Міністерство освіти і науки України
Державний університет "Одеська політехніка"
Навчально-науковий інститут комп'ютерних систем
Кафедра системного програмного забезпечення

Вонгуе Нгангом Франсуа Жералдін,
студент групи АС-161

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Програма для проведення частотного аналізу
технічних текстів англійською мовою

Спеціальність:
121 – Інженерія програмного забезпечення

Освітня програма:
Інженерія програмного забезпечення

Керівник:
Зіноватна Світлана Леонідівна,
канд. техн. наук, доцент

Одеса – 2021

ЗМІСТ

Міністерство освіти і науки України
Державний університет "Одеська політехніка"
Навчально-науковий інститут комп'ютерних систем
Кафедра системного програмного забезпечення

Рівень вищої освіти: другий (магістерський)
Спеціальність: 121 – Інженерія програмного забезпечення
Освітня програма: Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Любченко В.В.
«___» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Вонгуе Нгангом Франсуа Жералдіна, група АС-161

1. Тема роботи: Програма для проведення частотного аналізу технічних текстів англійською мовою
Керівник роботи: Зіноватна Світлана Леонідівна,канд. техн. наук, доцент
затверджені наказом ректора від «25» жовтня 2021 р. № 374-в.
2. Зміст роботи: проведення критичного аналізу існуючих рішень для обробки тексту та частотного аналізу, порівняльна характеристика програмних аналогів; розробка програмного аналізатору, побудова моделі нормалізації англійських слів; специфікація вимог до частотного аналізатора; проектування частотного аналізатору; вибір програмних інструментів розробки; програмна реалізація функціоналу та інтерфейсу застосування; тестування моделі ламетизатора, функціональності системи та ступеня зручності використання частотного аналізатора.
3. Перелік ілюстративного матеріалу: згідно зі слайдами презентації.

4. Консультанти розділів роботи

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

5. Дата видачі завдання: «01» вересня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання	Примітка
1	Аналіз існуючих рішень	2.09.2021 – 16.09.2021	вик.
2	Специфікація вимог до програми	17.09.2021 – 30.09.2021	вик.
3	Проектування програмної системи	01.10.2021 – 17.10.2021	вик.
4	Програмна реалізація системи	18.10.2021 – 11.11.2021	вик.
5	Тестування системи	14.11.2021 – 28.11.2021	вик.
6	Оформлення пояснювальної записки та графічного матеріалу	15.11.2020 – 29.11.2021	вик.

Здобувач вищої освіти _____ Вонгуе Нгангом Франсуа Жералдін

Керівник роботи _____ С. Л. Зіноватна

АНОТАЦІЯ

Метою роботи є підвищення зручності виконання частотного аналізу слів у англійських текстах завдяки розробці аналізатора з використанням нормалізації слів.Методи розробки базуються на технологіях мови програмування Python, безкоштовному інтегрованому середовищі PyCharm 2021.1.2, базі даних MySQL, системі контролю версій Git, з використанням словників баз WordNet та StarDict. Як результат роботи розроблено програмнезастосування, що дозволяє проводити частотний аналіз.

Ключові слова: частотний аналіз, англійський текст, Python, PyCharm, MySQL, Git, WordNet, StarDict.

ABSTRACT

The aim of the work is to increase the convenience of performing frequency analysis of words in English texts by developing an analyzer using word normalization. Development methods are based on Python programming language technologies, free PyCharm 2021.1.2 integrated environment, MySQL database, Git version control system, using dictionaries of WordNet and StarDict databases. As a result of work the software application allowing to carry out the frequency analysis is developed.

Keywords: frequency analysis, English text, Python, PyCharm, MySQL, Git, WordNet, StarDict.

ВСТУП

Під неструктурованими даними розуміються дані, які не мають чіткої моделі представлення та для яких не прописані правила перетворення для переходу від одного стану до іншого.Робота з неструктурованими даними по праву вважається однією з найбільш важких.У порівнянні з поданням даних в базах даних (як реляційних, так і нереляційних), вилучення даних з неструктурованих інформаційних блоків веде до підвищення трудомісткості процесу і зростання числа помилок [1].

При роботі з неструктурованими даними слід брати до уваги два аспекти. По-перше, хоча структура тексту може біти формально не визначена, вона все ж може бути присутня, але без інформування користувача про її наявність та певні ознаки методи обробки не надають бажаних результатів. По-друге, якщо структура тексту визначена, але не є корисною для вирішення певного завдання, то такий текст теж може вважатись як неструктурований.

Найбільш відомими методами обробки неструктурованих текстів є обробка природної мови (Natural language processing - NLP), інтелектуальний аналіз даних (Data mining) та текстова аналітика (Text mining). У даній кваліфікаційній роботі будуть вирішуватись завдання стосовно напрямку обробки природної мови.

Якщо мати можливість поставити у відповідність певному тексту частотний набір унікальних слів, що входять до нього, то можливо вирішення наступних проблем: ефективне вивчення слів англійської мови для розуміння змісту текста; створення семантичного ядра сайта, що рекламує товари та послуги, описані у тексті; класифікація текста та ін.

Все вищесказане показує *доцільність та актуальність* розробки програмного застосування, що дозволяє виконувати частотний аналіз текстів, наданих англійською мовою.

Метою роботи є підвищення зручності виконання частотного аналізу слів у англійських текстах завдяки розробці аналізатора з використанням нормалізації слів.

Для досягнення мети були поставлені та успішно вирішені *наступні завдання*:

– дослідження існуючих рішень та проблем обробки неструктурованих текстів;

– визначення принципів нормалізації англійських слів у тексті та побудова моделі нормалізації;

– визначення способу верифікації результатів;

– проектування програмного застосунку з використанням створеної моделі;

– програмна реалізація застосунку;

– тестування працездатності програмної системи та визначення ступеня зручності її використання.

Даний звіт є результатом виконання кваліфікаційної роботи магістра. Він містить огляд сучасного стану проблеми, власну розробку з урахуванням проектування, програмування та тестування системи, та лістинг програмного коду.

1 КРИТИЧНИЙ АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ДЛЯ ОБРОБКИ ТЕКСТІВ ТА ЧАСТОТНОГО АНАЛІЗУ

1.1 Проблеми обробки неструктурованих текстів

На сьогоднішній час стрімко зростають обсяги інформації, які стають доступними людям завдяки технологіям, що визначають цифрову епоху.Звичайні користувачі, які не володіють знаннями про принципи дії пошукових систем і класифікаторів, теж хочуть мати можливість якісно обробляти дані і отримувати від них максимальну користь в залежності від розв'язуваних ними завданьта персональних можливостей користувачів [2].

Користувачі мають потребу в автоматичних інструментах, які могли б давати їм відповіді на питання за результатами аналізу вихідних даних. У разі, коли вихідні дані представлені неструктурованих текстом, це веде до певних труднощів.

Традиційно автоматичні інструменти обробки тексту повинні вміти наступне:

– знаходити відповіді на поставлені користувачем питання і аргументувати ці відповіді;

– спеціальним чином організовувати текст з використанням різних маркерів, позначок, тегів для можливості виконання автоматичних або напівавтоматичних дій над текстом: перевірка, переклад, складання анотації;

– мати можливість виконувати два попередніх пункту без суттєвої переробки архітектури інструментарію при значному зростанні обсягів вихідних неструктурованих текстів.

Як показують аналітичні звіти, розподіл інформації, з якою працює користувач, за типами виглядає наступним чином:

– текстова інформація - 32%;

– зображення - 24%;

– аудіо - 14%;

– відео - 19%;

– інші (комбіновані) дані - 11%.

Крім того, існує безліч інструментів, які перетворюють аудіодані у текст, додають субтитри до повідомлень у відеоформаті чи додають слова-мітки до зображень, сигналів, відео, тощо. Все це свідчить про дуже великі об’єми текстової інформації, які повинні бути оброблені за допомогою сучасних програмних інструментів.

Розглянемо найбільш популярні завдання, які вирішуються на неструктурованому тексті.

Дуже поширеним є завдання виконання частотного аналізу. В ході виконання аналізу підраховується частота входження певних слів (або всіх слів, які зустрічаються) для заданого початкового тексту.

При вирішенні завдання перекладу текстових повідомлень з однієї мови на іншу потрібно виконати встановлення між орігінальним текстом та його перекладом. При цьому потрібно взяти до уваги контекст перекладу та ситуації, у яких деякі елементи оригінального тексту можуть залишитись не перекладеними, або навпаки мати багато варіантів перекладу. Внутрішній механізм, що є основою перекладача, знаходить однакові (еквівалентні) мовні одиниці – лексеми – та створює відповідну заміну. Створений переклад потребує перевірки на якість, тому аналізується як форма, так і зміст отриманого перекладу. Чим більш вони співпадають з формою та змістом ориганілу, тим більш релевантним є перекладач.

Для успішної роботи перекладача потрібні прикладні аспекти, що забезпечують використання довідників та словарів. Для кожної предметної області, особливо вузько направленої предметної області, існують словарі термінів, використання яких надає варіанти перекладу елементів оригінального тексту з найвищою оцінкою та загалом підвищує релевантність перекладу неструктурованих текстів [3].

Популярним в даний час напрямком є розробка систем типу «питання-відповідь». При цьому система приймає сформульоване користувачем питання на

природній мові, і видає користувачеві відповідь. Особлива цінність таких систем полягає в тому, що при достатній кількості вихідних даних користувач отримує істинну відповідь, а не просто весь перелік можливих варіантів, з яких необхідно єдиний вірний варіант вибирати потім самому. При цьому система повинна спочатку розібрати питання та визначити що ж питається; потім знайти потенційні відповіді; далі виконати оцінку отриманих відповідей; наприкінці повіднути відповідь за найвищою оцінкою. При цьому можуть бути використані методи datascience, у тому числі методи класифікації [4 – 6].

Для всіх цих напрямків потрібно виконання багатьох функції. Базовою функцією, що входить до усіх напрямків, є виділення слів. Саме цю функцію буде застосовано при розробці програмного частотного аналізатору у даній кваліфікаційній роботі.

1.2 Застосування частотного аналізу текстів у практичних задачах

Будемо розуміти частотний аналіз як обчислення кількості входжень кожного слова у заданий текст (чи набір текстів) та подання результатів обчислень у вигляді

$$W_i = <name_i, count_i>, i = 1..N, \quad (1.1)$$

де N – кількість унікальних слів у тексті.

