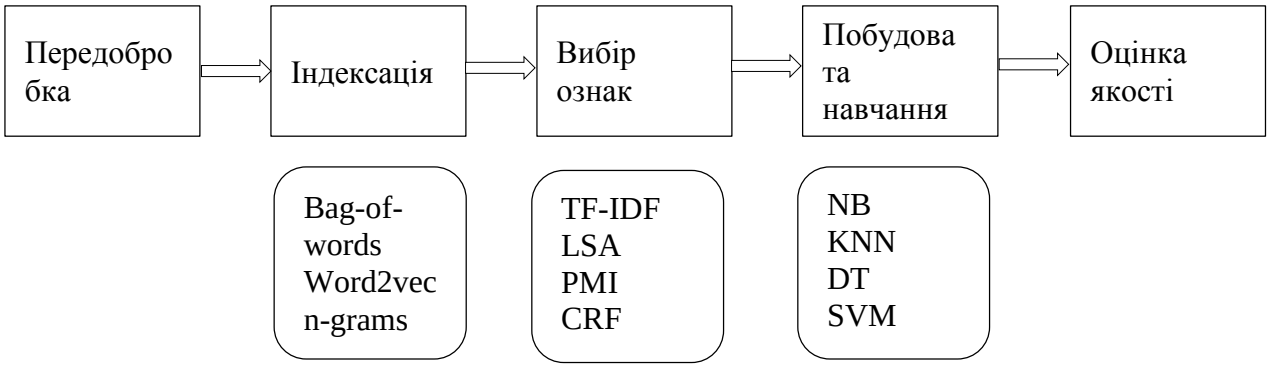

Рисунок 1.2 – Етапи процесу автоматичної класифікації текстів

Наприклад, модель «мішок слів» дозволяє представити документ у вигляді багатовимірного вектора слів та їх ваги в документі [2]. Іншими словами, кожен документ - це вектор у багатовимірному просторі, координати якого відповідають номерам слів, а координати - значенням ваг.

Ще одна поширена модель індексації - Word2vec [5]. Він представляє кожне слово у вигляді вектора, що містить інформацію про контекстуальні (споріднені) слова.

Інша модель індексації заснована на обліку n-грамів [8], тобто послідовностей із суміжних символів.

Очевидно, той самий метод індексації слід використовувати для навчальних та тестових документів.

Вимір простору функції. Комплексна обчислюваність різних методів класифікації безпосередньо залежить від виміру простору ознак. Тому для ефективної роботи класифікатора часто вдаються до зменшення кількості використаних знаків (термінів).

Зменшуючи розмірність терміна простір, ефект перенавчання може бути зменшений - явище, при якому класифікатор орієнтується на випадкові чи помилкові характеристики навчальних даних, а не на важливі та змістовні. Класифікатор перепідготовки працює добре на засвоєних ним екземплярах, і гірше на даних тесту. Щоб уникнути перепідготовки, кількість навчальних прикладів повинна бути пропорційною кількості використовуваних термінів. В деяких випадках зменшення розмірності простору функції в 10 разів (цілих 100) може призвести лише до незначної деградації класифікатора.

Існує кілька способів визначення ваги особливостей документа. Найбільш поширеним є обчислення функції TF-IDF [8, 9, 10]. Його основна ідея полягає в тому, що слова з високою частотою в рамках конкретного документа і з низькою частотою використання в інших документах набувають більшої ваги.

Розглянемо деякі методи, які використовуються для класифікації даних та документів.

Метод Байєса (Naive Bayes, NB) відноситься до імовірнісних методів класифікації.

До переваг методу відносять:

- висока швидкість роботи;
- підтримка поступового навчання;
- відносно проста програмна реалізація алгоритму;
- легко інтерпретувати результати алгоритму.

Недоліки методу: відносно низька якість класифікації та неможливість врахувати залежність результату класифікації від поєднання ознак.

Метод k Найближчих сусідів (KNN) k відноситься до методів метричної класифікації. Щоб знайти категорію, що відповідає документу d, класифікатор порівнює d з усіма документами в навчальному зразку L, тобто для кожного dz |

L обчислюється відстань r (dz, d). Далі з навчального зразка k вибираються документи, найближчі до d. Відповідно до методу k найближчих сусідів, документ d вважається належним до класу, який найбільш поширений серед сусідів цього документа, тобто для кожного класу сі розраховується функція ранжування.

Переваги методу:

- можливість оновлення навчальної вибірки без перенавчання класифікатора;

- стійкість алгоритму до аномальних викидів у вихідних даних;

- відносно проста програмна реалізація алгоритму;

- легка інтерпретація результатів алгоритму;

- хороша підготовка у випадку лінійно нероздільних зразків.

Недоліки методу:

- репрезентативність набору даних, що використовується для алгоритму;

- висока залежність результатів класифікації від обраної метрики;

- велика тривалість роботи через необхідність повного пошуку вибірки навчання;

- неможливість вирішити масштабні завдання за кількістю класів та документів.

Дерева рішень, DT посилається на методи логічної класифікації.

Деревом рішень називається ациклічний графік, відповідно до якого класифікуються об'єкти (у нашому випадку текстові документи), описані набором ознак. Кожен вузол дерева містить умову розгалуження за однією з особливостей. Кожен вузол має стільки гілок, скільки значення вибраного атрибута. Процес оцінки коригує послідовні переходи від одного вузла до іншого відповідно до значень особливостей об’єкта. Класифікація вважається завершеною, коли досягнуто одного з листя (кінцевих вузлів) дерева. Значення цієї літери визначатиме клас, до якого належить об’єкт. На практиці зазвичай використовуються двійкові дерева рішень, в яких рішення про перехід через краї просто перевіряється наявністю функції в документі. Якщо значення атрибута

менше певного значення, вибирається одна гілка, якщо більша або дорівнює іншій.

До переваг методу можна віднести порівняно просту програмну реалізацію алгоритму та просту інтерпретацію результатів алгоритму.

Недоліком цього методу є нестабільність алгоритму щодо викидів з вихідних даних та велика кількість даних для отримання точних результатів [10].

Тож існує багато методів та підходів до аналізу тексту. Переваги кожного підходу залежать від типів та обсягу аналізованих текстів та питань, які повинен вирішити аналітик. Навіть якщо існує єдиний підхід, можливі варіанти його застосування. Не існує єдиного методу, який найбільше підходить для всіх типів текстуального аналізу. Однак для обчислення частоти вибирається модель, заснована на TF-IDF.

1.7 Аналіз програм-аналогів

Існує багато програм для обробки та аналізу тексту на природній мові, які стосуються лінгвістичного та семантичного аналізу тексту, кодування, кластеризації та категоризації документів, виявлення об'єктів, створення моделі, автоматичної або ручної систематики, онлайнової аналітичної обробки даних та результатів інтерактивного графіку.

Особливістю програм якісного аналізу контенту є те, що вони розроблені не лише для аналізу тексту, але і для фрагментів кодування аудіо та відео, вони також допомагають розробляти теорії та надають можливість перевірити ці теорії.

Давайте розглянемо деякі з цих програм докладніше.

Textanz - це унікальний та необхідний інструмент для всіх, хто бере участь у написанні текстів. Ця програма аналізує всі типи текстового змісту та надає вам список чи словник слів, фраз та граматичних форм, а також частоту їх використання в тексті. Інформація, отримана в результаті аналізу Textanz, дозволяє перевірити надмірне використання або повторення слів або фраз у будь-

якому документі. Textanz забезпечує важливу функцію редагування, яка відсутня у більшості стандартних програм перевірки граматики та програм редагування [21]. На рисунку 1.3 показаний приклад програми

Таблиця 1.1 – Програми кількісного контент-аналізу

Програма	Призначення	Статус	Основні характеристики	Можливості та якість роботи
Textanz	Аналізує текст, надає список або словник слів, фраз і граматичних форм	Безкоштовна	Дозволяє перевірити надмірне використання або повторення слів та фраз в будь-якому документі, є можливість редагування тексту	Працює з документами формату RTF, MS Office, Open Office, HTML, XML, PDF і Дозволяє одночасний аналіз декількох документів
TextQuest	Аналізує текст	Платна але надається демо-версія з обмеженими можливос-тями	Безкоштовна версія має обмеження в 100 одиниць тексту. Надає можли- вість визначати частоту.	Використовує таблиці сортування слів, в залежності від довжини рядка символів.

Таблиця 1.2 – Програми якісного контент-аналізу

Програма	Призначення	Статус	Основні характеристики	Можливості та якість роботи
Kwali-tan	Призначена для аналізу текстів, зображень, аудіо та відео фраг- ментів кодування.	Платна, але надається демоверсія з обмеже-ними можливостя ми на невизначени й період.	Ефективно зберігає дані і має ряд функцій для якісного аналізу матеріалів, таких як кодування, отримання та класифікації кодів.	Більшість коду-вання здійс- нюється вручну, але Kwalitan також має інструмент для присвоєння кодів авто- матично.
MAX-QDA	Якісний аналіз даних, систе-матична оцінка та інтерпрета-ція текстів	Платна але надається демо-вер- сія з обме-женим часом користування строком на 30 днів.	Зручна в користуванні, є потужним інструментом для розвитку теорій та їх перевірки	Має великі і диферентційовані функції з хорошою візуалі-зацією, доступні для процесу кодування.

Слова та фразові частоти є найбільш обговорюваними сьогодні проблемами, і багато постачальників програмного забезпечення пропонують рішення в Інтернеті та офлайн. Можна відзначити наступні переваги Textanz:

Багато різновидів частот. Крім окремих слів, Textanz вміє обчислювати частоту словосполучень та словоформ. Програма оцінює фразу незалежно від пунктуації чи інтервалів. Через це однакові успадковані слова оцінюються як однакові фрази, незалежно від лінійних каналів чи різних місць для ком. Під словоформами ми розуміємо будь-який із наступної кількості символів у словах - від одного символу до всього слова. Цей режим рекомендується досвідченим користувачам Textanz і дозволяє виявляти слово, яке часто використовується, навіть якщо воно перетворюється в різні форми в тексті, такі як суфікс та / або префікс.

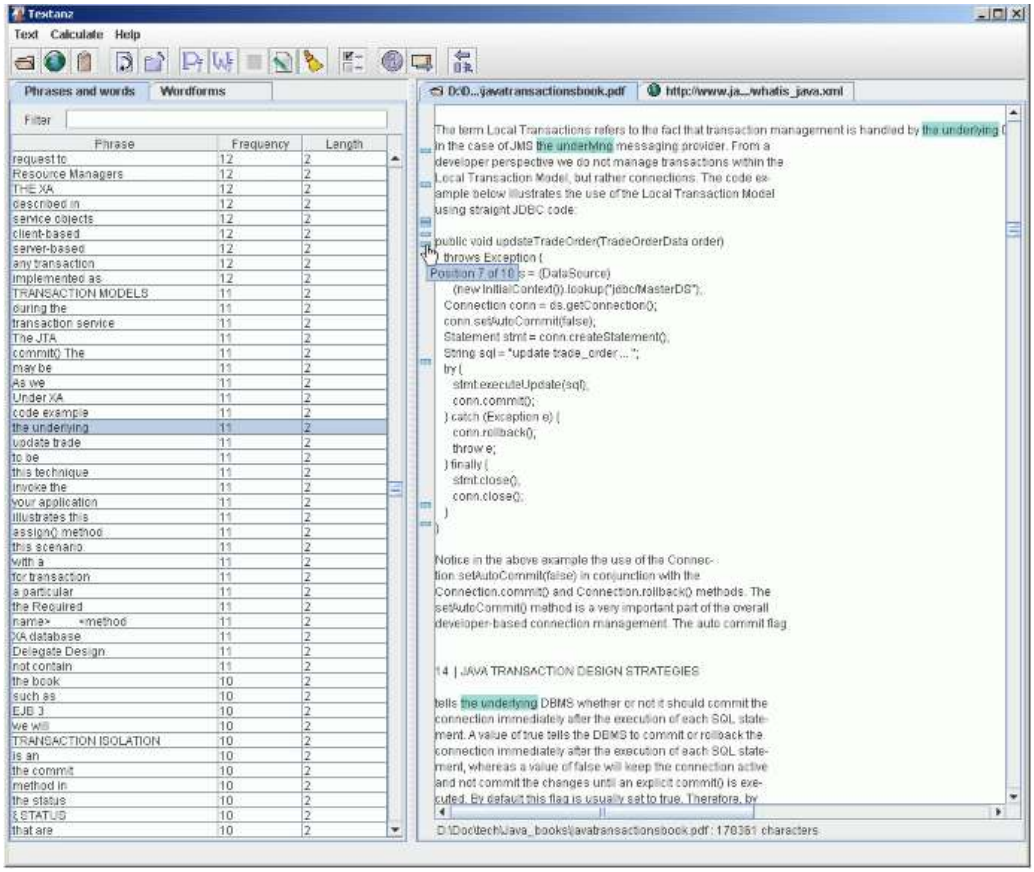

Рисунок 1.3 – приклад використання програми Textanz

Потужні можливості вилучення тексту. Використовуючи аналізаторний пакет Apache Tika, Textanz може витягувати текст з усіх відомих форматів документів, таких як RTF, MS Office, Open Office, HTML, XML, PDF тощо. Ви можете завантажити файл із локальної файлової системи в Textanz та надати URL-адресу для завантаження веб-сторінки та видалення тексту.