Доцільним є подання пар W_i відповідно до спадання значень $count_i$.Приклад результатів частотного аналізу наведений на рис. 1.1. Зазвичай, чим більш об’ємний та різноманітним є текст, тим більшою буде таблиця, що містить результат роботи частотного аналізатора. Розглянемо типові приклади використання частотного аналізатору для вирішення практичних задач [7]. У якості першого прикладу наведемо вивчення англійської мови через виділення та вивчення найбільш розповсюджених у наданому тексті слів.

Всього існує п’ять найбільш розповсюджених методів вивчення іноземної мови:

#	Слово	Кіль-кість
1	vpn	29
2	huawei	20
3	lite	13
4	keepsolid	11
5	device	9
6	download	5
7	turbo	5
8	protect	4
9	plan	4
10	app	4

Рисунок 1.1 – Приклад виконання частотного аналіза

1) метод «граматика та переклад», який включає вивчення правил граматики та слів, та застосування їх для перекладу текстів іноземною мовою;
2) «натуральний» метод, який працює через повторення та виправлення, та не включає окремого вивчення граматики;
3) метод «Каллан» - швидкі відповіді на поставлені запитання з використанням встановлених мовних паттернів;
4) «мовне занурення», що показує добрі результати при переїзді користувача до країни, у якій мешканці розмовляють потрібною користувачу мовою;
5) «коммунікативний» метод, який є узагальненням усіх попередній методів.

Як можна побачити, усі методи, крім «мовного занурення» передбачають попередню підготовку набору слів, з використанням яких може бути пройдена

певна тема. Тому вибір найбільш розповсюджених слів, які використовуються у відповідних до теми текстах, є важливим завданням.

Другим типовим прикладом практичного використання частотного аналізу тексту є побудова семантичного ядра сайта. «Семантичне ядро сайту (СЯ) — це упорядкований набір слів, їх морфологічних форм і словосполучень, які найбільш точно характеризують вид діяльності, товари або послуги, що пропонує вебсайт. Семантичне ядро має центральне ключове слово, як правило високочастотне, і всі інші ключові слова в ньому ранжуються у напрямку зниження частоти спільного використання з центральним запитом до загальної колекції документів. Таким чином, семантичне ядро представляється у вигляді семантичного графа, де довжини його ребер обернено пропорційні частоті спільної згадки.

Ключові слова (пошукові запити) семантичного ядра підбираються шляхом аналізу послуг або товарів компанії, аналізу статистики запитів, статистики сайту і вмісту конкуруючих сайтів. Склад семантичного ядра повинен максимально відповідати уявленням цільових відвідувачів веб-сайту про ту інформацію яка на ньому, на їхню думку, має бути присутня.»

У семантичне ядро *SemanticCore*входить множина

$$SemanticCore = \{HFR, MFR, LFR\}, \quad (1.2)$$

де *HFR* - високочастотні запити, що є підмножиною середньочастотних $HFR \subset MFR$,

MFR - середньочастотні запити, що є підмножиною низькочастотних $MFR \subset LFR$;

LFR – низькочастотні запити.

На рис. 1.2 наведено складові частини створення опису продукту, який може бути як комерційною розробкою, так і соціальним проектом. При необхідності просування продукту на ринку може знадобитись розробка веб-сайту. Як можна побачити, для цього визначається як функціонал продукту, так і словники предметної області. Семантичне ядро веб-сайту повинно враховувати його найбільш важливі ключові слова.

Рисунок 1.2 – Складові частини фази створення опису продукту

Якщо маємо контентне наповнення веб-сайту, то виділення найбільш використовуваних слів може допомогти з визначенням його семантичногоядра. В залежності від специфіки бізнес-моделі ресурсу веб-сайту цей підхід може бути більш чи менш придатним.

1.3 Огляд існуючих програмних аналогів

Розглянемо типові програмні аналоги, що дозволяють виконувати обробку неструктурованого тексту та проводити частотний аналіз слів тексту. Результати порівняльного аналізу наведені у табл. 1.1.

WordStat - сервіс, що призначений для оцінювання інтересу користувача до різних областей та тематик і підбору ключових слів для низько, середньо- та високочастотних запитыв для SEO-оптимізації сайтів [8]. Можна переглядати залежність запитів від географічного розташування користувачів. Має застарілий інтерфейс та середню зручність використування.

Pymorphy2 – морфологічний аналізатор, бібліотека мови програмування Python, що дозволяє автоматично виконувати нормалізацію слів [9]. Працює під версіями 2.7 та 3.5+. Крім нормалізації, вміє ставити слово у потрібну форму. Нажаль, працює лише для української та російської мов. Для використання потрібно розробка власного програмного модуля, що не є доступним рядовому користувачу, тому знижує зручність користування цим засобом.

Таблиця 1.1 – Порівняльна характеристика програмних продуктів

Назва продукту	Властивості продукту				
	Можливість роботи з англійською мовою	Можливість виконання частотного аналізу	Можли-вість нормалі-зації слів	Досту-пна ціна	Зручність вико рис-тання
WordStat	+	+	-	+	середня
Pymorphy2	-	-	+	+	середня
Text-Analizer від StatSoft	+	+	+	-	висока
TextOBRAZ	+	+	-	+	середня
Розроблюваний програмний продукт	+	+	+	+	+

Продукт Text-Analizerвід компанії StatSoft дозволяє працювати з великим набором текстових файлів у форматах .txt, .doc,.pdf та .xml. Він містить багато функціональних інструментів, але має високу ціну. Доступною для безкоштовного використання є тільки Trial-версія. Компанією StatSoft передбачено навчання користувачів володінню продуктом Text-Analizer [10].

TextOBRAZ – інструмент для оптимізації текстів, який призначений для визначення кількості повторень у певному тексті ключових слів та фраз. Можна включати чи відключати врахування пробілів. Незважаючи на деяку загальність інтересів з розроблюваним у даній роботі продуктом, TextOBRAZ має метою виключення знайдених повторень. Крім того, аналізований текст потрібно набирати вручну чи вставляти з буферу обміну, тому зручність цього засобу є теж середньою [11].

Як можна побачити, жоден аналог не відповідає усім означеним потребам, що доказує необхідність розробки власного застосування для проведення частотного аналізу.

1.4 Висновки до розділу

У першому розділірозглянуто напрямки обробки неструктурованих текстів. Формально визначено результат роботи частотного аналізатора. Наведено практичні приклади застосування результатів частотного аналізу текстів англійською мовою.

Проведено огляд та аналіз програмних аналогів за вимогами можливості роботи з англійською мовою, виконання частотного аналізу, нормалізації англійських слів, доступності використання продукту стосовно ціни, а також аналіз ступеня зручності, підвищення якого заявлено як основна мета розробки. Наведено висновок про актуальність зазначеної розробки.

2 РОЗРОБКА ПРОГРАМНОГО АНАЛІЗАТОРУ

2.1 Принципи роботи частотного аналізатора

У загальному вигляді алгоритм роботи аналізатора показано на рис. 2.1.

Рисунок 2.1 – Загальний алгоритм роботи частотного аналізатора

Розглянемо процедуру парсингу тексту англійською мовою на окремі слова. Для цього зручно використовувати механізм регулярних виразів.

Отже, почнемо з простого, отримаємо файли, розпарсити їх на слова, підрахуємо, відсортуємо, і видамо результат.

Для початку складемо регулярний вираз для пошуку англійських слів в тексті.

Спочатку розглянемо прості англійські слова, на кшталт «mother»чи «Law». Регулярний виразExpr, призначений для цього, повинен шукати одину або більше букв англійського алфавіту:

$$Expr = ([a\text{-}zA\text{-}Z] +) \quad (2.1)$$

Крім простих, у англійській мові є складені слова, що утворюються з простих за допомогою тире, наприклад, «mother-in-law». Для розпізнавання цього слова потрібно розбити його на підвирази:

- «mother-»
- «in-»
- «law»

Це означає, що можуть йти послідовності, складені з англійських малих і великих букв, після яких йде «тире». Після «тире» без пробілу може йти:

- така ж послідовність з «тире», якщо це просте слово не є останнім у складеному;
- послідовність з англійських малих і великих букв, якщо це просте слово є останнім у складі складеного слова.

Регулярний вираз Expr, що вміє розпізнавати таке складене слово, визначається як:

$$Expr = (([a\text{-}zA\text{-}Z] + \text{-}?) * [A\text{-}zA\text{-}Z] +) \quad (2.2)$$

Якщо вважати, що у виразі проміжнийпідвираз може бути присутній не обов’язково, то регулярний вираз Exprнабуває наступного вигляду:

$$Expr = ((?: [A-zA-Z] + -?) * [A-zA-Z] +) \quad (2.3)$$

Нам ще залишилося включити в вираження слова з апострофом виду «did not = didn’t». Для цього замінимо в першому подвираженія "-?" на "[- ']?".

Розглянемо варіант, коли у виразі є апостроф, на який цей вираз закінчується. Це може бути у разі застосування присвійного займенника для множини, наприклад: «children’» або «sisters’». У такому разі регулярний вираз Expr має вигляд:

$$Expr =([A-zA-Z] +[- ']?) \quad (2.4)$$

Таким чином, зупиняємось на наступному регулярному виразі Expr, який і будемо використовувати:

$$Expr = ((?: [A-zA-Z] + [- ']?) * [A-zA-Z] +) \quad (2.5)$$

2.2 Побудова моделі нормалізації англійських слів

2.2.1 Особливості нормалізації дієслів. При нормалізації дієслів слід розрізняти правильні дієслова від неправильних. Для правильних дієслів потрібно оброблювати закінчення, тобто перетворювати дієслова за правилом («V» - «Verb»):

<BaseV><End1V>→<BaseV><End2V>, (2.6)

де <BaseV> - незмінна частина дієслова (verb),

<End1V> - закінчення для ненормалізованого дієслова;

<End2V> - закінчення для нормалізованого дієслова.

Між <End1> та <End2> можна встановити пари, наведені у табл. 2.1.

Таблиця 2.1 – Правила зміни закінчень при нормалізації правильних дієслів

№ п/п	<End1>	<End2>
1	"s"	""
2	"ies"	"y"
3	"es"	"e"
4	"es"	""
5	"ed"	"e"
6	"ed"	""
7	"ing"	"e"
8	"ing"	""

При неможливості відтворити дієслово за допомогою правил для правильних дієслів потрібно застосувати правила для неправильних дієслів:

IF (NOT SUCCESS) (f(Verb1→Verb2) | Verb1= <BaseV><End1V>,
Verb2= <BaseV><End2V>)
THEN f(Irregular_Verb1 →Irregular_Verb2) (2.7)

де Irregular_Verb1 – форма неправильного дієслова, що зустрілась у тексті, який аналізується;

Irregular_Verb2 – первинна форма відповідного неправильного дієслова.