Потім Textanz обчислює частоту слів і фраз для всіх текстів. Це корисно, наприклад, коли потрібно виконати пакетну обробку і знайти випадки перехресного кодування між двома або більше документами.

Різне представлення результатів. Після сканування тексту Textanz надає результати у вигляді відсортованого списку. Ви можете здійснити двонаправлене сортування за будь-яким стовпцем (маркери, довжини, частоти) та фільтруванням. Можна також вибрати всі зустрічі слова або фрази в початковому тексті, щоб вибрати один або кілька рядків [22].

HAMLET - одна з найбільш підходящих аналогових програм для аналізу англомовного тексту для стійких мовних конструкцій (рис. 1.4). Основна мета програми - пошук текстових файлів слів або категорій у заданому списку слів та обчислення загальних частот у будь-якій заданій контекстній одиниці в межах речень або у вигляді фраз у визначених інтервалах слів.

Програмне забезпечення використовується як для вимірювання емпіричних властивостей тексту, так і для візуалізації отриманих даних. Частотні слова частоти або загальні частоти для пар слів представлені у вигляді виділених контекстних одиниць, а відповідні стандартизовані загальні частоти організовані в матрицю подібності, яка може бути представлена як комбінація аналізу кластерів та багатовимірного масштабування для виявлення значущого слова асоціації. Крім того, програма дозволяє порівняти результати використання багатовимірного масштабування з матрицями загальних частот, раніше отриманих з ряду текстів.

Основна ідея HAMLET II полягає в тому, щоб шукати текстові файли для слів або категорій у певному списку словників та підраховувати їх спільні

частоти в межах будь-якого конкретного контекстного підрозділу в рамках пропозицій або як колакації в межах заданого проміжку слів [23].

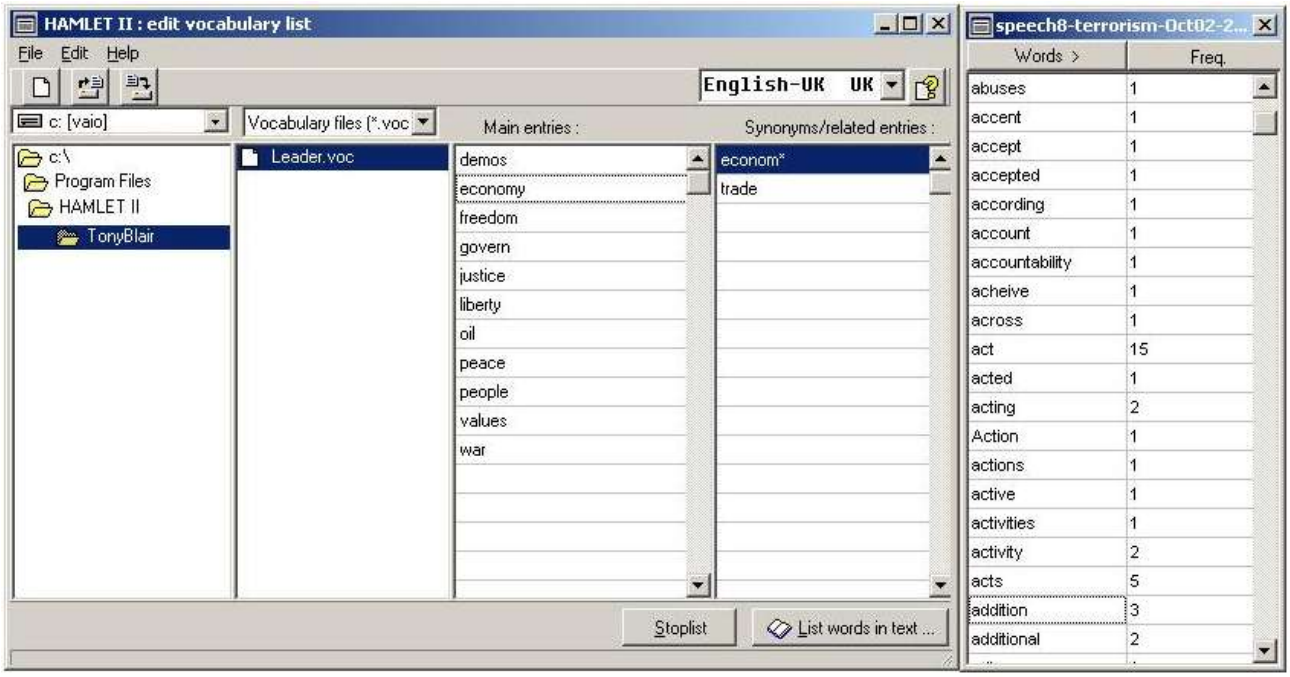

Рисунок 1.4 – приклад використання програми HAMLET

Проаналізувавши дані програми можна зробити висновок, що для аналізу англомовного тексту на наявність сталих мовних конструкцій спеціалізованої програми немає, проте програми, які можуть виконувати такі функції мають певні недоліки. Точність таких програм, зазвичай, підвищується за рахунок збільшення слів у словнику. Проте через великий об’єм словників зменшується швидкість аналізу.

1.8 Висновок

Досліджено предметну область аналізу англомовного тексту на наявність сталих мовних конструкцій, що показало актуальність проблеми, яка полягає у потребі простого, багатофункціонального інструменту для обробки тексту, який забезпечить зручний та ефективний аналіз.

Проведений аналіз сталих мовних конструкцій в англомовних текстах надав можливість визначити принципи подання тексту та відбору ознак, які відрізняються від відомих поданням тексту як набір функцій, що дозволило визначити та видалити слова, які є загальними для різних класів та виділити необхідні терміни.

У результаті дослідження методів та інформаційних технологій для аналізу тексту обрано метод Байєса, який у порівнянні з широко розповсюдженими методами, як дерев рішень, k-найближчих сусідів, забезпечує високу швидкість роботи, підтримку поступовогно навчання та відносно просту програмну реалізацію алгоритму, що робить доцільним його використання як основа аналізу.

Проведений аналіз сучасних програм, що використовуються для аналізу текстової інформації, обчислення частоти та термінів слова в документах, класифікації інформації тощо виявив, що вони не повністю відповідають поставленій задачі і мають суттєві недоліки. Вибрано аналог (програма HAMLET).

Аналіз існуючих систем показав, що на даний час подібні програми для аналізу англомовного тексту на наявність сталих мовних конструкцій не забезпечують потрібну швидкість та високу достовірність.

2. РОЗРОБКА МОДЕЛІ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТЕКСТУ НА НАЯВНІСТЬ СТАЛИХ МОВНИХ КОНСТРУКЦІЙ

2.1 Дослідження можливостей методів аналізу англомовного тексту на наявність сталих мовних конструкцій

Розглянемо методи вирішення задачі аналізу тексту в обраному аналозі - програмі HAMLET [23].

Перевага вимірювання емпіричних властивостей текстів поєднується з особливостями графічного візуалізації HAMLET. Якісний та кількісний аналіз є невід'ємною частиною дизайну HAMLET. Перевага полягає в тому, що HAMLET II забезпечує максимальну прозорість процесів, пов'язаних з єдиним, зручним для користувача інтерфейсом, залишаючи користувачеві повний контроль.

Індивідуальні частотні слова (f_i), частотні співвідношення (f_{ij}) для пар слова (i, j), обидва виражені в термінах обраної одиниці контексту, а також відповідні стандартизовані частоти суглобів $s_{ij} = (f_{ij}) / (f_i + f_j - f_{ij})$ організовані за допомогою матриці подібності, яка може бути представлена комбінацією кластерного аналізу та багатовимірного масштабування для виявлення значних вербальних асоціацій.

Тоді можна порівняти результати застосування багатовимірного масштабування до загальних за частотою еквівалентних матричних списків словників, отриманих із ряду текстів, використовуючи проксимальні індивідуальні відмінності (PINDIS), або застосувати до матриці індивідуальні відмінності масштабування (INDSCAL) себе. Процедура SUBRSTAT Forrest Young, яка перетворює неевклідові "об'єктні простори" в дуги, дозволяє їм більш точно аналізувати результати.

Обрахунок частотності TF-IDF (від англ. TF – term frequency, IDF – inverse document frequency) [5] – статистичний показник, що використовується для оцінки значущості слів у контексті документа, що є частиною колекції документа

або корпусу. Вага (значення) слова пропорційна кількості вживань слова в документі і обернено пропорційна частоті вживання слова в інших документах колекції. TF-IDF використовується для аналізу тексту та пошуку інформації.

TF – відношення числа входжень обраного слова до загальної кількості слів документа. Таким чином, оцінюється важливість слова t_i в межах обраного документа.

$$TF = \frac{n_i}{\sum_k n_k}, \tag{2.1}$$

де n_i є число входжень слова в документ, а в знаменнику – загальна кількість слів в документі.

IDF (зворотна частота документа) - інверсія частоти, з якою зустрічається слово в колекційному документі. Використання IDF зменшує вагу часто вживаних слів.

$$IDF = \log \frac{|D|}{|(d_i \supset t_i)|}, \tag{2.2}$$

Де |D| – кількість документів колекції;

$|(d_i \supset t_i)|$ – кількість документів, в яких зустрічається слово t_i (коли $n_i \neq 0$).

Вибір основи логарифму у формулі не має значення, оскільки зміна основи змінить вагу кожного слова постійним коефіцієнтом, тобто співвідношення ваги залишиться незмінним.

Іншими словами, TF-IDF є продуктом двох факторів: TF та IDF.

TF-IDF надаватимуть більшої ваги словам з високою частотою зустрічання в документі та низькою частотою використання в інших документах колекції [6].

Методи текстового майнінгу розроблені для автоматичного отримання знань з текстових матеріалів, що дозволяє експерту обробляти величезну кількість інформації. Одночасно пошук ключових слів, кластеризація та

класифікація документів, побудова смислових мереж використовуються для оцінки змісту та визначення загальних тем.

Наївний класифікатор Байєса [7] - класифікатор ймовірності, який використовує байєсівську теорему для визначення ймовірності спостереження (елемента вибірки), що належить до одного з класів, припускаючи (наївну) незалежність змінних.

У цьому методі ймовірність того, що документ d належить до класу c, обчислюється так:

$$P(c|d) \approx P(c) \prod_{1 \le k \le n_d} P(t_k|c). \quad (2.3)$$

Тут $P(t_k|c)$ – умовна імовірність, що термін t_k з'явиться в документі з класу c, $P(t_k|c)$ – міра правильного розпізнавання класу c по терміну t_k , P(c) – апріорна імовірність того, що документ належить класу c. Якщо терміни документа не дозволяють чітко відокремити один клас від іншого, то варто вибрати той з них, що має більш високу апріорну імовірність. Послідовність $\langle t_1, t_2, \dots, t_{n_d} \rangle$ складається з лексем документа d, що є частиною словника, використовуваного для класифікації, а n_d – кількість таких лексем у документі d. Наприклад, послідовність $\langle t_1, t_2, \dots, t_{n_d} \rangle$ для документа Beijing and Taipei join the WTO, що складається з одного речення, може мати вигляд Beijing, Taipei, join, WTO, де n_d=4, якщо видалити стоп-слова and і the. Мета класифікації текстів – знайти найкращий клас для документа. У методі NB найкращим вважається найбільш ймовірний клас, чи клас c_{map}, що має максимальну апостеріорну імовірність (MAP).

$$c_{map} = \arg\max P(c|d) = \arg max P(c) \prod_{1 \le k \le n_d} P(t_k|c) \quad (2.4)$$

Цей метод використовує імовірнісну модель, в якій класифікація та включення до відповідної категорії документів здійснюється шляхом оцінки

ймовірності появи слів у документі [8]. Ймовірності можна використовувати для оцінки найближчих категорій тестового документа.

Основними перевагами наївного байєсівського класифікатора є простота впровадження та низькі обчислювальні витрати на навчання та класифікацію. У рідкісному випадку, коли риси справді незалежні (або майже незалежні), наївний байєсівський класифікатор (майже) оптимальний. Основним недоліком методу є відносно низька якість класифікації у більшості реальних завдань. Цей метод часто використовується як базовий метод при порівнянні різних методів машинного навчання.

2.2 Розробка алгоритму аналізу логіко-лінгвістичних моделей текстових документів

Аналіз логіко-лінгвістичних моделей текстових документів потрібно здійснювати за чітко визначеним алгоритмом (рисунок 2.1).

1. Аналізується перший параметр лінгвістичної складової моделі cq_i. Так як тип тексту визначає його структуру, а також стилістичні, семантичні та синтаксичні особливості, то в залежності від значення cq_i для тексту будуть характерні певні граматичні особливості, на які буде звертатися увага при подальшому аналізі. Тоді можна сказати, що існує такий оператор $Q_i(r_i)$, який ставить у відповідність конкретному значенню змінної cq_i із множини можливих значень CQ вектор граматичних параметрів r_i:

$$Qi(ri): CQ \rightarrow cq_i \qquad (2.5)$$

де $CQ=\{ cq_1,\ldots, cq_i,\ldots, cq_n \}$ – множина існуючих типів текстів, n – кількість типів.

2. Фіксується кількість складних частин електронного документа f_j, $j = 1, m$, m – кількість складних синтаксичних частин.