У загальному випадку модель Mv нормалізації дієслова виглядає наступним чином:

$$Mv = (Rules\ 1\ \&\&\ v \in Regular \parallel\ Rules\ 2\ \&\&\ v \in !Regular), \quad (2.8)$$

де Regular - множина правильних дієслів;

Rule1 – правила перетворення правильного дієслова у нормалізовану форму;

Rule2 – правила перетворення неправильного дієслова у нормалізовану форму.

2.2.2 Принципи нормалізації іменників.При нормалізації іменників теж потрібно розрізняти правильні іменники від неправильних. Для правильних іменниківвиявлено, що потрібно оброблювати закінчення, тобто перетворювати іменники за правилом («N» - «Noun»):

<BaseN><End1N>→<BaseN><End2N>, (2.9)

де <BaseN> - незмінна частина іменника (noun),

<End1N> - закінчення для ненормалізованого іменника;

<End2N> - закінчення для нормалізованого іменника.

Між <End1N> та <End2N> можна встановити пари, наведені у табл. 2.2.

Таблиця 2.2 – Правила зміни закінчень при нормалізації правильних іменників

№ п/п	<End1>	<End2>
1	"s"	""
2	"’s"	""
3	"’"	""
4	"ses"	"s"
5	"xes"	"x"
6	"zes"	"z"
7	"ches"	"ch"
8	"shes"	"sh"
9	"men"	"man"
10	"ies"	"y"

При неможливості відтворити іменник за допомогою правил для правильних іменників потрібно застосувати правила для неправильних іменників:

IF (NOT SUCCESS) (f(Noun1→Noun2) | Noun1= <BaseN><End1N>,

Noun2= <BaseN><End2N>)

THEN f(Irregular_Noun1 →Irregular_Noun2). (2.10)

де Irregular_Noun1 – форма неправильного іменника, що зустрілась у тексті, який аналізується;

Irregular_Noun2 – первинна форма відповідного неправильного іменника.

У загальному випадку модель Mn нормалізації іменників виглядає наступним чином:

$$Mn = (Rules\ 1\ \&\&\ v \in Regular \parallel\ Rules\ 2\ \&\&\ v \in !Regular), \quad (2.11)$$

де Regular - множина правильних іменників;

Rule1 – правила перетворення правильного іменника у нормалізовану форму;

Rule2 – правила перетворення неправильного іменника у нормалізовану форму.

2.2.3 Представлення прислівників у нормальній формі. Особливих правил для нормалізації прислівників у англійській мові немає, тому основним завданням є визначення, чи є певний прислівник правильним або ні.

У загальному випадку модель Mav («av» - «adverb») нормалізації іменників виглядає наступним чином:

$$Mav = (Vocabulary1(av), av \in Regular \parallel Vocabulary2(av), av \in !Regular), \quad (2.11)$$

де Regular - множина правильних прислівників;

Vocabulary1 – словник правильнихприслівників;

Vocabulary2 – словник неправильних прислівників.

2.2.4 Правила нормалізації прикметників. При нормалізації прикметників потрібно визначити їх форму. Якщо прикметник находиться у вищому ступені (comparative degree) чи у найвищому ступені (superlative degree), то потрібно перевести його до звичайного ступеня, його ще називають позитивним (positive

degree). Для цього потрібно оброблювати закінчення, тобто перетворювати прикметники за правилом («Aj» - «Adjective»):

<BaseAj><End1Aj>→<BaseAj><End2Aj>, (2.12)

де <BaseAj> - незмінна частина прикметника,

<End1Aj> - закінчення для ненормалізованого прикметника;

<End2Aj> - закінчення для нормалізованого прикметника.

Між <End1Aj> та <End2Aj> можна встановити пари, наведені у табл. 2.3.

Таблиця 2.3 – Правила зміни закінчень при нормалізації прикметників

№ п/п	<End1>	<End2>
1	"er"	""
2	"er"	"e"
3	"est"	""
4	"est"	"e"

Mav = f(Aj1→Aj2) | Aj1= <BaseAj><End1Aj>,

Aj2= <BaseAj><End2Aj> (2.13)

2.3 Застосування словників

Для того, щоб мати можливість застосовувати до слів певні правила з метою їх нормалізації, потрібно мати словники, у яких слова розмічені відповідно до частин мови. Для цього у роботі застосовано структури даних, надані великою лексичною базою даних англійської мови, розробленою представниками Прінстонського університету [12].

Слова різних частин мови - іменники, дієслова, прикметники та прислівники - об’єднані у набори пізнавальних синонімів (синсети), кожен із яких виражає

окреме поняття, що може бути застосованим для подальшого перекладу значень слів, але не є корисним для даної роботи.Ці синсети визначені та взаємопов’язані між собою за допомогою певних понятійно-семантичних, та також лексичних відносин.

База словників WordNetмістить файли саме для роботи з дієсловами, іменниками, прислівниками та прикметниками. Для кожної з перелічених частин мови є по два файла: index.pos та data.pos, де pos - це noun, verb, adj та adv.«Кожен індексний файл - це алфавітний список усіх слів, знайдених у WordNet у відповідній частині мови.

Слова у файлі індексу вказуються лише з малої літери, незалежно від того, як вони були внесені у файли лексикографа. Це дозволяє здійснювати пошук у базі даних без урахування регістру.

Файли зі списком винятків, pos.exc, використовуються, щоб допомогти морфологічному процесору знаходити базові форми від неправильних змін форми слова.» Це стосується, наприклад, неправильних дієслів, які змінюються за часом не відповідно загальним правилам, чи іменників, які утворюють форму множини шляхом, що відрізняється від загальних правил.

Файли баз даних (словники) представлені у текстовому форматі ASCII, що дозволяє легко застосовувати їх як при машинній обробці, так і при сприйнятті людиною.

Незважаючи на те, що існує онлайн-версія WordNet, можливо завантаження та застосування саме словників англійських слів. Вони можуть бути використані при розробці проекту мовою програмування високого рівня, наприклад, Pythonз застосуванням будь-якого сучасного фреймворку.

2.4 Спосіб верифікації результатів роботи частотного аналізатора

Виконання саме підрахунку кількості слів не є складним завданням, та виконується безпомилково. Необхідність перевірки та верифікації результатів

виникає саме при спробі нормалізації слів. При цьому можуть виникнути 2 категорії помилок:

– нормалізація виконано некоректно, тому отримане слово не знаходиться у технічних текстах;

– слово віднесено до невірної частини мови, тому до нього були застосовані неверні правила нормалізації.

Обидва випадки є помилками, та з точки зору якості роботи системи потрібно їх уникати.

Верифікація нормалізації полягає у видачі одного з двох відповідей:

1) нормальна форма отримана правильно – нульова гіпотеза;

2) нормальна форма не відповідає початковому слову – альтернативна гіпотеза.

Метрики верифікації грунтуються на можливі помилки класифікації, які формалізовані для подібних завдань. З огляду на, що є 2 можливих відповіді алгоритму і 2 варіанти для об'єктивної ситуації, всього можливо 4 результати: «True Positive»–рішення вірно прийнято,«True Negative»–рішення вірно відкинуто, «False Positive» – відкинути гіпотезу H0, в той час, як вона є вірною(помилка 1 роду), «FalseNegative» – прийняття альтернативної гіпотези, коли насправді вірна нульова (помилка2 роду).

Якщо розглядати системупроведення частотного аналізу, то результат «False Negative» (пропуск цілі) передбачаєроботу з некоректним словом, для якого частотний аналіз не був передбачений. Результат «False Positive» (помилкова тривога,помилкове спрацьовування) означає, що наша система не змогла правильно виконати нормалізацію та відмовила у виконанні частотного аналіза. Ці дві помилки не можна вважати рівними за ступенем важливості, при вирішенні різних завданьпотрібно враховувати можливі ризики і втрати та визначати їх «вартість». Результат «False Positive» означає, що потрібно повторно отримати початкове слово та виконати нормалізацію.

Значення TP і FP можуть бути використані для визначення точності роботи алгоритму нормалізації.

Точність обчислюється як

$$Precision = TP / (TP + FP), \quad (2.14)$$

тобто показує наскільки алгоритмє коректним при запобіганні помилковості FP-спрацьовувань.

2.5 Висновки до розділу

У другому розділі розглянуто принципи роботи частотного аналізатора та створено регулярний вираз для розпізнавання англійських слів різної структури.

Побудовано модель нормалізації слів. Розглянуто особливості нормалізації для дієслів, іменників, прислівників та прикметників.

Проаналізовано можливості бази словників WordNetта показано можливість використання їх при розробці аналізатору.

Визначено спосіб верифікації результатів роботи частотного аналізатора.

3 СПЕЦИФІКАЦІЯ ВИМОГ ДО ЧАСТОТНОГО АНАЛІЗАТОРА

3.1 Варіанти використання системи

З частотним аналізатором можуть працювати дві категорії осіб –Administrator та User. Вони обидва успадковуються від абстракції Actor. Розмежування прав доступу до частотного аналізатора і послідовність роботи з ним спрощено виглядає наступним чином:

– Administrator – авторизується та завантажує словникі. При виборі словників обирає певну предметну область та завантажує словники саме з тією термінологією, що є релевантною обраній предметній області. Також може завантажувати загальні словники, що містять загальновживані слова;

– User – основний користувач системи. Саме вын ставить перед собою завдання визначення найбільш вживаних слів з текстових файлів для того, щоб оптимізувати їх вивчення, або створювати семантичне ядро сайту за певною тематикою, та ін.

Формальний аналіз предметної області та вимог до роботи програмного частотного аналізатора дозволяє виділити акторів: «Абстрактний користувач» (Actor), від якого успадковуються «Administrator» та «User».

Традиційно для описуфункціональних вимог використовується діаграма варіантів використань Unified Modeling Language, вона наведена у кваліфікаційній роботі на рис. 3.1. До діаграми варіантів використань UseCase додано опис сценаріїв використання.

Варіант використання № 1 – «Реєстрація»

Основний виконавець: Абстрактний користувач «Actor».

Передумови: Абстрактний користувач «Actor»бажає визначатись як «Administrator», для цього він повинен спочатку зареєструватись для можливості подальшої авторизації для власної ідентифікації.

Післяумови: «Administrator»або «User» зареєстрований у програмі частотного аналізатора.

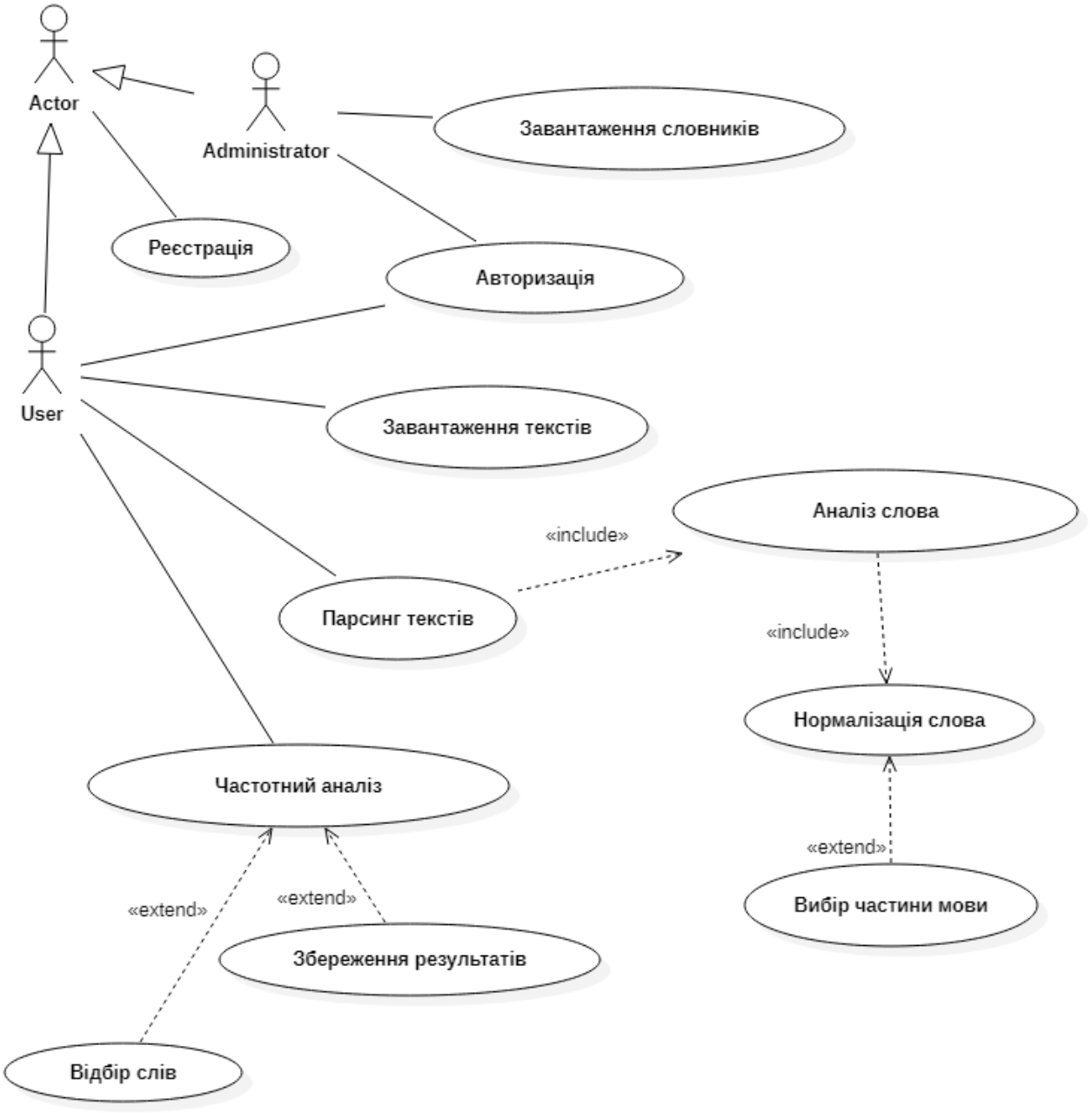

Рисунок 3.1 – Діаграма варіантів використання частотного аналізатора

Основний успішний сценарій.

1. Абстрактний користувач «Actor» обирає роль та натискає кнопку «Реєстрація».
2. Система надає користувачеві реєстраційну форму, у яку потрібно для реєстрації ввести логін, власний пароль та номер телефону.
3. Абстрактний користувач «Actor» вводить дані, які від нього вимагає програмний аналізатор.

4. Аналізатор перевіряє чи є введені дані припустимими за форматом. У якості логіну має бути будь-який рядок, який не співпадає з логіном іншогокористувача (якщо користувачів декілька). Логіном повинен бути рядок, що включає букви латинського алфавіту, з різних регістрів, та будь-який символ, що не є ані цифрою, ані буквою. Номер телефону вводиться у форматі +38(код оператора)(номер телефону).
5. Програмний аналізатор створює хеші для введених реєстраційниї даних користувача.
6. Програмний аналізатор зберігає введені реєстраційні дані.
7. Завершення прецеденту.

Альтернативні сценарії.

3а) Абстрактний користувач «Actor»відмовляється вводити реєстраційні дані.

1) Завершення прецеденту без додавання у систему нових адміністраторів.

4а) Аналізатор визначає помилку при введенні логіну (введене значення співпадає з логіном іншого адміністратора).

1) Система надає повідомлення про дублювання інформації.
2) Завершення прецеденту без збереження змін.

4б) Аналізатор визначає помилку формату при введенні паролю.

1) Система надає довідкові дані про вимоги до паролів.
2) Завершення прецеденту без збереження змін.

4в) Аналізатор визначає помилку формату при введенні номеру телефону.

1) Система надає довідкові дані про вимоги до формату номеру телефону.
2) Завершення прецеденту без збереження змін.

6а) Помилка у збереженні даних.

1) Повідомлення по помилку збереження. Завершення преценту.

Варіант використання № 2 – «Авторизація»

Основний виконавець: Абстрактний користувач «Actor».

Передумови: Абстрактний користувач «Actor» має реєстраційний профіль у системі.

Післяумови: Абстрактний користувач «Actor» увійшов у систему як «Administrator» чи «User».

Основний успішний сценарій.

1. Абстрактний користувач «Actor» заходить в розділ авторизації та обирає роль.
2. Частотний аналізатор запитує його логін, власний пароль та номер телефону. Якщо абстрактний користувач «Actor» є звичайним користувачем, авторизуватись не потрібно, перехід до завершення прецеденту.
3. Абстрактний користувач «Actor» заповнює дані щодо логіну, паролю та номера мобільного телефону у відповідних полях екранної форми програми.
4. Система зчитує введені дані, створює хеш для введених користувачем даних, шукає ці дані у файлі даних та порівнює їх між собою.
5. Система надає права доступудля користувача «Administrator» або «User». Авторизація є успішною.
6. Успішний вхід в частотний аналізатор та отримання доступу до функціоналу системи.
7. Завершення авторизації.

Альтернативний сценарій – «Авторизація».

3а) Не всі поля для авторизації заповнені користувачем.

1) Частотний аналізатор видає повідомлення про необхідність заповнити всі вказані поля.

3б) Абстрактний користувач «Actor» відмовляється вводити логін,власний пароль та номер телефону.

1) Авторизація не виконана. Прецедент авторизації завершився невдачно.

5а) Частотний аналізатор не може найти обліковий запис, що відповідає введеним авторизаційним даним.

1) Програмавидає повідомлення про помилку авторизації, користувачеві надається можливість повторного вводу даних.

3. Варіант використання «Завантаження словників».

Основний виконавець: Користувач системи «Administrator».

Передумова: користувач авторизований та відноситься до профілю «Administrator», бажає завантажувати словники.

Післяумови: Словники завантажено та додано у систему.

Основний успішний сценарій.

1. Адміністратор натикає кнопку «Завантажити словники».
2. Система надає можливість вибору предметної області.
3. Система надає користувачеві перелік існуючих предметних областей, для яких вже було додано словники.
4. Адміністратор обирає предметну область.
5. Адміністратор обирає файл (чи файли) зі словниками, які він бажає додати до частотного аналізатора.
6. Система перевіряє формат файлу.Він повинен бути у текстовому форматі ASCII.
7. Програмний аналізатор додає словник до переліку файлів, які він використовує.
8. Завершення прецеденту.

Альтернативний сценарій – «Завантаження словників».

2а) Немає зв’язку з базою даних словників.

1) Система надає повідомлення про помилку та пропонує спробувати з’єднання ще раз.

4а) Адміністратор відмовляється обирати предметну область.

1) Система видає повідомлення про те, що для продовження роботи потрібно обрати предметну область.

5а) Адміністратор відмовляється обирати словник.

1) Програмний аналізатор видає повідомлення про необхідність обрати англійський словник.

6а) Формат словника є невідповідним вимогам.

1) Система видає повідомлення про помилку формату словника.

7а) Помилки зв’язку системи з базою даних, неможливо приєднати новий словник.

1) Система видає повідомлення про неможливість додавання нового словника.

4. Варіант використання «Завантаження текстів».

Основний виконавець: Користувач системи «User».

Передумова: користувач бажає завантажити один чи декілька текстів для проведення у них частотного аналізу.

Післяумови: Необхідні тексти додані у систему.

Основний успішний сценарій.

1. Користувач обирає пункт меню «Додати новий текст».
2. Система надає можливість завантажити текст з диску або визначити шлях до веб-сторінки з текстом.
3. Користувач обирає спосіб додавання тексту.
4. Якщо обрано «Завантажити з диску», система надає користувачеві дерево каталогів для можливості переходу на потрібну папку та пошуку певного файлу.
5. Якщо обрано «Завантажити веб-сторінку», система надає форму для вводу шляху до сторінки.
6. Користувач задає дані для вибору тексту.
7. Система завантажує обраний користувачем текст.

8. Якщо потрібно завантажити ще один текст, перехід до п. 3.
9. Завершення прецеденту.

Альтернативний сценарій – «Завантаження текстів».

6а) Користувач відмовляється обирати текст.

1) Завершення прецеденту.

6б) Немає інтернет-зв’язку.

1) Система видає повідомлення про відсутність зв’язку.
2) Завершення прецеденту.

5. Варіант використання «Парсинг текстів».

Основний виконавець: Користувач системи «User».

Передумова: користувач завантажив тексти, система повинна виконати їх парсинг.

Післяумови: Тексти розбиті на набори окремих слів.

Основний успішний сценарій.

1. Користувач обирає пункт меню «Парсинг текстів».
2. Система звертається до файлів даних та застосовує правила регулярних виразів для парсингу.
3. Система отримує набір окремих слів.
4. Для кожного слова система застосовує варіант використання «Вибір слова».
5. Завершення прецеденту.