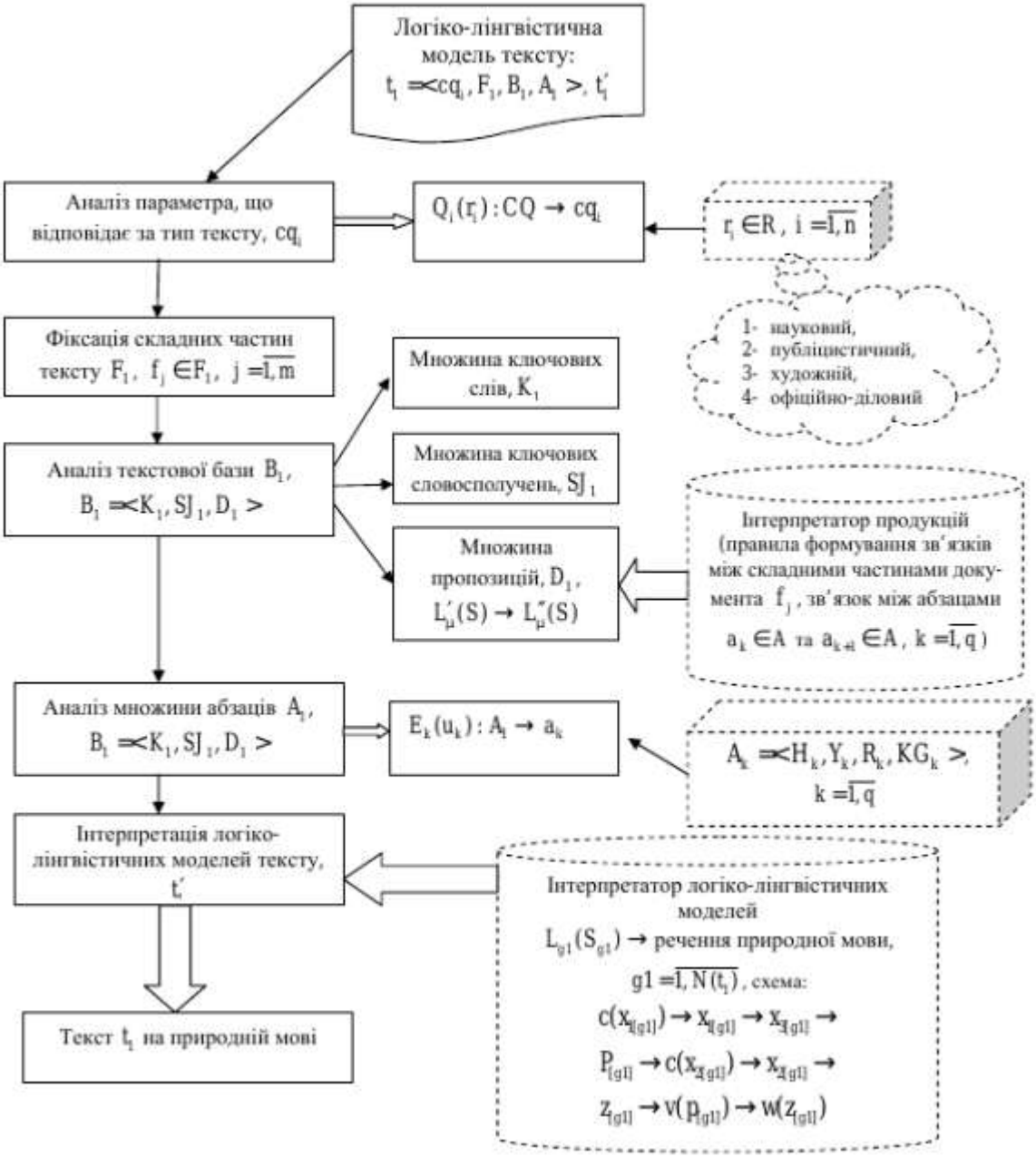

Рисунок 2.1 – Алгоритм аналізу логіко-лінгвістичних моделей текстових документів

3. На відміну від двох попередніх параметрів, текстова база B_1 є однією із значущих змінних для формування змістовного портрета документа. Множина ключових слів, множина комбінацій ключових слів і множина речень є основою для отримання знань з електронного документа.

2.3 Обґрунтування використання методів-фільтрів для аналізу англомовного тексту на наявність сталих мовних конструкцій

Методи фільтрації не взаємодіють з алгоритмом тренувань і вибирають оптимальні підмножини функцій, використовуючи лише інформацію, отриману з навчальної вибірки. Методи фільтрації виконуються на етапі попередньої обробки, перед виконанням алгоритму навчання. Фільтри методів можуть як незалежно оцінювати інформативність ознак навчання, так і оцінювати підмножину ознак у сукупності. У першому випадку необхідно визначити значення граничної константи (потрібно відкинути ті особливості, інформативність яких для алгоритму навчання нижче порогового значення). У другому випадку потрібно шукати простір знаків підмножини. Цей підхід має найменшу обчислювальну складність серед розглянутих підходів, а також має масштабованість та простоту використання. Такі методи показують досить хороші результати на практиці.

Методи фільтрації на основі незалежного ранжування ознак означають, що для кожної ознаки обчислюється деяка функція оцінювання (ранг). В оптимальному підмножині функцій залишаються лише ті ознаки, ранг яких перевищує певне заздалегідь визначене порогове значення. Порогові значення можуть бути обрані для перехресної перевірки, або навчальний зразок можна розділити на три підмножини, одна з яких буде використана для коригування значення параметра (він перевірятиме різні класифікатори, отримані при різних порогових значеннях), інший - тренувати класифікатор на остаточному підмножині. функції, а третє - для перевірки отриманого класифікатора.

Отже, якщо існує певна множина документів D, множина класів $C=\{c_1,\ldots,c_{|M|}\}$, до яких можуть відноситися документи, і існує деяка цільова залежність $a^*:D\times C\rightarrow\{0,1\}$, значення якої відомі лише на кінцевому наборі документів навчальної вибірки

$$D^n = \{(d_1, c_1), K, (d_n, c_n)\} \subset D \times C. \qquad (2.6)$$

Вихідні дані:

T – навчальна вибірка, тобто множина пар $\{(x_k, y_k)\}$;

x_k – k-й вхідний об'єкт, k = 1,…, m;

y_k – вихідні змінні (класи), k = 1,…, m, $y_k \in \{c_1,..., c_{|M|}\}$;

M – число класів;

n – число ознак;

X – множина ознак;

x^i – i-та ознака, i = 1,…, n.

Задача класифікації текстів – пошук найкращого наближення a цільової функції a^* на базі навчальної вибірки. Якість алгоритму відбувається за контрольним вибором $D^k \subset D$ - порівнюють відповіді, видаючи алгоритм контрольних виборів, з істинними заздалегідними відомими для них відповідей за допомогою функціональної якості.

2.4 Розробка математичної моделі для аналізу англомовного тексту на наявність сталих мовних конструкцій

Для того, щоб оцінити насиченість тексту сталими конструкціями, слід виконати наступні перетворення.

Текст, що потребує аналізу:

$$Doc = \{Doc_k(W, W_d)\}, \quad (2.7)$$

де W – набір слів, що входять до документу;

W_d – залежності або зв’язки між словами.

Для пошуку слів, які можна рекомендувати проектувальнику в якості сутностей аналізу, можна використовувати статистичний підхід (метод підрахунку TF-індексу) в поєднанні з методами синтаксичного аналізу текстів для моделювання для моделювання зв’язків між сутностями та атрибутами.

Показник TF (англ. term frequency – частота слова) [9] – статистична міра, яка використовується для оцінки важливості слова в контексті документа Doc_k (W,W_d). Вона визначається як відношення ня n_i деякого слова W_i до загальної кількості слів документа.

$$TF(W_i, Doc_k) = \frac{n_i}{\sum_k n_k}. \quad (2.8)$$

Атрибути БД - це слова та фрази, які не повинні мати високе значення TF. Ви також можете використовувати TF-IDF (термін "частота", "TF-IDF") для вимірювання важливості суб'єкта господарювання; статистичний захід, який використовується для оцінки важливості слова в контексті документа, який є частиною колекції документа або корпусу .

Виходячи з вищенаведеної математичної моделі, яка враховує результати аналізу тексту з використанням лінгвістичного та статистичного підходів, може бути запропонований наступний загальний алгоритм підтримки процесу моделювання [10].

1 етап: попереднє опрацьовування документу Doc_k (W, W_d) з метою виявлення ключових слів $W_k \subset W$, серед атрибутів БД. На цьому етапі проводиться графематичний аналіз тексту $Doc(W, W_d)$ та виділення інформативних частин слів. Кандидати в ключові слова $W \in W_k$ обираються серед слів, які представляють цінність при даному типі обробки.

2-й етап розрахунок частоти зустрічаємості сутностей tf (W_i, Doc_k) в документі Doc_k (W, W_d) ключових слів $W_i \in W_k$. З урахуванням tf (W, Doc_k) слово може бути віднесено до множини ключових атрибутів.

3-й етап: підтримки проектувальника. Під час цього проводиться синтаксичний аналіз речень, у складі яких є слова-атрибути, що претендують на роль ключових показників.

4-й етап: корегування показників моделі на основі виділених проектувальником сутностей та атрибутів. За результатами обраних

проектувальником сутностей та атрибутів можуть бути проведенні перерахунки показників моделі:

Значення мінімальних частот зустрічаємості сутностей tfe_{min} та атрибутів tfa_{min} можуть бути перераховані на основі частот зустрічаємості виділених сутностей та атрибутів.

В результаті отримуємо коефіцієнти частоти сутностей, з яких робимо загальний висновок.

Після проведення обрахунку частоти зустрічаємості сутностей необхідно віднести текст до певної категорії. Для цього доцільно розробити алгоритм класифікації на основі методу Байєса [11].

Нехай P (c_i | d) – ймовірність того, що документ, представлений вектором d = (t_1, ..., t_n), відповідає категорії c_i для i = 1, ..., | C |. Завдання класифікатора полягає в тому, щоб підібрати такі значення c_i і d, при яких значення ймовірності P (c_i | d) буде максимальним:

$$CSV(d) = \arg\max P(c_i|d) \tag{2.9}$$

Для обчислення значень P (c_i | d) користуються теоремою Байєса:

$$P(c_i|d) = \frac{P(c_i)P(d|c_i)}{P(d)} \tag{2.10}$$

де P (c_i) - безумовна ймовірність того, що документ віднесений до категорії c_i;

P (d | c_i) - ймовірність знайти документ, представлений вектором d = (t_1, ..., t_n), в категорії c_i;

P (d) – ймовірність того, що довільно взятий документ можна представити у вигляді вектору ознак d = (t_1, ..., t_n).

По суті P (c_i) є відношенням кількості документів з навчальної вибірки L, віднесених до категорії c_i, до кількості всіх документів з L.

P (d) не залежить від категорії c_i, а значення t_1, ..., t_n задані заздалегідь, тому знаменник – це константа, що не впливає на вибір найбільшого з значень $P(c_i \mid d)$.

Обчислення P (d | c_i) важко через велику кількість ознак t_1, ..., t_n, тому роблять «наївне» припущення про те, що будь-які дві координати, що розглядаються як випадкові величини, статистично не залежать одне від одного. Тоді можна скористатися формулою

$$P(d|c_i) = \prod_{k=1}^{n} P(t_k|c_i) \tag{2.11}$$

Далі пояснюється припущення, чому цей алгоритм називають наївним.

Припущення умовної незалежності.

На природній мові ймовірність того, що слово залежить від контексту. Байєсівський класифікатор представляє документ у вигляді набору слів і ймовірностей, які умовно не залежать одна від одної. Такий підхід іноді називають моделлю мішка зі словами. Виходячи з цього припущення, умовна ймовірність документа наближає до умовного добування ймовірності всіх слів.

$$P(d|c) \approx P(w_1|c)P(w_2|c) \dots P(w_n|c) = \prod_{i=1}^{n} P(w_i|c). \tag{2.12}$$

Виходячи з цього, одержимо:

$$c_{map} = \arg\max_{c \in C} \left[P(c) \prod_{i=1}^{n} P(w_i|c) \right]. \tag{2.13}$$

Проблема арифметичного переповнення.

При великому об’ємі документа довічно перемножують велику кількість дуже маленьких чисел. Для того, щоб цього уникнути, використовують

властивість алгоритма добутку logab = log$a$ + log$b$. Так як логарифм - функція монотонна, її застосування до обох частин вирази змініть тільки чисельне значення, проте параметра, при яких досягається максимум, залишаються незмінними. При цьому, логарифм від кількості близьких до нуля буде числом від'ємним, але в абсолютному значенні фактично більшим, ніж вихідне число, що робить логарифмічні значення ймовірності зручнішим для аналізу. Отож, слід переписати формулу використовуючи логарифм.

$$c_{map} = \arg\max_{c \in C} \left[logP(c) + \sum_{i=1}^{n} logP(w_i|c) \right]. \qquad (2.14)$$

При цьому можна використовувати будь-яку основу логарифма.

Оцінка параметрів моделі Байєса.

Оцінка ймовірностей відбувається на навчальній вибірці. Імовірність класу оцінюється відношенням кількості документів, що належать до певного класу до загальної кількості документів. Подібним чином обчислюється ймовірність сталих мовних конструкцій в класі. Тобто в чисельнику записують скільки разів мовна конструкція зустрічається в документах класу, а в знаменнику – сумарна кількість конструкцій у всіх документах цього класу.