Альтернативний сценарій – «Парсинг текстів».

2а) Помилка у зверненні до файлів даних.

1) Частотний аналізатор видає помилку про неможливість роботи з файламитекстів.
2) Завершення прецеденту.

6. Варіант використання «Аналіз слова».

Основний виконавець: Користувач системи «User».

Передумова: система виконала парсинг текстів.

Післяумови: користувач бажає отримати слова у нормальній формі.

Основний успішний сценарій.

1. Система обирає чергове слово з тексту.
2. Система виконує пошук слова у словниках та визначає частину мови, до якої належить слово.
3. Система застосовує варіант використання «Нормалізація слова».
4. Завершення прецеденту.

Альтернативний сценарій – «Аналіз слова».

2а) Немає доступу до словників.

1) Система надає повідомлення про недоступність словників. Завершення прецеденту.

7. Варіант використання «Нормалізація слова».

Основний виконавець: Користувач системи «User».

Передумова: система виконала парсинг текстів та визначення частин мови для слів.

Післяумови: слова повинні бути представлені у нормалізованій формі.

Основний успішний сценарій.

1. Система виконує пошук слова у основному словнику відповідно до частини мови. В залежності від отриманого результату система визначає правила нормалізації.
2. Якщо у основному словнику слово не знайдено, система виконує пошук слова у словнику виключень. В залежності від отриманого результату система визначає нормалізовану форму для слова.
3. Система надає користувачеві отриманий список нормалізованих слів. При необхідності користувач застосовує для певного слова варіант використання «Вибір частини мови».
4. Система зберігає нормалізовані слова.

5. Завершення прецеденту.

Альтернативний сценарій – «Нормалізація слова».

1а) Немає доступу до основного словника.

1) Завершення прецеденту без оновлення даних.

2а) Немає доступу до словника виключень.

1) Завершення прецеденту без оновлення даних.

3а) Помилково визначена частина мови.

1) Користувач вручну вказує частину мови для певного слова.
2) Система оновлює інформацію щодо слова.

5а) Неможливо зберегти дані.

1) Повідомлення про помилку, завершення прецеденту без збереження даних.

Варіант використання 8 – «Вибір частини мови»

Основний виконавець: Користувач системи «User».

Передумови: Користувач бажає налаштувати частину мови для певного слова.

Післяумови: Частина мови налаштована.

Основний успішний сценарій.

1. Користувач обирає певне слово та натискає кнопку «Визначити частину мови».

2. Система надає випадаючий список з можливими варіантами частин англійської мови.

3. Користувач обирає потрібну та натискає кнопку «Застосувати».

4. Система зберігає оновлену інформацію.

5. Завершення прецеденту

Альтернативний сценарій – «Вибір частини мови».

3а) Користувач відмовляється обирати частину мови.

1) Дані не оновлюються. Завершення прецеденту.

4а) Неможливо зберегти дані.

1) Повідомлення про помилку, завершення прецеденту без збереження даних.

Варіант використання 9 – «Частотний аналіз».

Основний виконавець: Користувач системи «User».

Передумови: Користувач бажає провести частотний аналіз текстів.

Післяумови: Отримано пари вигляду W = <name, count>, впорядковані за спаданням значення count.

Основний успішний сценарій.

1. Користувач натискає кнопку «Частотний аналіз».
2. Система підраховує частоту слів.
3. Система виконує сортування списку слів за спаданням частоти.
4. Система надає користувачеві отриманий список.
5. Для збереження результатів за вимогою користувача виконується варіант використання «Збереження результатів».
6. Для відбіру слів (наприклад, для подальшого вивчення) за вимогою користувача виконується варіант використання «Відбір слів».
7. Завершення прецеденту.

Альтернативний сценарій 9 – «Частотний аналіз»

1а) Користувач відмовляється виконувати частотний аналіз.

1) Завершення прецеденту.

Варіант використання 10 – «Відбір слів».

Основний виконавець: Користувач системи «User».

Передумови: Користувач бажає за результатами частотного аналізу відібрати собі певні слова.

Післяумови: Набір слів сформований та збережений.

Основний успішний сценарій.

1. Система надає користувачеві результати частотного аналізу.

2. При необхідності додати слово у власник список користувач обирає це слово та натискає кнопку «Додати».

3. Якщо потрібно продовжити додавання, перехід до п. 2.

4. Для збереження списку слів користувач натискає кнопку «Зберегти список».

5. Система надає можливість обрати шлях та ім’я текстового файлу, у якому буде збережений список.

6. Користувач обирає шлях та ім’я файлу та натискає кнопку Ок.

7. Система зберігає список.

8. Завершення прецеденту.

Альтернативний сценарій 10 – «Відбір слів»

Альтернативні сценарії.

2а) Користувач відмовляється додавати слова.

1) Завершення прецеденту.

4а) Користувач відмовляється зберігати обрані слова.

1) Завершення прецеденту.

6а) Користувач відмовляється обирати шлях та/чи файл.

1) Завершення прецеденту.

7а) Неможливо зберегти дані.

1) Повідомлення про помилку, завершення прецеденту без збереження даних.

Варіант використання 11 – «Збереження результатів».

Основний виконавець: Користувач системи «User».

Передумови: Користувачмаєрезультати частотного аналізу та бажає їх зберегти.

Післяумови: Результати частотного аналізу збережено.

Основний успішний сценарій.

1. Система надає користувачеві результати частотного аналізу.

2. При необхідності зберегти результати користувач натискає кнопку «Зберегти».

3. Система надає можливість обрати шлях та ім’я текстового файлу, у якому будуть збережені пари W = <name, count>, впорядковані за спаданням значення count.

4. Користувач обирає шлях та ім’я файлу та натискає кнопку Ок.

5. Система зберігає список.

6. Завершення прецеденту.

Альтернативний сценарій 11 – «Збереження результатів»

Альтернативні сценарії.

2а) Користувач відмовляється зберегти пари результатів.

1) Завершення прецеденту.

4а) Користувач відмовляється обирати шлях та/чи файл.

1) Завершення прецеденту.

5а) Неможливо зберегти дані.

1) Повідомлення про помилку, завершення прецеденту без збереження даних.

3.2 Вимоги до нефункціональних характеристик

Нефункціональні вимоги призначено для визначення чи обмеження певних аспектів чи елементів, тобто для формалізації обмежень, які звужують вибір оптимальних рішень для апаратно-програмної реалізації. Користувачі, тобто групи осіб, що зацікавлені в проектованій системі, створюють твердження про власні потреби до цієї системи. Вони стосуються різних аспектів системи: архітектурних, поведінкових, та ін.

Виділимо наступні характеристики якості: функціональність, ефективність, переносимість, безпека. Визначення властивостей до цих характеристик можливо як в якісному, так і в кількісному вигляді. Забезпечення усіх нефункціональних

характеристик є обов’язковою частиною вимог до програмного частотного аналізатору.

Наведемо перелік та опис основних характеристик якості для розроблюваної програмної системи.

Функціональність

«Здатність до взаємодії»: Частотний аналізатор повинен взаємодіяти з базою даних, словниками та файлами збережених користувачами даних для отримання інформації та її використання.

«Захищеність»: дані адміністратора та користувачів (логін та пароль) повинні хешуватись.

Переносимість

«Адаптованість»: Програма підтримує кросплатформовість, її можна буде запускати на різних платформах.

Ефективність

«Використання ресурсів»: Програма не повинна використати більше 512 МБ, у тому числі словники – не більше ніж 256 Мб.

Безпека

«Безпека програми»: Програма повинна розповсюджуватись лише з відома автора. Використання словників йде за ліцензією WordNet без права комерційного перепродажу.

«Безпека персональних даних»: Як для адміністратора, так і для користувача, персональні дані повинні захищатись від несанкціонованого доступу. Реєстраційні дані можуть бути відновленими за номером телефону.

«Безпека результатів»: Отримані результати частотного аналізу повинні бути лоступними тому користувачеві, який саме й проводив частотний аналіз. Списки слів, відібраних користувачами для подальшого власного користування, повинні бути доступними тільки самому користувачеві.

Сценарії атрибутів якості

«Часові характеристики»:

– виконання частотного аналізу повинно виконуватися менше, ніж за 3 секунди;

– ідентифікація користувача за вказаними логіком-паролем повинна виконуватися менше, ніж за 1,5 секунди;

– підключення нового словника повинно виконуватися менше, ніж за 2 секунди;

– збереження списку обраних користувачами слів повинно проходити менше, ніж за 1,5 секунд.

«Надійність»

1) Якщо користувач бажає зберегти список певних слів, то система виконує це успішно не менш ніж у 95% випадків.

2) Якщо адміністраторпідключає новий словник, то програмний аналізаторможе ним користуватись не менше ніж у 90% випадків.

3) У випадку критичної помилки при роботі зі словниками аналізатор зберігає роботоспроможність у 85% випадків.

4) Якщо програмний аналізатор зберігає дані щодо введеного рукописного підпису, то ймовірність вдалого збереження не менше 0.9.

5) Якщо система зберігає дані щодо результатів фільтрації введеного рукописного підпису, то ймовірність вдалого збереження не менше 0.9.

«Продуктивність»

1) Якщо адміністратор чи користувач програмного аналізатора бажає пройти авторизацію, то він отримує повідомлення о результаті менш ніж за 1,5 секунди після завершення вводу даних.

2) Якщо користувачвиконує частотний аналіз, то система надає результат менше ніж за 3 секунди.

3) Якщо користувач реєструється у системі, то система оброблює його персональні дані менш ніж за 1,5 секунди.

4) Якщо користувачзберігає список певних слів, то система виконує це менш ніж за 2 секунди після підтвердження операції збереження.

5) Якщо адміністратор запитує перелік можливих для підключення словників, то програмний аналізатор показує повідомлення о результаті менш ніж за 2,5 секунди.

6) Якщо користувач завантажує тексти для проведення над ними аналізу, то системі на це знадобитьсяменш ніж за 3 секунди.

«Безпека»

1) Якщо система запитує інформацію щододаних користувача, то отримує її в цілісності не менш, ніж у 90% випадків.

2) Якщо користувач створює власний список слів, то можливість «злому» його іншою особою менше ніж у 5% випадків.

«Зручність користування»

1) Якщо користувачстворює власний список певних слів, то для відкриття програмної форми для цього йому потрібно не більше ніж 2,5 секунди.

2) Якщо користувач бажає переглянути результати частотного аналізу, то для цього для відкриття програмної форми йому потрібно не більше ніж 2 секунди.