Проблема невідомих слів.

Якщо на етапі класифікації ми зустрічаємо конструкцію, яку ми не бачили на етапі навчання, то її значення, як і значення звичайних слів дорівнюватимуть нулю. Це приведено до того, що документ із цим словом не може бути класифікований. Позбутися цієї проблеми аналізуючи більшу кількість документів не можна. Неможливо створити навчальну вибірку, яка б охоплювала всі можливі конструкції, включаючи помилки, синоніми і т. д. Типовим рішенням проблем невідомих слів є адитивне згладжування (згладжування Лапласа). Ідея, полягає в тому, що ми додаємо одиницю до частоти кожного слова.

$$P(w_i|c) = \frac{W_{ic} + 1}{\sum_{i' \in V}(W_{i'c} + 1)} = \frac{W_{ic} + 1}{|V| + \sum_{i' \in V} W_{i'c}} \quad (2.15)$$

Підставивши обрані оцінки у формулу (2.14), одержимо остаточну формулу, по якій будемо проводити класифікацію.

$$c_{map} = \arg\max_{c \in C} \left[log \frac{D_c}{D} + \sum_{i=1}^{n} log \frac{W_{ic} + 1}{|V| + \sum_{i' \in V} W_{i'c}} \right]. \quad (2.16)$$

Класифікатор.

Щоб реалізувати класифікатор, слід вибрати навчальну вибірку, яка має відповідність між класом і документом. Далі вибрати статистику, що буде використовуватися на етапі класифікації:

- відносна частота класів, тобта як часто зустрічаються документи одного класу;
- сумарна кількість сталих мовних конструкцій документів кожного класу;
- відносні частоти конструкцій в межах одного класу;
- розмір словника.

Цю інформацію будемо називати моделлю класифікатора.

Формування імовірнісного простору.

Оцінки, які отримуємо використовуючи даний алгоритмом, не задовольняють двом формальними властями, які вимагають задовольняти всі імовірнісні оцінки:

- потрібно бути в діапазоні від нуля до одиниці;
- їхня сума повинна дорівнювати одиниці.

Для того, щоб вирішити ці завдання, потрібно з логарифмічних оцінок сформувати імовірнісний простір. Тобто: позбутися від логарифмів і нормувати суму по одиницях.

2.5 Проектування структури інформаційної технології аналізу англомовного тексту на наявність сталих мовних конструкцій

Виявлення знань в тексті – це нетривіальний процес створення справді нових, потенційно корисних та зрозумілих зразків у неструктурованих текстових даних. Якщо ви хочете знати, що вони використовуються, вони представляють логічно доступний текст без будь-яких обмежень щодо його структури. Загалом такі документи можуть бути складними та великими та включати не лише текстові, але й графічні статті. Обробляючи текстові тексти, можна уявити велику кількість коротких кроків (мал. 2.2) [12]:

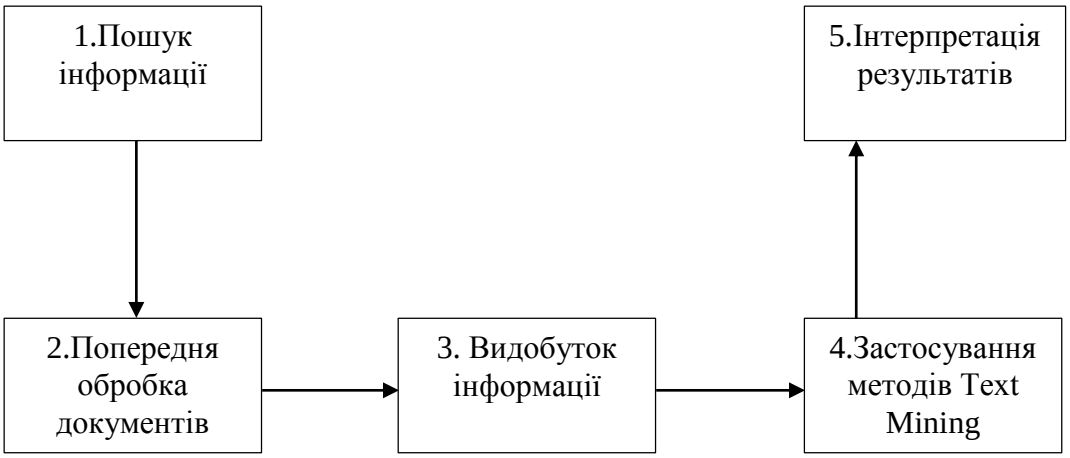

Рисунок 2.2 – Структурна схема аналізу текстових документів

1. Пошук інформації. Перший крок - визначити документи, які потрібно проаналізувати, та зробити їх доступними.

2. Попередня обробка документів. На цьому кроці виконуються найпростіші, але необхідні перетворення документів для їх представлення у формі, з якою працюють методи Text Mining. Мета таких перетворень - видалити непотрібні слова та зробити текст більш суворим. На етапі попередньої обробки слова стоп видаляються. Такі слова є допоміжними, тому в них мало інформації про зміст документа. Також перетворює всі символи у верхній або нижній регістр.

3. Видобуток інформації. Вилучення інформації з відібраних документів передбачає визначення ключових понять, які будуть додатково проаналізовані.

4. Застосування методів Text Mining. На цьому етапі витягуються шаблони та зв’язки, наявні в текстах.

5. Інтерпретація результатів. Останній крок у процесі аналізу включає інтерпретацію результатів. Як правило, це або представити результати природною мовою, або зобразити їх графічно.

Вилучення ключових понять із тексту можна розглядати як окремий етап аналізу тексту, так і як конкретне завдання програми [13]. У першому випадку витягнуті факти використовуються для вирішення різних завдань аналізу: класифікації, кластеризації тощо. У завданні вилучення ключових понять із тексту, деяких сутностей, подій та зв’язків (у цьому випадку сталеві конструкції та їх частота у тексті) представляють інтерес.

Завдання аналізу тексту в цьому випадку - знайти в тексті слова, які відповідають мовним конструкціям, знайденим у базі даних, обчислити частоту появи таких конструкцій у тексті та присвоїти текст певному стилю.

Вхід програмного модуля містить англомовний текст, набір ключових слів і фраз, які зберігаються в базі даних. Після отримання цієї інформації починається пошук мовних конструкцій у тексті та їх частота обчислюється за допомогою методу обчислення TF-індексу. Тоді, виходячи з попереднього аналізу, класифікація тексту - віднесення до певного стилю. На виході ми отримуємо результати аналізу, які відповідають завданню аналізу. Система матиме інтерфейсний модуль, структурний модуль пошуку, модуль обчислення частоти та модуль відображення.

Розроблена модель функціонування інформаційної технології зображена на рисунку 2.3. Структура інформаційної технології аналізу англомовного тексту на наявність сталих мовних конструкцій зображено на рисунку 2.4. Схема загального алгоритму функціонування інтелектуальної технології зображена на рисунку 2.5.

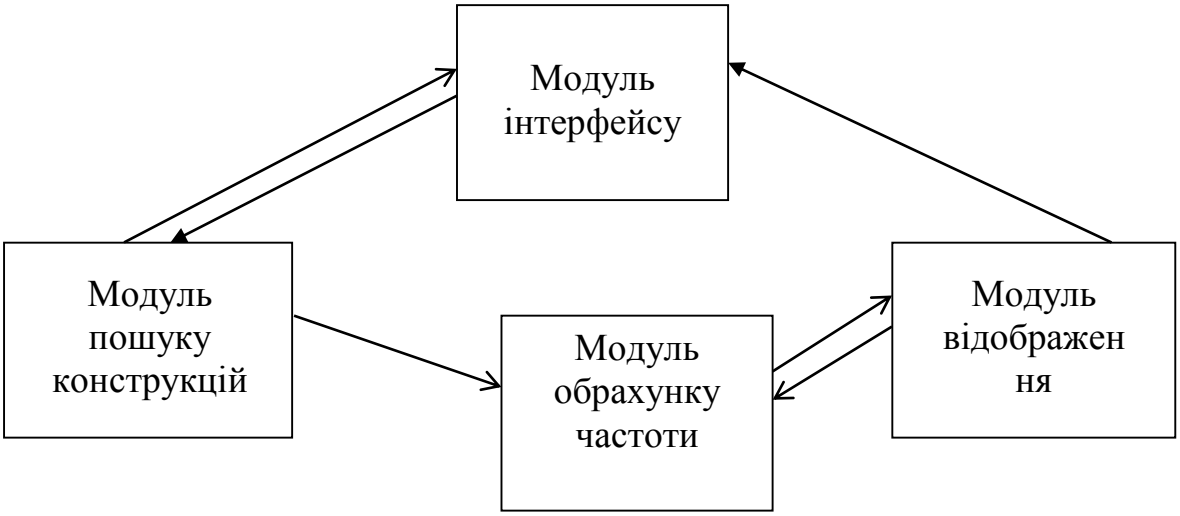

Рисунок 2.3 – Модель функціонування інформаційної технології

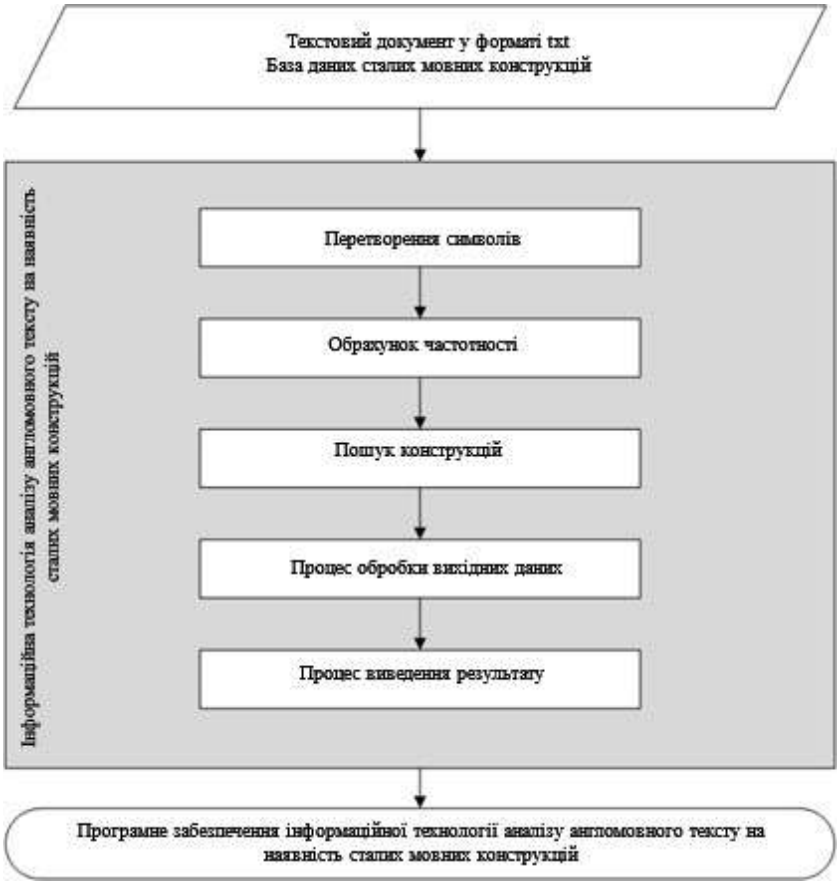

Рисунок 2.4 – Структура інформаційної технології аналізу англомовного тексту на наявність сталих мовних конструкцій