«Супроводження»

1) Якщо потрібно змінити функціонал частотного аналізатора, то зміни вступлять у дію менш ніж за 2 місяці.

2) Якщо потрібно відмовитись від використання словників компанії WordNet, то для створення та розробки механізмів використання нових словників буде потрібно не більш, ніж 6 місяців.

3) Якщо потрібно портувати систему на іншу платформу (наприклад, мобільну), то портування буде виконано не більш, ніж за 5 місяців.

4) Якщо відмічено велику кількість відгуків про помилкову роботу аналізатора, то виправлення вступлять у силу не більш, чим через 3тижні.

5) Якщо потрібно збереження списків слів користувачів у іншому форматі, то для цього потрібно не більш ніж 2 тижні.

3.3 Системні вимоги

Системні вимоги для роботи аналізатора наведені у табл. 3.1.

Таблиця 3.1 – Інформація щодо системних вимог

Вимога	Мінімальне значення	Рекомендоване значення
Платформа	Debian	Windows, Ubuntu, MacOS
Контролер	Клавіатура, миша	Клавіатура, миша
Оперативна пам'ять	512 Мб	2 Гб і більше
Вільний простір на жорсткому диску	64 Мб	512 Мб і більше
Процесор	Intel Core I3	Core i5 і більше
Розрядність процесора	32	64
Відеоадаптер	Intel	Nvidia, AMD

3.4 Висновки до розділу

У третьому розділі виконано специфікацію вимог до частотного аналізатора. Для цього спочатку визначено функціональні вимоги з визначенням варіантів використання програмного аналізатора. Потім формалізовано нефункціональні вимоги, розроблено сценарії атрибутів якості. Наприкінці створено системні вимоги з визначенням мінімальних та рекомендованих значень.

4 ПРОЕКТУВАННЯ ЧАСТОТНОГО АНАЛІЗАТОРУ ТЕКСТІВ

4.1 Проектування архітектури системи

З точки зору архітектури розроблювана система не є складною. Система має дві категорії користувачів, для яких потрібні різні представлення даних та доступ до них, тому для розробки аналізатору буде доцільно використати архітектурний патерн Model-View-Controller(MVC). За його допомогою буде зручно відокремити взаємодію з користувачем та інтерфейс від внутрішньої реалізації системи.

Тому архітектура програми складається з трьох частин:

– Model – дані, які зберігаються у системі, та способи їх обробки;

– View – вигляди, які бачать користувачі різних категорій;

– Controller – механізм обробки повідомлень, щонадходять від користувачів різних категорій.

Узагальнена архітектура системи наведена на рис. 4.1.

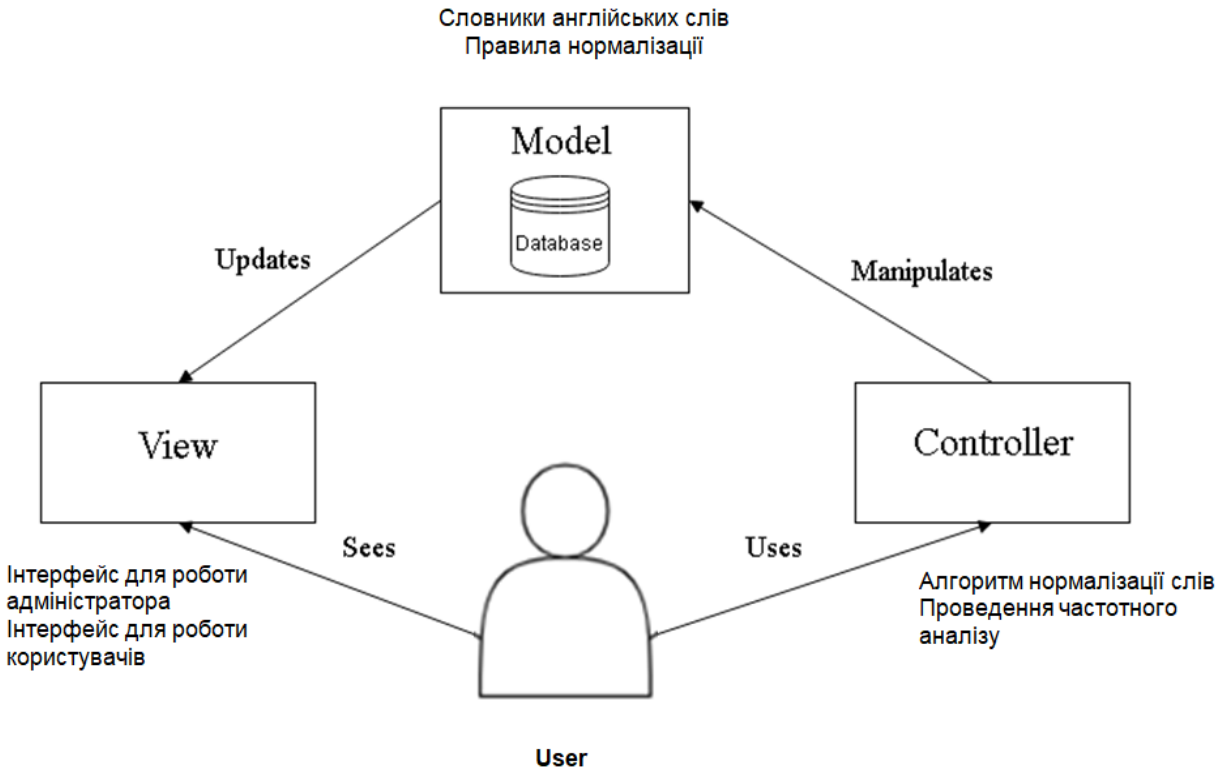

Рисунок 4.1 – Узагальнена архітектура програмного частотного аналізатору

4.2 Структури даних словників та спосіб їх використання

Словники зберігаються у вигляді csv-файлів. Для стандартної бази іменників, дієслів, прикметників та прислівників існує по 2 файли: базові слова, що змінюються за правилами (файлиindex.noun, index.verb, index.adj, index.adv) та винятки (файли noun.ext, verb.ext, adj.ext, adv.ext).

Файл index.*містить тільки леми (нормалізовані форми слів):

[лема]

…

[лема]

Впорядковувати слова не обов’язково, але це підвищує швикість обробки словника.

Файли*.ext містить дані у форматі:

[слово-виняток] [пробіл] [лема]

…

[слово-виняток] [пробіл] [лема]

Узагальнений алгоритм підключення словників наведений на рис. 4.2. Для того, щоб розуміти з якими словниками працювати для нормалізації виділеного слова, використовується APIWordNet, який містить механізми визначення частини мови для слова у будь-якій формі.

Після звернення APIWordNet повертає маркер частини мови, що визначає відповідні словникі, або ж маркер помилки, якщо частину мови визначити неможливо. Останній випадок свідчить про неможливість нормалізації наданого слова та відсутності відповідних словників.

За потребою можна підключати власні словники, що містять слова-терміни певної технічної області. Зазвичай такі слова змінюються відповідно правил, тому словникі з виключеннями не потрібні. Кількість словників, які може підключати користувач, є необмеженою. Порядок перегляду таких словників користувач може встановлювати самостійно.

Рисунок 4.2 – Узагальнений алгоритм підключення словників

За бажанням користувача можна встановлювати наступні порядки пошуку слів у словниках:

– файл *.ext, файл index.*, власні словники

або

– власні словники, файл *.ext, файл index.*

4.3 Детальне проектування частотного аналізатору

Для детального проектування частотного аналізатору розглянемо виконання основнихдій з плином часу, що дозволяють продемонструвати діаграми діяльності та взаємодії.

Рисунок 4.3 показує роботу для прецеденту «Реєстрація». Незареєстрований користувач (Actor) повинен зареєструватись для отримання доступу до функціоналу системи. При цьому для можливості управління словниками він повинен зареєструватися як Адміністратор. При реєстрації як звичайний користувач він отримує можливість за даними свого профіля зберігати набори слів для подальшого використання.

Користувач Actor надсилає власні логін та пароль, які контролер перевіряє на відповідність вимогам та зберігає у базі даних. При наявності у системі декількох адміністраторів пари логін/пароль повинні бути унікальними, що забезпечує виконання базових вимог до безпеки.

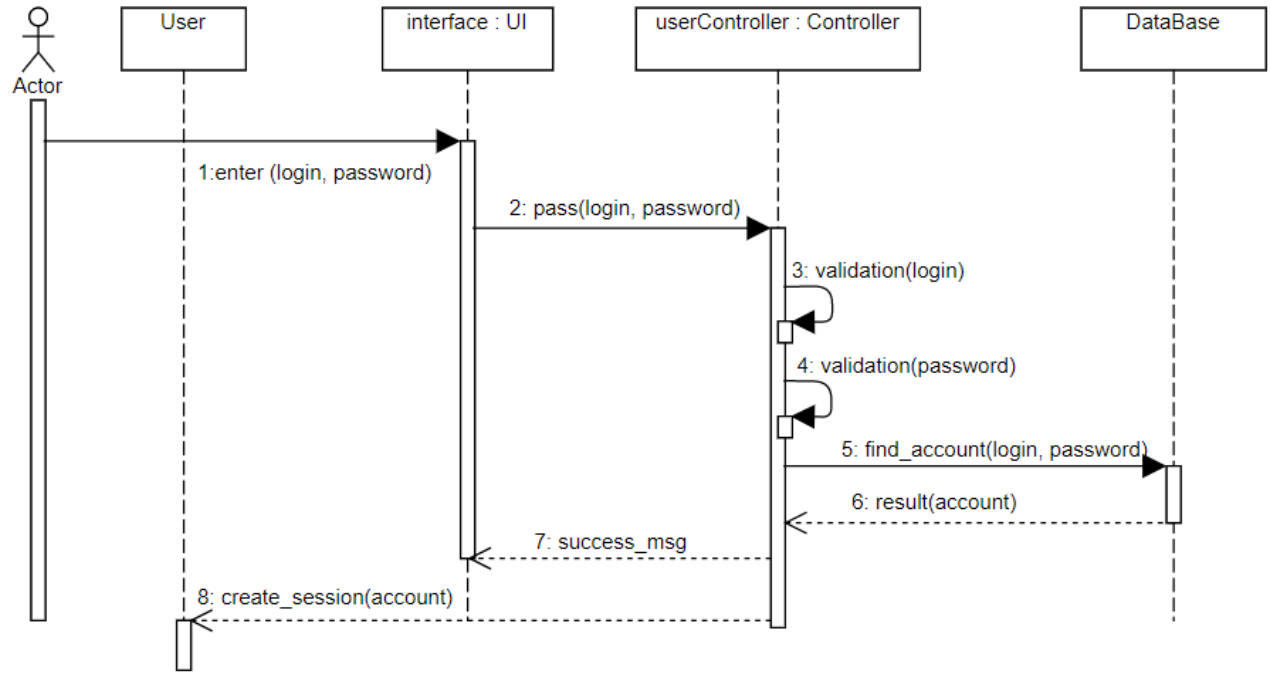

Рисунок 4.3 – Діаграма послідовностей прецеденту «Реєстрація»

На рис. 4.4 показано прецедент для авторизації Адміністратора. Йому потрібно ввести логін та пароль, які проходять валідацію та за якими здійснюється пошук у БД. Після успіху користувач Адміністратор отримує доступ до свого акаунту.