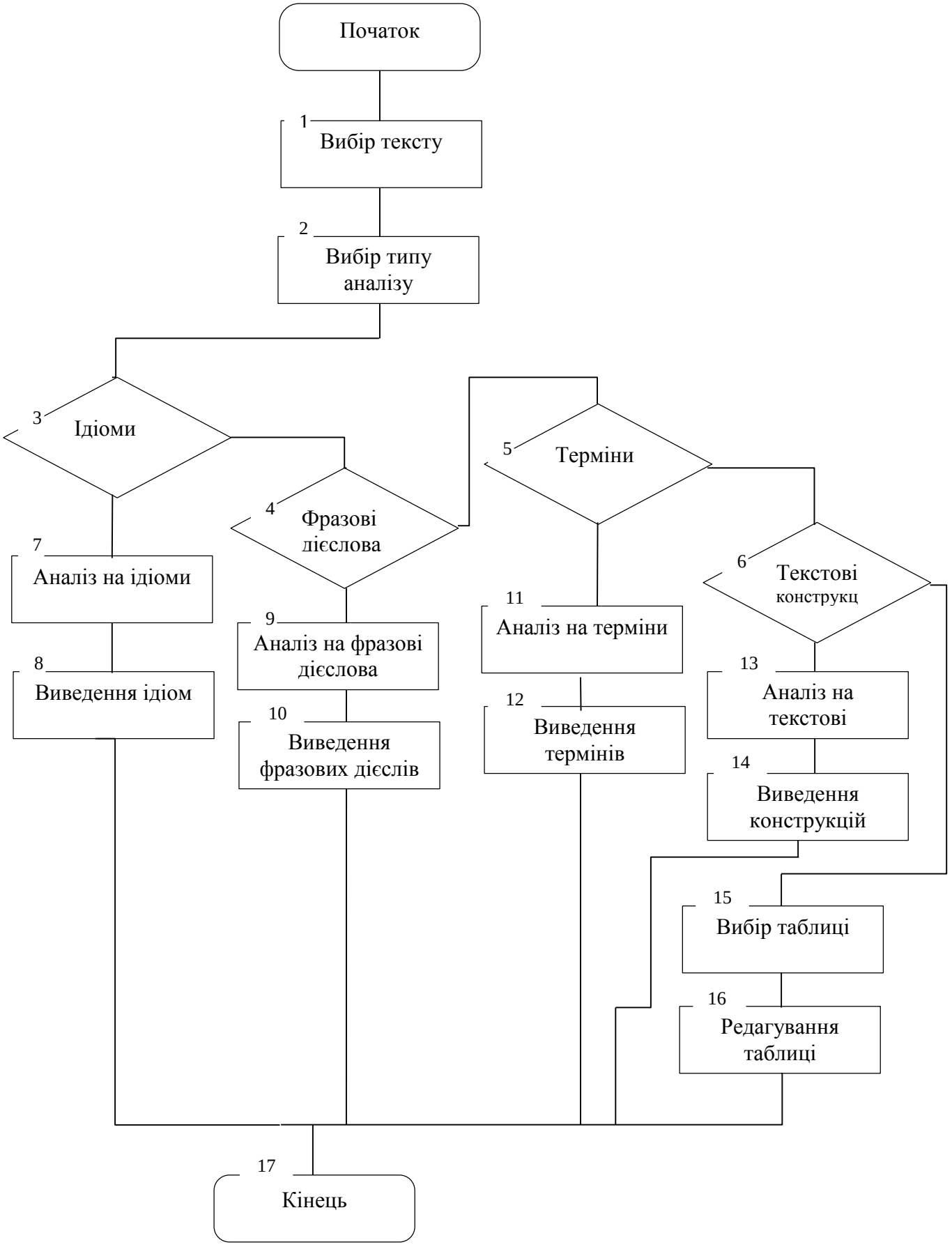

Рисунок 2.5 - Схема загального алгоритму функціонування інтелектуальної технології

2.6 Розробка бази даних англомовних сталих конструкцій

Для опису предметної області необхідні наступні сутності: тематика, терміни, ідіоми, фразові дієслова, текстові конструкції. В універсальне відношення потрібно включити атрибути, що описують дані сутності. Отже, універсальне відношення для даної бази даних буде мати наступний вигляд:

R (назва тематики, коефіцієнт, термін, переклад, сфера застосування, ідіома, переклад ідіоми, значення, тематика, слово, переклад дієслова, аналог, застосування дієслова, фраза, переклад конструкції, тематика).

Ступінь універсального відношення – 16.

Представлення ER-моделі для бази даних зображено на рисунку 2.6.

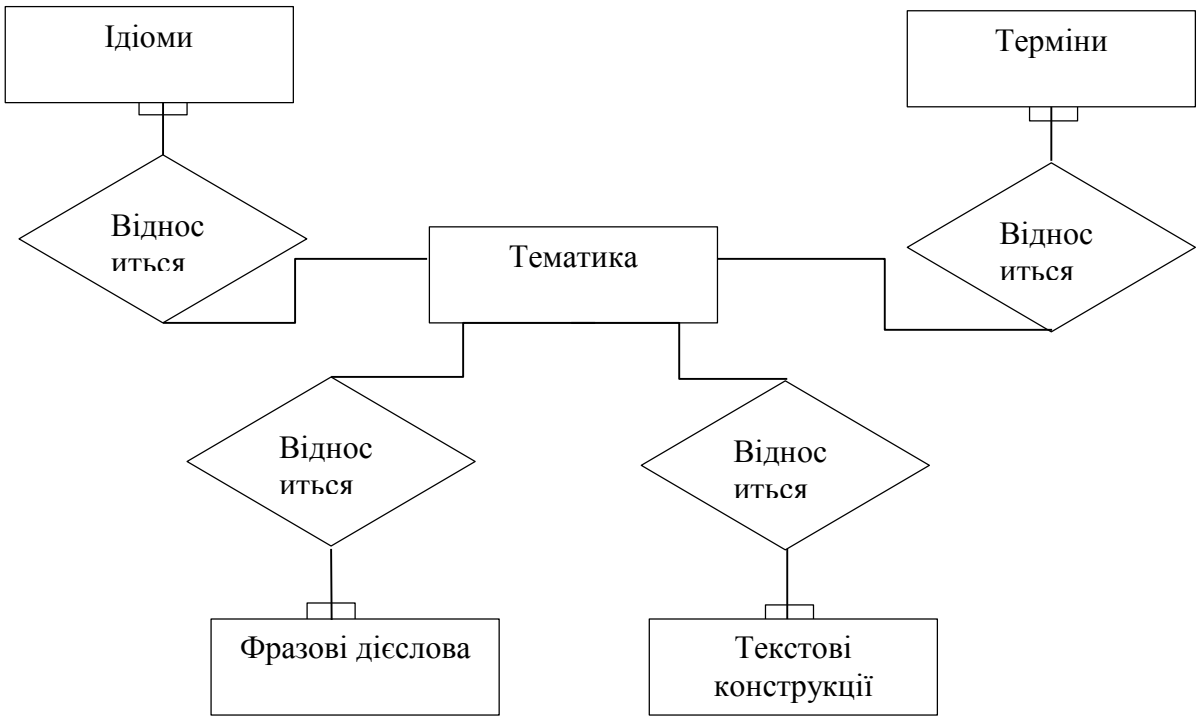

Рисунок 2.6 – ER-модель для бази даних сталих мовних конструкцій

Представимо базу даних, використовуючи реляційну модель даних, яка має переваги простоти та доступності для розуміння користувачем; єдина використовувана інформаційна структура – «таблиця»; суворі правила проектування, засновані на математичному апараті; повна незалежність даних;

зміни в прикладній програмі при зміні реляційної бази даних мінімальні; можливість розширення бази даних шляхом додавання нових елементів, записів без зміни існуючих підсхем і додатків.

Згідно з правилами побудови концептуальних схем домену у вигляді ER-структур, сутності представлені позначеними прямокутниками, асоціаціями ромбусами, а зв'язки між ними непрямими ребрами, над якими ступінь зв'язку та необхідна пояснення може бути встановлено.

2.7 Висновок

Дослідження показали, що для вирішення проблеми аналізу англомовного тексту на наявність сталих мовних конструкцій методи підрахунку TF-індексу та Байєсівського класифікатора мають переваги у швидкості роботи, що обумовлена низькими обчислювальними витрати на навчання та класифікацію.

Обґрунтовано використання даних методів для розв'язання поставленої задачі та методів фільтрації, в наслідок чого запропоновано математичну модель для даної інформаційної технології. Охарактеризовано загальний підхід до аналізу документів, а також методи, що використовуються в програмі-аналозі. Розроблено алгоритм аналізу логіко-лінгвістичних моделей текстових документів.

Запропоновано структуру інформаційної технології аналізу англомовного тексту на наявність сталих мовних конструкцій.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ АНГЛОМОВНОГО ТЕКСТУ НА НАЯВНІСТЬ СТАЛИХ МОВНИХ КОНСТРУКЦІЙ

3.1 Обґрунтування вибору мови та середовища програмування

Кожна мова та технології існують для конкретних завдань та цілей. Наприклад, деякі мови призначені лише для розробки сайтів, інші - лише для розробки програм Windows, а деякі можуть поєднувати обидві функції. Існує кілька мов, які підходять для розробки "звичайних" програм для Windows. Найпопулярніші - C #, C ++ та Java.

Java - одна з найпоширеніших і популярних мов програмування на сьогоднішній день. Java - мова програмування, розроблена Sun Microsystems з 1991 року і офіційно випущена 23 травня 1995 року. На сьогоднішній день Java пройшла довгий шлях, випускається багато різних версій. Java активно використовується для створення програмного забезпечення для цілого ряду пристроїв: звичайних ПК, планшетів, смартфонів та мобільних телефонів і навіть побутової техніки [24].

Переваги Java [25]:

1. Однією з головних переваг Java є незалежність платформи, на якій запускаються програми: той же код можна запускати під операційними системами Windows, Solaris, Linux, Machintosh та іншими.

Це дійсно необхідно, коли програми завантажуються через Інтернет для подальшого запуску в різних операційних системах.

2. Синтаксис мови Java схожий на синтаксис мови C ++, і для програмістів, які знайомі з C і C ++, вивчити це не складно.

3. Java - цілком об'єктно-орієнтована мова, навіть більше, ніж C ++. Усі сутності на Java - це об'єкти, за винятком кількох примітивних типів, таких як числа. (Об'єктно-орієнтоване програмування полегшує розробку складних проектів.)

4. Висока надійність. Творці надали мові Java інструменти, які виключають саму можливість створення додатків, які б приховували найпоширеніші помилки. У Java робиться наступне:

У таблиці 3.1 наведені порівняльні характеристики java з іншими мовами програмування, такими як C #, C ++, PYTHON.

Однією з ключових особливостей для розробників є не тільки підтримка багатьох функцій, але і безкоштовність мови [26].

Таблиця 3.1 – Порівняння мов програмування Java, C#, C++, PYTHON

Можливість	JAVA	C#	C++	PYTHON
Об’єктно-орієнтована	+	+	+	+
Статистична типізація	+	+	+	-
Динамічна типізація	-	+	-	+
Явна типізація	+	+	+	-
Неявна типізація	-	-	-	+
Можливість компіляції	+	+	+	+
Багатопотокова компіляція	+	-	+	-
Ручне управління пам’яттю	-	+	+	-
Збір сміття	+	+	-	+
Інструкція goto	+	+	+	-
Підтримка try/catch	+	+	-	+
Інтерфейси	+	+	+	+
Перегрузка функції	+	+	+	-
Безкоштовність	+	-	-	-
Сумісність різних версій	+	-	-	-

Характеристика середовища Eclipse

Існує кілька реалізацій програмних платформ Java, серед яких Eclipse займає особливе місце. Навколишнє середовище розробки модульних кросплатформенних додатків має ряд обов'язкових функцій, наявних в інших подібних продуктах. Серед них:

• Загальні технології для розгортання програм для оточення користувачів.

• Інструментарій для створення графічних інтерфейсів будь-якої складності.

• Бібліотеки для віддаленого управління об’єктами та доступу до програмного забезпечення до даних.

• Інструменти для запуску, налагодження, компіляції, моніторингу та інших дій програми.

Eclipse - це провідна і, на думку багатьох галузевих експертів та розробників Java, найкраще середовище виконання Java на сьогоднішній день. Eclipse - це вільнодоступна основа для розробки модульних прикладних платформ, розроблених та підтримуваних Фондом Eclipse [27].

Особливості платформи Eclipse:

• крос-платформа - працює на операційних системах Windows, Linux, Solaris та Mac OS X;

• За допомогою програми Eclipse можна програмувати на багатьох мовах, таких як Java, C і C ++, PHP, Perl, Python, Cobol та інші;

• є основою для розробки інших інструментів і пропонує великий набір API для створення модулів;

• Використовуючи підхід Rich Client Platform (RCP), Eclipse - це інструмент для створення практично будь-якого клієнтського програмного забезпечення [28].

Характеристика мови запитів SQL

SQL (Structured Query Language) - приклад мови перетворення даних або мови електронних таблиць для перетворення вхідних даних на потрібний вихід. Особливість пропозицій цієї мови полягає в тому, що вони орієнтовані більше на кінцевий результат обробки даних, ніж на процедуру цієї обробки. Сам SQL визначає, де розташовані дані, які індекси та навіть найефективніші послідовності операцій слід використовувати для їх отримання: не потрібно вказувати ці препарати у запиті до бази даних. SQL є першою і досі єдиною стандартною мовою бази даних, яка набула широкого поширення [29].

Впровадження в SQL концепції таблично-орієнтованих операцій дозволило створити компактну мову з невеликим (менше 30) набором

пропозицій. SQL можна використовувати як інтерактивно (для виконання запитів), так і як вбудований (для створення додатків).

Мова маніпулювання даними використовується для маніпулювання даними в таблицях баз даних. Він складається з чотирьох основних команд [30]:

- SELECT (вибрати);
- INSERT (вставити);
- UPDATE (обновити);
- DELETE (видалити).

Для реалізації даної інформаційної технології обирано саме цю мову, тому що мова SQL призначена для маніпулювання даними в реляційних базах даних, визначення структури баз даних та для управління правами доступу до даних в багатокористувацькому середовищі.

3.2 Програмна реалізація

На рисунку 3.1 зображено загальну UML-діаграму класів системи аналізу тексту на наявність сталих мовних конструкцій.

Були розроблені класи Main, MainGUI, SelectType, Analiz, Stata.

Клас Main відповідає за початкову активність програми. Викликає клас MainGUI.

Клас MainGUI відповідає за відображення головного меню. Він має методи stvor(), shov() та import(), які відповідають за відображення, приховування головного меню та вибір тексту для аналізу відповідно.

Клас SelectType відповідає за визначення типу аналізу для подальшої роботи.

Клас Analiz, клас з основною логікою програми, в ньому відбувається обрахунок частоти вживання сталих конструкцій та визначення стилю за допомогою застосування алгоритмів підрахунку TF-індексу та НБК. Результатом роботи даного класу є проаналізований текст.

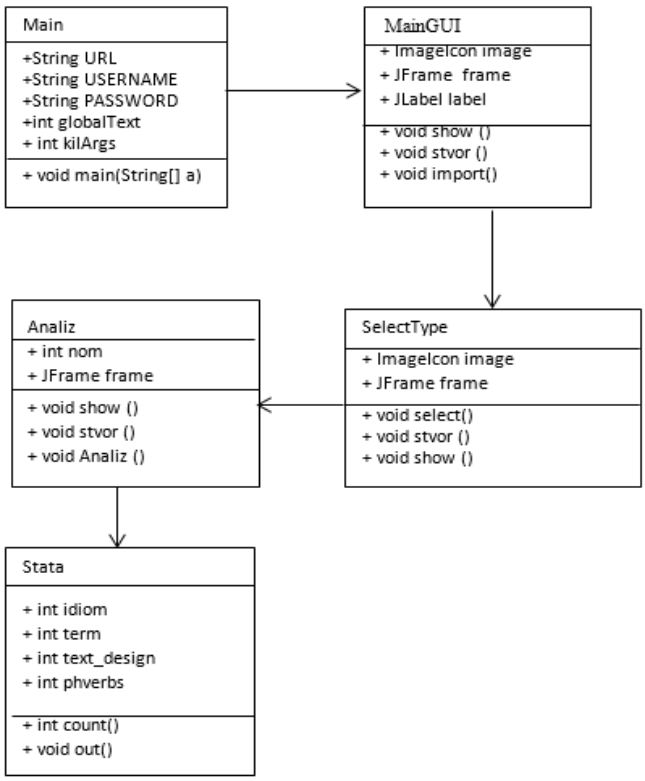

Рисунок 3.1 – Загальна UML-діаграма класів

Клас Stata призначений для відображення статистичної інформації проаналізованого тексту.

3.3 Розробка користувацького інтерфейсу

Класи, відповідальні за віконний користувальницький інтерфейс:

1) MainGUI – містить оброблювачі події, пов'язані із взаємодією користувача з додатком;

а) onClickText – обробляє натискання кнопки «Вибрати» для вибору файлу з текстом;

б) onClickLoad – обробляє натискання кнопки «Про додаток». Відкриває діалогове вікно для вибору серіалізованої бази знань;

в) onClickPredict – відкриває діалогове вікно, у якому потрібно вибрати файл із текстом для визначення тематики, або зчитує текст із відповідної форми;

г) showKnowledgeBaseInfo – метод для виводу інформації про базу знання класифікатора;

д) List createChooser(String title) – метод для вибору файлу виду аналізу;

е) void createErrorDialog – метод для створення діалогу з повідомленням про помилку.

2) Analiz – клас для зберігання в пам'яті додатка моделі класифікатора.

а) List trainingFiles – список файлів для навчання класифікатора;

б) List testingFiles – список файлів з невідомими документами, які потрібно класифікувати;

в) knowledgeBase – база знань класифікатора;

г) void trainClassifier() – метод-обгортка для навчання класифікатора;

д) Map> predict – метод-обгортка для класифікування невідомого документа.

3) NaiveBayesUIExample – точка входу в додаток. Визначення дизайну додатка з файлу ui.fxml.

4) Файл ui.fxml – файл, у якому описаний дизайн користувальницького інтерфейсу.

3.4 Тестування розробленої інформаційної технології та аналіз результатів роботи

Для тестування інформаційної технології аналізу англомовного тексту на наявність сталих мовних конструкцій були взяті різні новинні статті, сторінки казок, статті наукових та технічних журналів, розміщені в текстових файлах.

Було проведено 500 запусків додатку, протестовано можливості його роботи, що дало змогу адекватно оцінити його роботу. Тестування проводилось на текстах англійською мовою.

Після запуску програми відкривається вікно початкової активності (рисунок 3.2). Вікно початкової активності містить кнопку «Про додаток» та поле для вибору тексту.

Необхідно обрати текст, який має бути проаналізовано, або натиснути кнопку «Про додаток», щоб дізнатися принцип роботи програми.

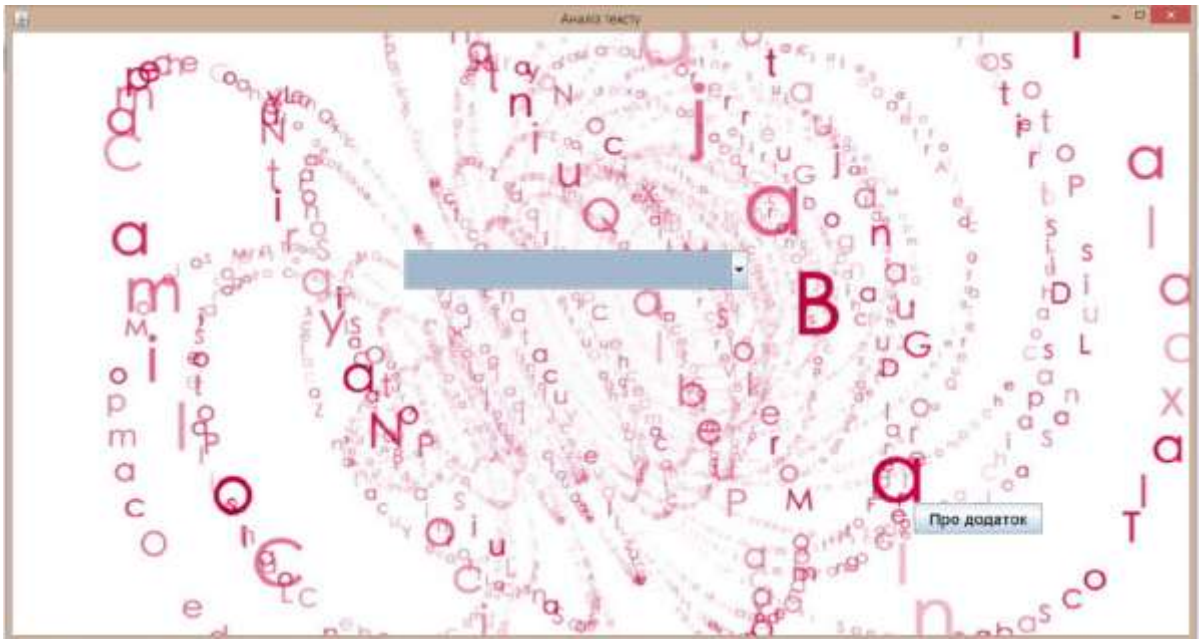

Рисунок 3.2 – Вікно початкової активності

При переході до наступного вікна користувач визначає необхідний тип аналізу і натискає відповідну кнопку (рисунок 3.3).

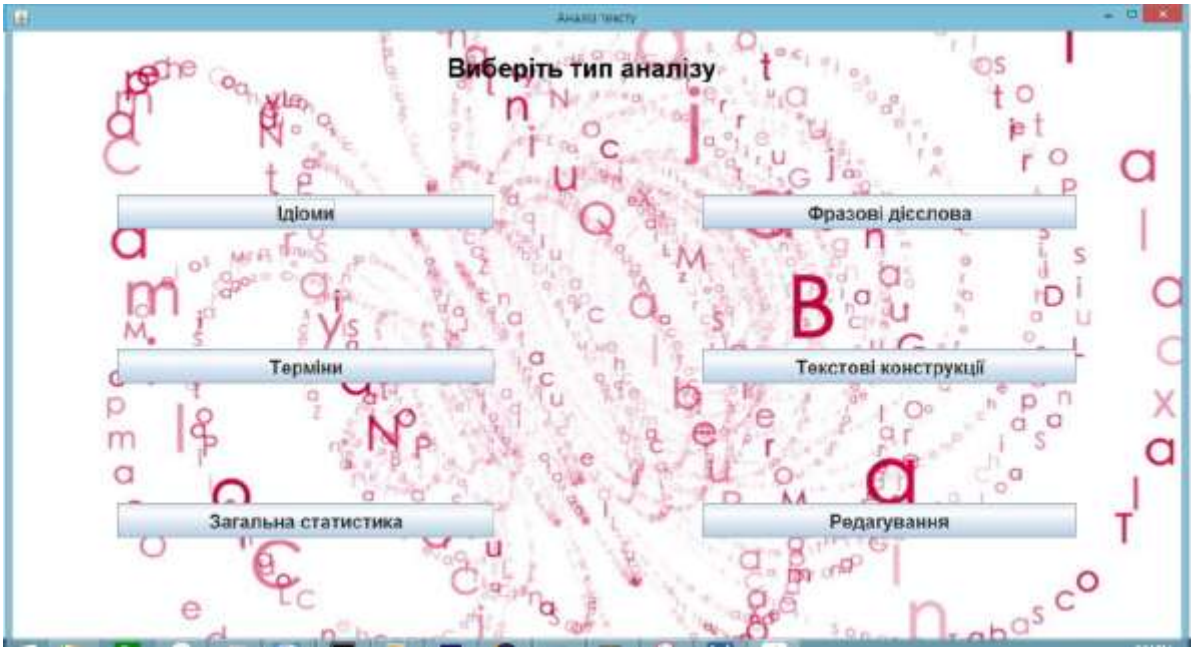

Рисунок 3.3 – Вибір типу аналізу

Як результат роботи програми виводиться загальна характеристика аналізу, або окремі таблиці, в яких виведені конструкції, наявні в аналізованому тексті (рисунок 3.4, 3.5, 3,6).