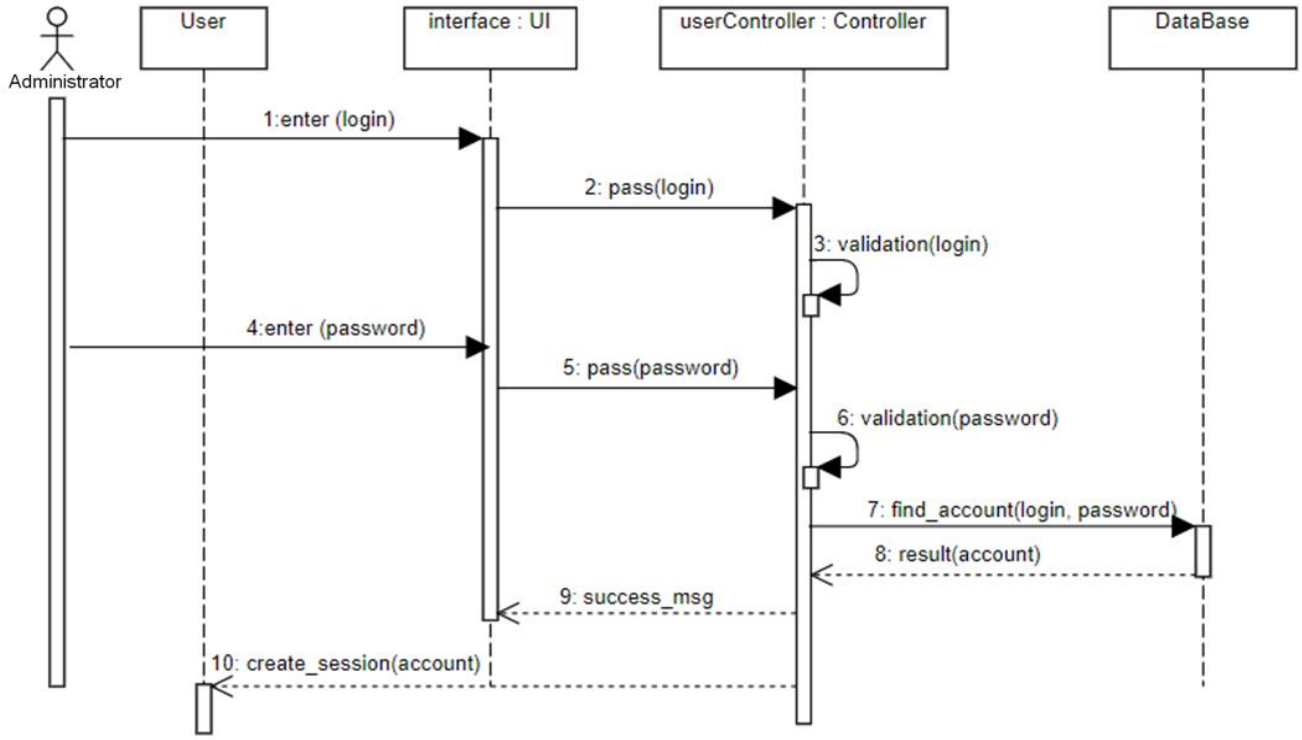

Рисунок 4.4 – Діаграма послідовностей прецеденту «Авторизація»

На рис. 4.5 наведена діаграма діяльності лематизатора. Спочатку виконується налаштування шляхів до словників, потім отримається слово та виконується перевірка чи є це слово складеним. Складене слово виглядає як два чи три простих слова, поєднаних знаком «тире». Якщо зустрічається складене слово, воно розбивається на прості слова, які шукаються у відповідних словниках та приводяться до нормальної форми, після чого нормалізовані прості слова знов поєднуються для утворення складеного слова.

На рис. 4.6 та 4.7 наведені діаграми діяльності, що демонструють процес нормалізації саме для дієслів та іменників.

Як було зазначено у другому розділі, існують правила нормалізації правильних дієслів, тому якщо дієслово не знайдено у словнику неправильних дієслів, до нього застосовуються певні правила нормалізації. Аналогічно відбувається для іменників, які можуть створювати множину з використанням звичайних правил або як виключення. Якщо іменник не знайдений у словнику виключень, до нього застосовуються правила нормалізації іменників.