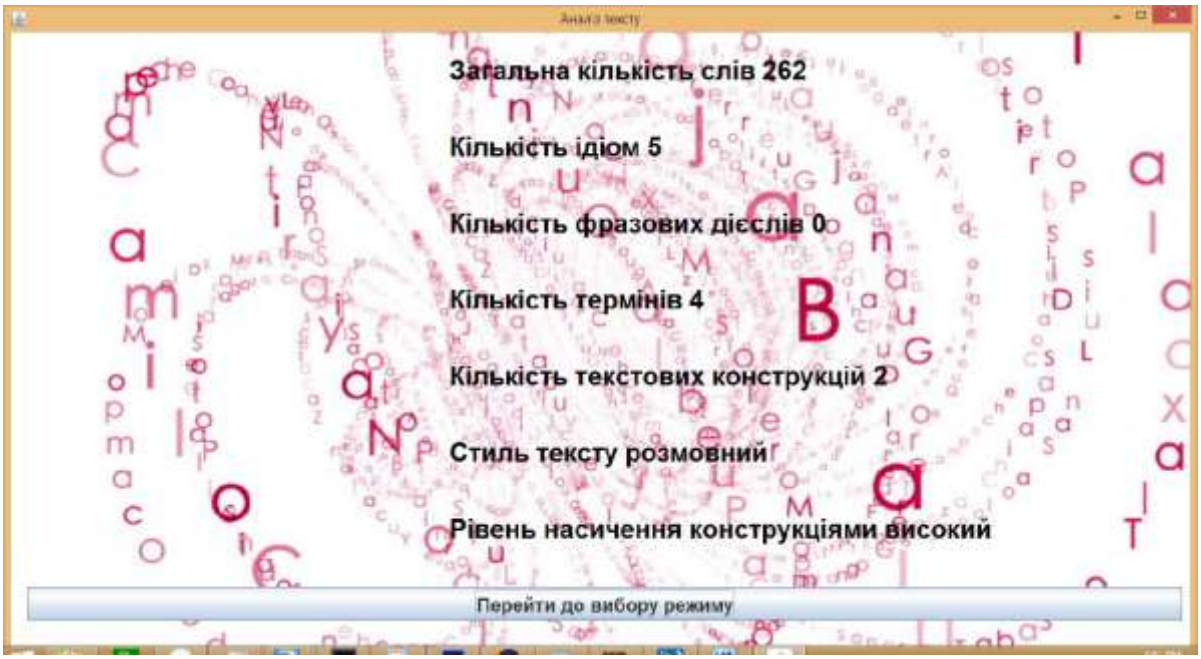

Рисунок 3.5 – Виведення конструкцій, наявних в тексті

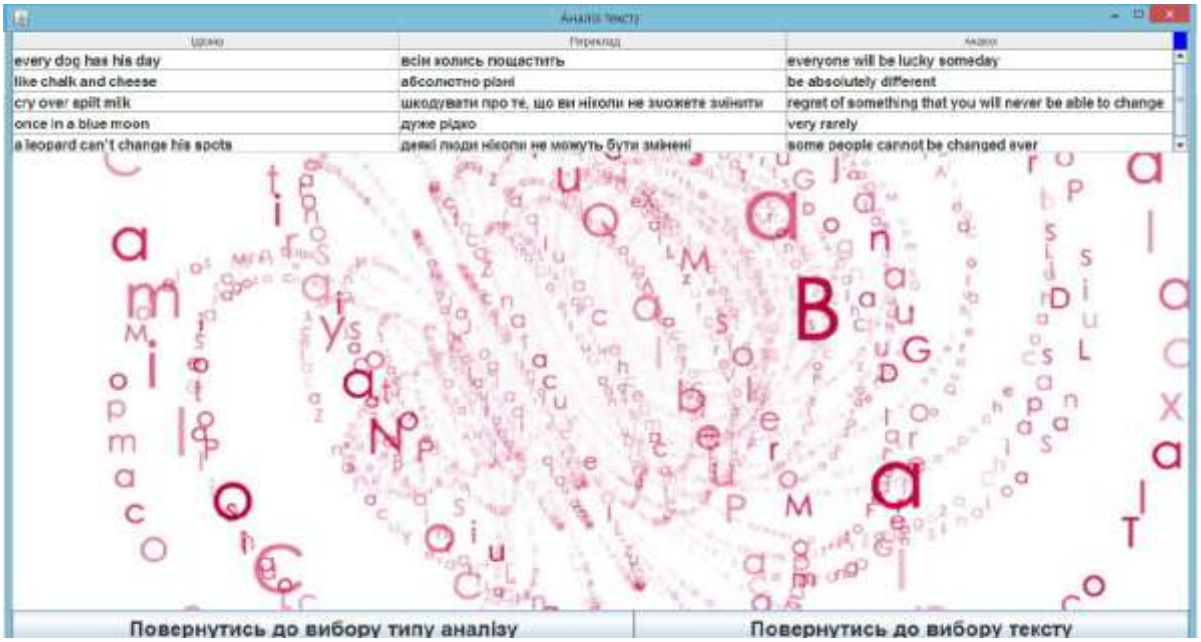

Рисунок 3.5 – Виведення конструкцій, наявних в тексті

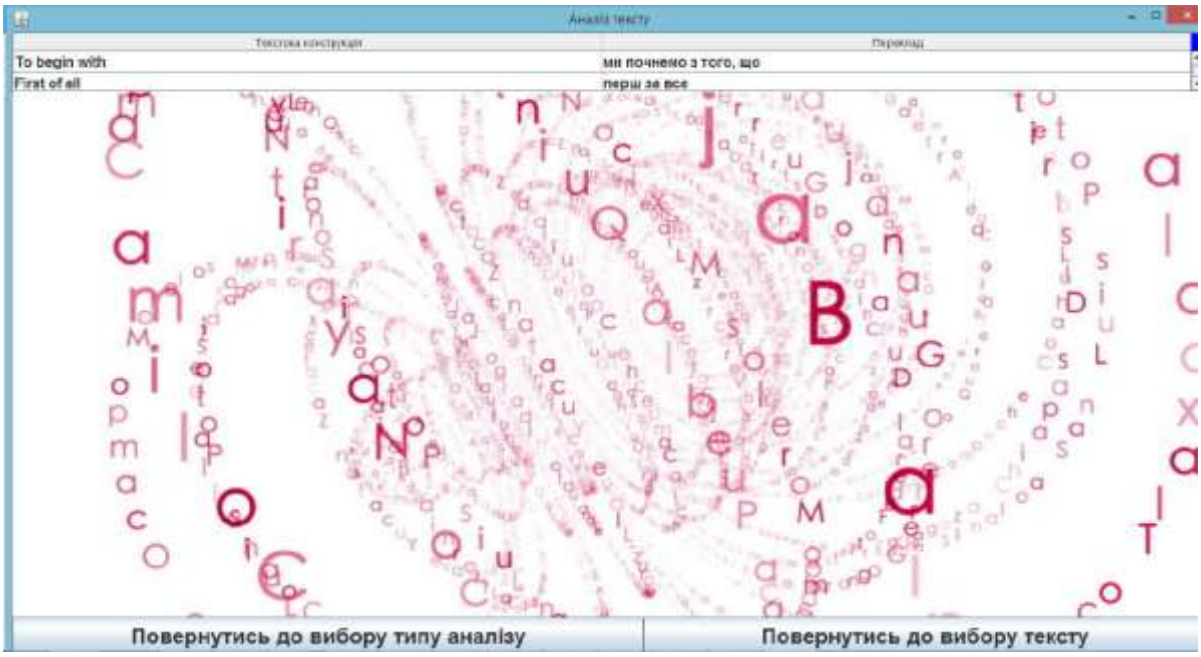

Рисунок 3.6 – Виведення конструкцій, наявних в тексті

В інформаційній технології є також можливість переглянути та редагувати та додати записи в базі даних (рисунок 3.7, 3.8). Для цього в головному меню потрібно натиснути кнопку «Редагування».

Рисунок 3.7 – редагування записів

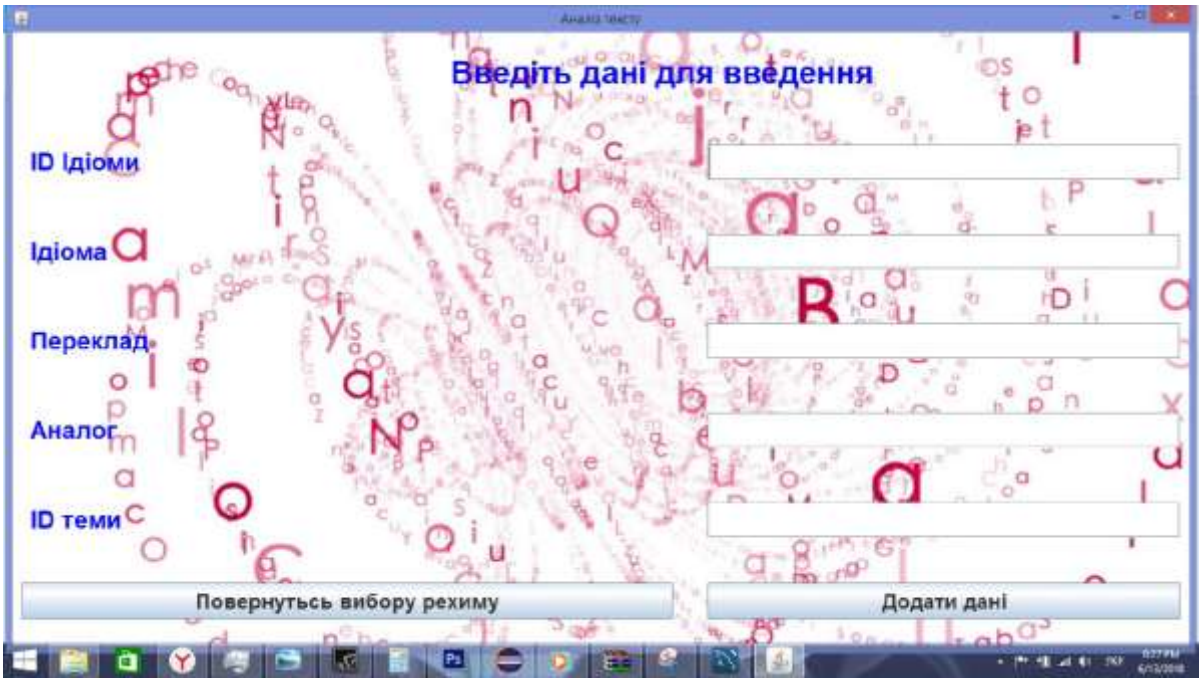

Рисунок 3.8 – додавання нових даних

В таблиці 3.2 наведено результати порівняння програмної реалізації обраного алгоритму з програмами аналогами, що застосовують для розв’язання задачі обробки тексту. Розроблений додаток дає швидкість вищу ніж програми-аналоги. Порівняно з прототипом швидкість зросла майже на 3,5%, що є дуже хорошим результатом.

Таблиця 3.2 – Порівняльний аналіз достовірності роботи різних алгоритмів вирішення задачі

Програма	Швидкість аналізу документів, с
Розроблений додаток	0.4163
HAMLET	0.4304
Textanz	0.4511
TextQuest	0.4857
Yoshikoder	0.4742

Отже, результати тестування відповідають очікуваним параметрам.

3.5 Висновок

Здійснено обґрунтування вибору мови програмування для реалізації інформаційної технології аналізу англомовного тексту на наявність сталих мовних конструкцій – Java. Розглянуто особливості середвища програмування Eclips.

Розроблено та описано UML-діаграму класів, та описано їх функціонування. На основі цього була програмно реалізована інформаційна технологія.

Проведено тестування розробленої інформаційної технології, та проаналізовано її результати. Тестування підтвердило правильність роботи довівши збільшення швидкості роботи майже на 3,5%.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть Колесницький О. К. та Озеранський В. С.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою.

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.1.

Таблиця 4.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	1. Експерт 1	2. Експерт 2
	Бали, виставлені експертами:	
1	4	4
2	4	3
3	3	4
4	4	3
5	3	4
6	4	4
7	3	3
8	4	4
9	4	4
10	4	3
11	3	4
12	3	4
Сума балів	$СБ_1 = 44$	$СБ_2 = 44$
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{1}^{3} СБ_i}{2} = 44$	

Отже, з отриманих даних таблиці 4.1 видно, що нова розробка має високий рівень комерційного потенціалу.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи.

Для розробки нового програмного продукту необхідні такі витрати.

Основна заробітна плата для розробників визначається за формулою (4.1):

$$З_о = \frac{М}{Т_р} \cdot t, \quad (4.1)$$

де М- місячний посадовий оклад конкретного розробника;

$Т_р$ - кількість робочих днів у місяці, $Т_р$ = 21 день;

t - число днів роботи розробника, t = 50 днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 4.2.

Таблиця 4.2 – Розрахунки основної заробітної плати

Працівник	Оклад М, грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	5500	261,90	5	1309,5
Інженер-програміст	4000	190,47	50	9523,5
Всього:				10833

Розрахуємо додаткову заробітну плату:

$$З_{дод} = 0,1 \cdot 10833 = 1083,3 \text{ (грн.)}$$

Нарахування на заробітну плату операторів НЗП розраховується як 37,5…40% від суми їхньої основної та додаткової заробітної плати:

$$Н_{зп}= (З_о + З_р)\cdot\frac{\beta}{100}, \quad (4.2)$$

$$Н_{зп}= (10833+1083,3)\cdot\frac{36,3}{100} = 4325,5 \text{ (грн.)}.$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$А=\frac{Ц\cdot Н_а}{100}\cdot\frac{Т}{12}, \quad (4.3)$$

де Ц – балансова вартість обладнання, грн;

$Н_а$ – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

Т – Термін використання (Т=3 міс.).

Таблиця 4.3 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань,грн
Персональний комп’ютер	9000	25	3	562,5
Всього:				562,5

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_1^n Н_і \cdot Ц_і \cdot К_і, \quad (4.4)$$

де n – кількість комплектуючих;

Ні - кількість комплектуючих і-го виду;

Ці – покупна ціна комплектуючих і-го виду, грн;

Кі – коефіцієнт транспортних витрат (приймемо Кі = 1,1).

Таблиця 4.4 - Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Флешка	шт.	180	1	180
Пачка паперу	уп.	120	1	120
Ручка	шт.	10	1	10
Всього з урахуванням транспортних витрат				341

Витрати на силову електроенергію розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п ; \quad (4.5)$$

де В – вартість 1кВт-години електроенергії (В=1,7 грн/кВт);

П – установлена потужність комп’ютера (П=0,6кВт);

Ф – фактична кількість годин роботи комп’ютера (Ф=190 год.);

$К_п$ – коефіцієнт використання потужності ($К_п < 1$, $К_п = 0,7$).

$$В_е = 1,7 \cdot 0,6 \cdot 190 \cdot 0,7 = 135,66 \text{ (грн.)}$$

Розрахуємо інші витрати $В_{ін}$.

Інші витрати $І_в$ можна прийняти як (100…300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$В_{ін} = (\mathbf{1..3}) \cdot (З_о + З_р). \quad (4.6)$$

Отже, розрахуємо інші витрати:

$$В_{ін} = \mathbf{1} * (\mathbf{10833} + \mathbf{1083,3}) = \mathbf{11916,3} \text{ (грн)}.$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$В = З_о + З_д + Н_{зп} + А + К + В_е + І_в$$

В = 10833+1083,3+4325,5+562,5+341+135,66+11916,3= 29197,26 (грн.)

Розрахуємо загальну вартість наукової роботи $B_{заг}$ за формулою:

$$В_{заг} = \frac{В_{ін}}{\boldsymbol{\alpha}} \quad (4.7)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$В_{заг} = \frac{\mathbf{29197,26}}{\mathbf{1}} = \mathbf{29197,26}$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{В_{заг}}{\beta} \quad (4.8)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{29197,26}{0,9} = \mathbf{32441,4} \text{ (грн.)}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо отримання прибутку від реалізації результатів нашої розробки. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Це забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності .

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства $\Delta П_і$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta П_i = \sum_1^n (\Delta П_я \cdot N + П_я \Delta N)_i \qquad (4.9)$$

де $\Delta П_я$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$П_я$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки витрати на виготовлення інформаційної технології зменшаться на 20 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 20 грн), а кількість користувачів, які будуть користуватись збільшиться: протягом першого року – на 200 користувачів, протягом другого року – на 150 користувачів, протягом третього року – 100 користувачів. Реалізація інформаційної технології до впровадження результатів наукової розробки складала 700 користувачів, а прибуток, що отримував розробник до впровадження результатів наукової розробки – 200 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту $\Delta П_1$ протягом першого року складатиме:

$$\Delta П_1 = 20 \cdot 700 + (200 + 20) \cdot 200 = 58000 \text{ грн.}$$

Протягом другого року:

$$\Delta П_2 = 20 \cdot 700 + (200 + 20) \cdot (200 + 150) = 91000 \text{ грн.}$$

Протягом третього року:

$$\Delta П_3 = 20 \cdot 700 + (200 + 20) \cdot (200 + 150 + 100) = 113000 \text{ грн.}$$

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $Е_{абс}$вкладених інвестицій розраховується за формулою:

$$Е_{абс} = (ПП - PV), \quad (4.10)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

т – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки 2, 3,4.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 4.1.

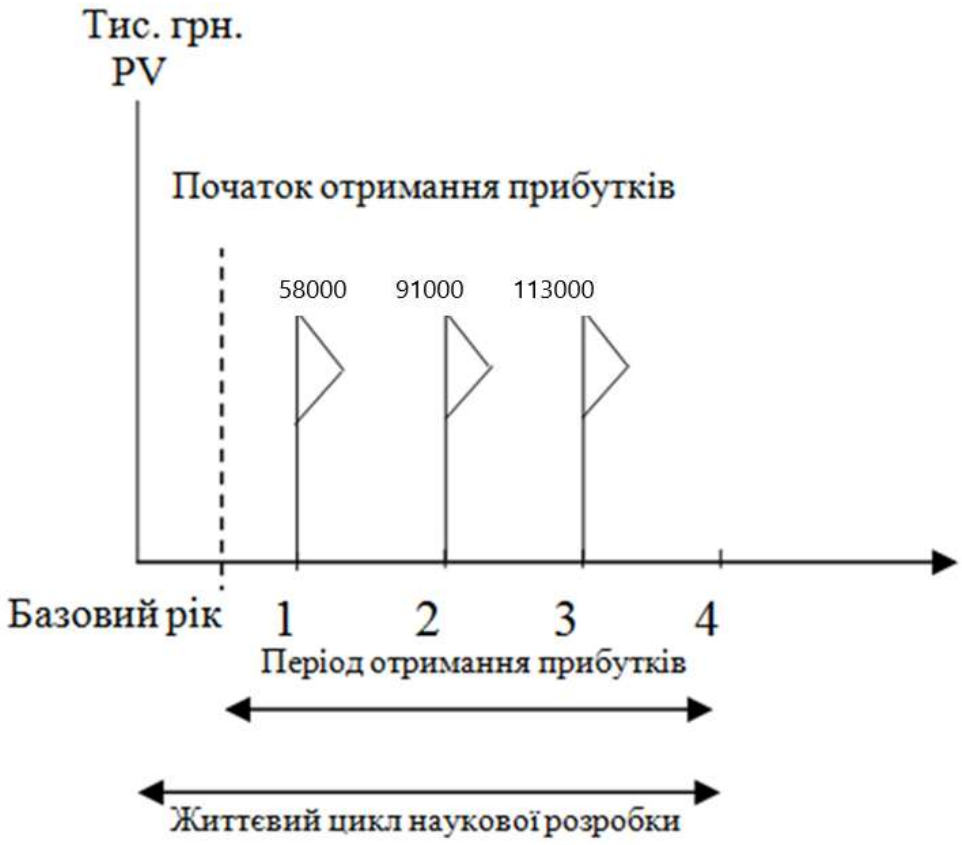

Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо вартість чистих прибутків за формулою:

$$\mathbf{ПП} = \sum_{\mathbf{1}}^{\boldsymbol{m}} \frac{\mathbf{\Delta П}_{\boldsymbol{i}}}{(\mathbf{1}+\boldsymbol{\tau})^{\boldsymbol{t}}} \quad (4.11)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки.

Отже, розрахуємо вартість чистого прибутку:

$$ПП = \frac{32441{,}4}{(1+0{,}1)^0} + \frac{58000}{(1+0{,}1)^2} + \frac{91000}{(1+0{,}1)^3} + \frac{113000}{(1+0{,}1)^4} = 225925{,}44 \text{ (грн.)}$$

Тоді розрахуємо $E_{абс}$:

$$Е_{абс} = 225925{,}44 - 32441{,}4 = 193484{,}04 \text{ грн.}$$

Оскільки $E_{абс} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $Е_в$ за формулою:

$$Е_в = \sqrt[Т]{1 + \frac{Е_{абс}}{PV}} - 1 \quad (4.12)$$

де $E_{абс}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій PV = ЗВ, грн;

$T_ж$ – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_в = \sqrt[3]{1 + \frac{193484,04}{32441,4}} - 1 = 0,90 \text{ або } 90\ \%$$

Далі, розраховану величина $E_в$ порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{мін}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{мін}$ визначається за формулою:

$$\tau = d + f,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні d = 0,2;

f – показник, що характеризує ризикованість вкладень, величина f = 0,1.

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки $E_в$ = 90% > $\tau_{мін}$ = 0,3 = 30%, то у інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_в}$$

$$T_{ок} = \frac{1}{0,9} = 1,11 \text{ року}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

4.5 Висновок

На основі зроблених підрахунків в економічній частині магістерської кваліфікаційної роботи досягнуті наступні результати:

– визначено, що рівень комерційного потенціалу розробки є високим.

– витрати на розробку та її впровадження складають 32441,4 грн.;

– абсолютний ефект від впровадження результатів нашої розробки протягом 3-х років складе 113000 грн.

– вартість інвестицій, що можуть бути вкладені в нашу розробку становить 398906 тис. грн;

– термін окупності системи, що розробляється складає 1,11 року, що вписується в задані часові рамки та є показником доцільності розробки.

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи реалізовано інформаційну технологію для аналізу англомовного тексту на наявність сталих мовних конструкцій яку засновано на сумісному використанні методу Байєсівського класифікатора та алгоритм підрахунку TF-індексу в поєднанні з алгоритмами логіко-лінгвістичного моделювання для аналізу текстового документу, що забезпечмло підвищення швидкості аналізу англомовного тексту.

У першому розділі проведено аналіз сучасних методів обробки тексту та контент-аналізу, а також програм-аналогів, які використовуються у для вирішення задач аналізу, що який показав актуальність проблеми, що полягає в необхідності створення зручного та ефективного інструменту для аналізу текстових документів, що зробить ефективнішим процес вивчення сталих мовних конструкцій англійської мови. Здійснено коротку порівняльну характеристику знайдених програм-аналогів, за рахунок чого визначено їх переваги та недоліки, що дало змогу обрати за прототип програму HAMLET. Проведений аналіз сталих мовних конструкцій в англомовних текстах надав можливість визначити принципи подання тексту та відбору ознак, що дозволило сформулювати проблему аналізу англомовних текстових документів та здійснити постановку задачі.

Другий розділ присвячено математичним методам та алгоритмам, що використовуються для задач аналізу текстів. Розроблено математичну модель для аналізу англомовного тексту на наявність сталих мовних конструкцій, яка заснована на методах підрахунку TF-індексу та Байєсівського класифікатора, що за рахунок простоти і швидкості роботи алгоритмів призводять до швидкого та правильного вирішення даної задачі. Запропоновано структуру інформаційної технології, спроектовано схему алгоритму роботи розроблюваної програми та ER-діаграму.

Третій розділ присвячено проектуванню та розробці інформаційної технології аналізу англомовного тексту на наявність сталих мовних конструкцій.

Обґруновано вибір мови програмування Java, яка є платформо незалежною, об’єктно-орієнтовною та має високу надійність; розглянуто особливості середовища Eclipse.

На основі отриманих результатів у попередніх розділах створено UML-діаграму класів. Відповідно до математичної моделі, схеми алгоритму та UML-діаграми класів реалізовано інформаційну технологію аналізу англомовного тексту на наявність сталих мовних конструкцій.

Проведено тестування програми, яке довело доцільність проектування та використання розробленої інформаційної технології у порівнянні з обраним прототипом за рахунок збільшення швидкості обробки тексту майже на 3,5%, завдяки перевагам розробленої моделі.

У четвертому розділі здійснено оцінювання комерційного потенціалу розробки. Проведено технологічний аудит із залученням експертів. Згідно висновків експертів, рівень комерційного потенціалу розробки вище середнього. Головним конкурентом розробки є програма «HAMLET». Здійснено прогнозування витрат на виконання науково-дослідної роботи. Розраховано витрати на заробітну плату та амортизаційні відрахування, витрати на силову електроенергію. Загальні витрати становлять 32441,4 грн.

За результатами досліджень опубліковано тези доповіді у збірнику праць всеукраїнської науково-практичної інтернет-конференції [1] та XLVII Науково-технічній конференції факультету інформаційних технологій та комп’ютерної інженерії (2018) [2]. Подано заяву про реєстрацію авторського права на твір.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Миколюк І. О. Дослідження та оцінка основних методів для задачі аналізу англомовного тексту на наявність сталих мовних конструкцій/ Молодь в науці: дослідження, проблеми, перспективи (МН-2019). URL: https://conferences.vntu.edu.ua/index.php/mn/mn2019/paper/view/8178
2. Миколюк І. О., Суприган О. І. Аналіз методів розпізнавання мовлення/ Конференції ВНТУ електронні наукові видання, XLVII Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2018). URL: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2018/paper/view/5144
3. Бабенко Л. Г. Лингвистический анализ художественного текста / Л. Г. Бабенко, И. Е. Васильев, Ю. В. Казарин. – Екатеринбург, 2000. – 125 с.
4. Большакова Е. И. Автоматическая обработка текстов на естественном языке и компьютерная / Е. И. Большакова, Е. С. Клишинский. – М. : МИЭМ, 2011. – 272 с.
5. Семчинський С.В. Семантична інтерференція мов / С.В. Семчинський. – К.: Вища школа, 1974. – 83с.
6. Основные концепции и подходы при создании контекстно-поисковых систем на основе реляционных баз данных. URL: https://www.osp.ru/os/2003/12/183694/
7. Большакова Е. И. Автоматическая обработка текстов на естественном языке и компьютерная / Е. И. Большакова, Е. С. Клишинский. – М. : МИЭМ, 2011. – 272 с.
8. Яцко, В.А. Некоторые проблемы разработки современных систем автоматического реферирования текста [Текст] / В.А. Яцко, Т.Н. Вишняков // Научно-техническая информация. Сер.2. - 2007.
9. Geiß J. Latent semantic sentence clustering for multi-document summarization / J. Geiß. – Cambridge, 2011. – 156 p.

10. Герд А.С. Автоматизация в лексикографии и словари-конкордансы / А.С.Герд // Филологические науки.- 1981.- №1.- С.72-78.
11. Шалак В.И. Современный контент-анализ. Приложения в области: политологии, психологии, социологии, культурологии, экономики, рекламы. — М.: Омега - Л, 2009. — 272 с.
12. Барсегян А. А. Технология анализа данных: Data Mining, Visual Mining, Text Mining, OLAP / А. А. Барсегян. – Санкт-Петербург : БХВ-Петербург, 2007. – 384 с.
13. Программы лингвистического анализа и обработки текста. URL: http://asknet.ru/analytics/programms.htm
14. Alexa, M., Zuell, C. Text Analysis Software: Commonalities, Differences and Limitations: The Results of a Review.// Quality & Quantity — No. 34 — 2000. — Pp. 299-321.
15. Kwalitan. URL: http://www.kwalitan.nl/engels/
16. WordStat. URL: http://www.provalisresearch.com/wordstat.html
17. Подбор программ статистической обработки текста URL: http://www.bues.ru/blog/express/8.php
18. Textanz. URL: access mode: http://www.textanz.com/
19. HAMLET II 3.0 – computer assisted text analysis. URL: http://apb.newmdsx.com/hamlet2.html
20. Кравець Р.Б. Застосування багатозначної логіки для інтелектуального аналізу даних [Текст] / Кравець Р.Б., Шаховська Н.Б. // Вісник Національного університету «Львівська політехніка». – Л.: Вид-во Національного ун-ту «Львівська політехніка», 2002. – №468: Комп’ютерна інженерія та інформаційні технології. – С. 58–65.
21. Катренко А.В. Методи видобування знань в інформаційних системах [Текст] / Катренко А.В. // Вісник Державного університету «Львівська політехніка». – Л.: Вид-во Державного ун-ту «Львівська політехніка», 1999. – № 383: Інформаційні системи та мережі. – С. 96–101.

22. Захаров, В.П. Автоматическое выявление терминологических словосочетаний /В.П. Захаров, М.В. Хохлова //Структурная и прикладная лингвистика. Вып.10. - Санкт-Петербург: Изд-во С.-Петерб. ун-та, 2014. – С. 182-200.
23. Yatsko, V.A. TF*IDF Revisited / V.A. Yatsko // International journal of computational linguistics and natural language engineering. – 2013. – Vol. 2, Issue 6. – P. 385-387. – URL: http://www.ijclnlp.org/ vol2issue6/paper60.pdf.
24. Наївний баєсів класифікатор URL: https://uk.wikipedia.org/wiki/ Наївний_баєсів_класифікатор
25. Маннинг К.Д., Рагхаван П., Шютце Х. Введение в информационный поиск.— М.: Изд-во «Вильямс», 2011.
26. Java основні відомості URL: https://uk.wikipedia.org/wiki/Java
27. А. Н. Васильев − Java. «Объектно-ориентированое програмирование».
28. Порівняння мов програмування URL: https://ru.wikipedia.org/wiki/Сравнение_языков_программирования
29. Eclipse - середовище розробки Java URL: http://hi-news.pp.ua/tehnka-tehnologyi/3971-eclipse-seredovische-rozrobki-java.html
30. Eclipse основні відомості URL: https://ru.wikipedia.org/wiki/Eclipse_(среда_разработки)
31. Кириллов В.В. Структуризованный язык запросов (SQL). - СПб.: ИТМО, 1994. - 80 с.
32. Основні оператори sql URL: http://hi-news.pp.ua/kompyuteri/5628-osnovn-operatori-sql.html