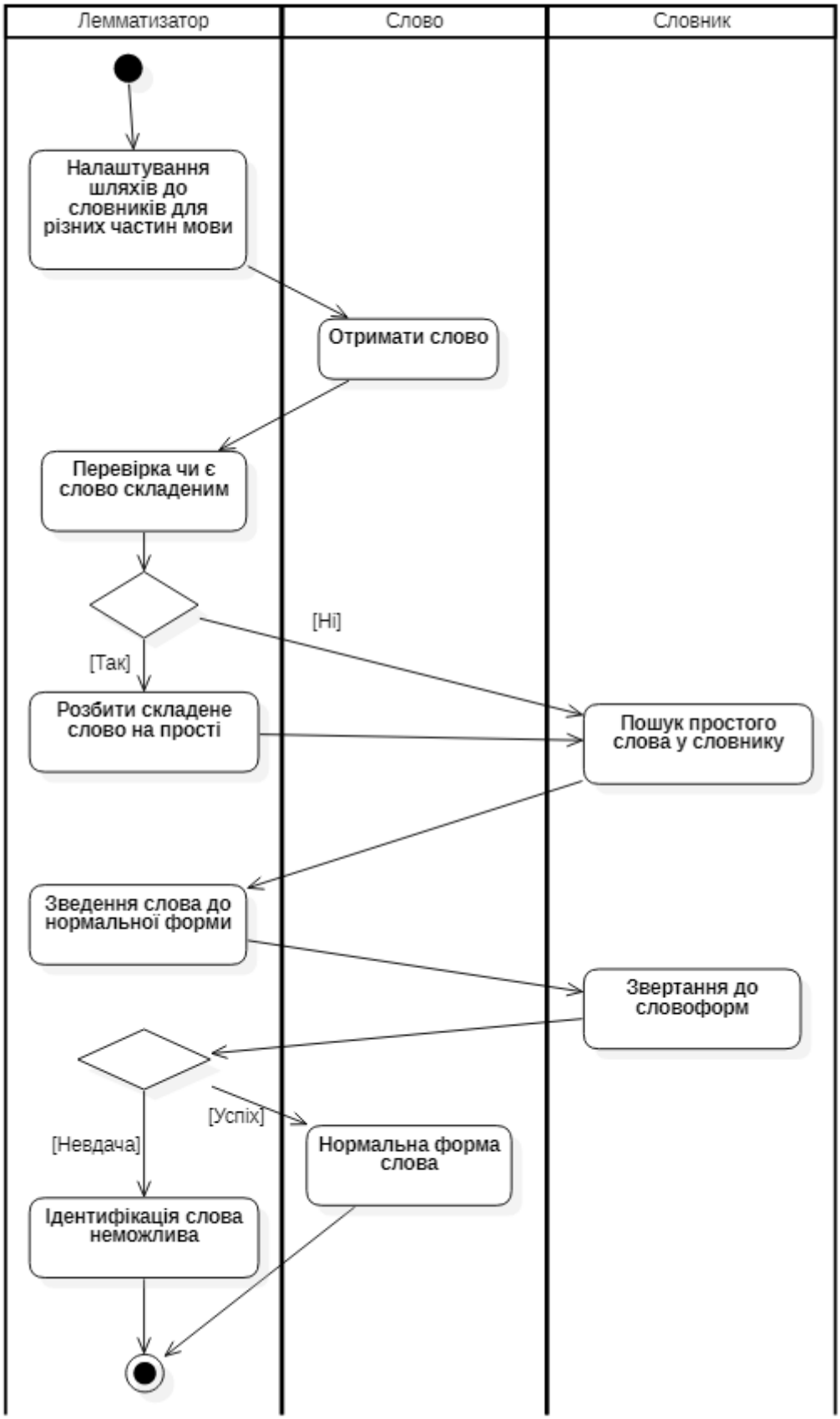

Рисунок 4.5 – Діаграма діяльності роботи лематизатора

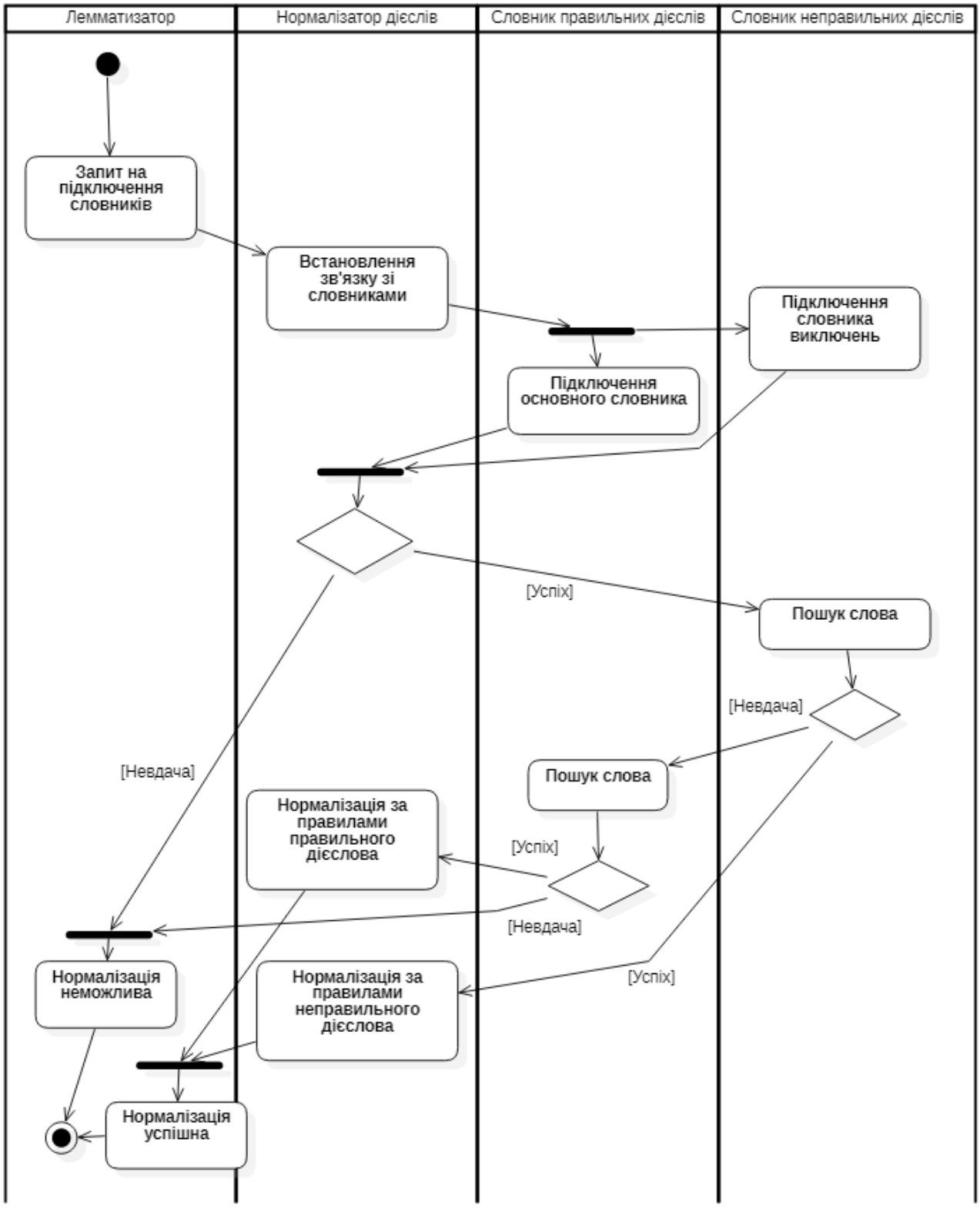

Рисунок 4.6 – Діаграма діяльності нормалізації дієслова

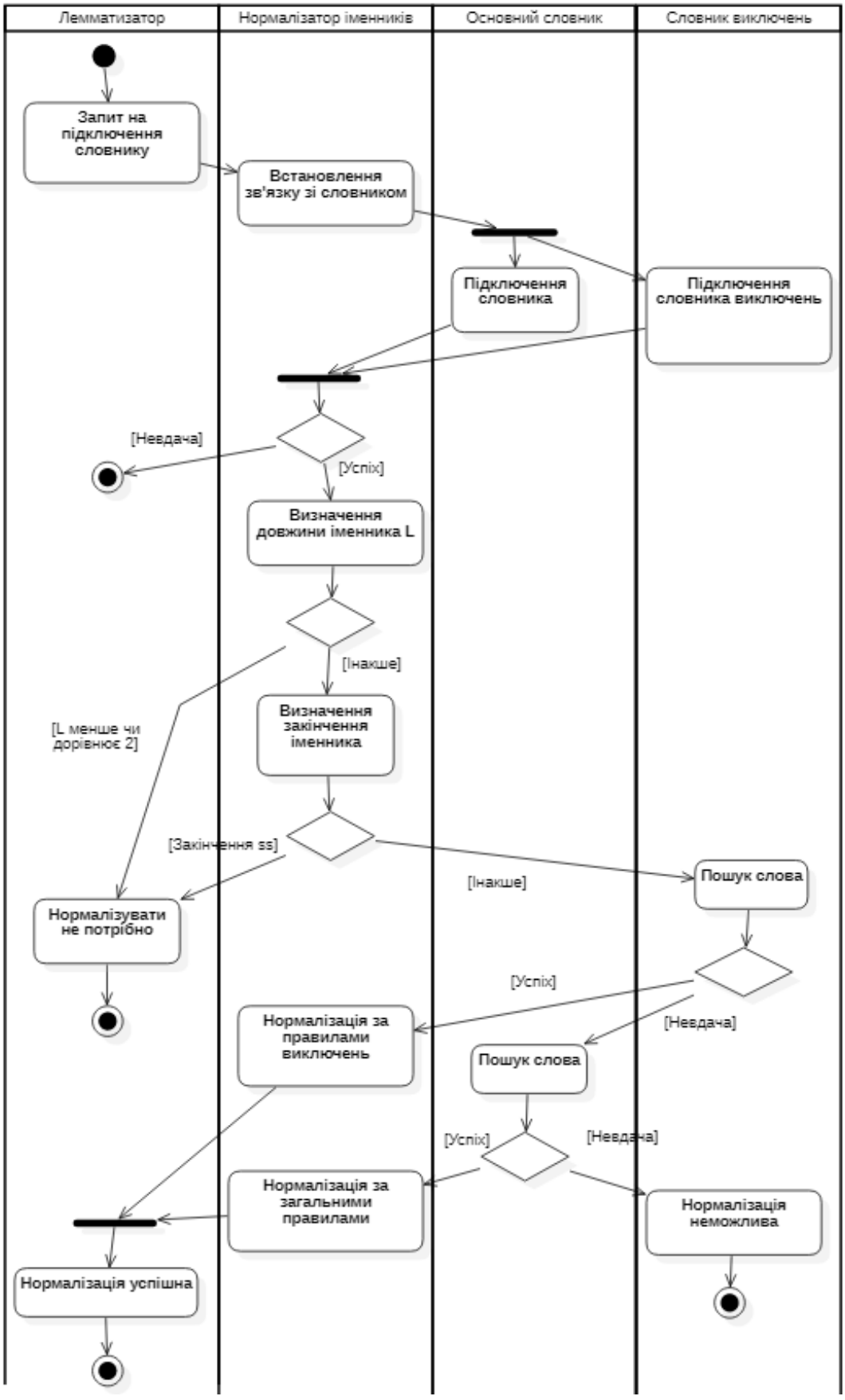

Рисунок 4.7 – Діаграма діяльності нормалізації іменника

4.4 Структура бази даних

Для виконання будови моделі БД тарозробки її структури потрібно визначити сутності, а також зв’язки між ними й атрибути сутностей. Після аналізу предметної області визначено такий набір сутностей, які використовуються для програмного проведення частотного аналізу технічних текстів, створених англійською мовою:

- SystemUsers – таблиця даних для усіх користувачів програмної системи незалежно від профілю;
- Role – таблиця з переліком можливих ролей (профілів);
- PartOfSpeech – таблиця частин мови, що є у англійських текстах;
- Vocabulary – мовні словники;
- Seance – таблиця сеансів використання частотного аналізатора;
- FrequencyAnalysis – таблиця, що містить пари «слово, кількість»;
- ListOfWords – таблицяз наборами слів, які користувач бажає зберегти для подальшого використання.

Тепер визначимо види зв’язків між виділеними сутностями та спроектуємо структури відповідних їм таблиць.

Користувач «SystemUsers» (табл. 4.1) містить дані по користувачам системи, в тому числі й реєстраційні дані. Зазначимо, що для приватних даних призначається процедура хешування, тобто приватні дані не зберігаються у базі даних у «відкритому» вигляді.

Таблиця 4.1 – БД: Реляційна таблиця SystemUsers

Поле	Тип даних	Опис	Ключ
IdUser	integer (8)	Ідентифікатор користувача системи	PK

Продовження таблиці 4.1

Поле	Тип даних	Опис	Ключ
IdRole	integer (3)	Ідентифікатор ролі (профілю) користувача системи	
Surname	varchar (50)	Прізвище користувача системи	
Name	varchar (50)	Ім’я користувача системи	
Login	varchar (30)	Хеш логіну користувача системи	
Password	varchar (50)	Хеш паролю користувача системи	

Таблиця «Role» є довідником припустимих ролей (профілів) користувачів системи. Зараз передбачено два типу профілів – Адміністратор та Користувач, але при необхідності у наступному цей набір може бути розширений. Структура «Role» наведена у табл. 4.2.

Таблиця 4.2 – БД: Реляційна таблиця Role

Поле	Тип даних	Опис	Ключ
IdRole	integer (8)	Ідентифікатор ролі	PK
RoleName	varchar (80)	Назва ролі	

Довідник «PartOfSpeech» містить набір усіх частин мови, що підлягають розпізнаванню та можуть у подальшому приймати участь у підрахунку кількості

входжень у технічні тексти. Зараз передбачено чотири типи частин мови – іменник, дієслово, прикметник, прийменник, але при необхідності у наступному цей набір може бути розширений. Структура та поля «PartOfSpeech» наведені у табл. 4.3. Таблиці «SystemUsers» та «Role» пов’язані відношенням один-до-багатьох.

Таблиця 4.3 – БД: Реляційна таблиця PartOfSpeech

Поле	Тип даних	Опис	Ключ
IdPartOfSpeech	integer (4)	Ідентифікатор частини мови	PK
NamePartOfSpeech	varchar (30)	Назва частини мови	

Таблиця «Vocabulary» є допоміжною для можливості використання мовних словників. Слова різних частин мови розташовані у різних словниках. Для кожної частини мови може бути застосовано один чи декілька словників. Тому таблиці «PartOfSpeech» та «Vocabulary» пов’язані відношенням один-до-багатьох. Структура та опис полів «Vocabulary» міститься у табл. 4.4.

Таблиця 4.4 – БД: Реляційна таблиця Vocabulary

Поле	Тип даних	Опис	Ключ
IdVocabulary	integer (8)	Ідентифікаторсловника	PK
NameVocabulary	varchar (30)	Назва словника	
IdPartOfSpeech	integer (4)	Ідентифікатор частини мови	

Продовження таблиці 4.4

Поле	Тип даних	Опис	Ключ
IsMain	boolean	Чи є словник основним (IsMain=true)або словником виключень (IsMain=false)	
Path	string (1000)	Повний шлях до словника (з його ім’ям)	

Сутність «Seance» (табл. 4.5) відповідає за інформацію по сеансам проведення частотного аналізу. Кожен користувач системиможе багато разів проводити частотний аналіз, тому таблиці «SystemUsers» та «Seance» пов’язані відношенням один-до-багатьох.

Таблиця 4.5 – БД: Реляційна таблиця Seance

Поле	Тип даних	Опис	Ключ
IdSeance	integer (8)	Ідентифікатор сеансу	PK
IdUser	integer (8)	Ідентифікатор користувача системи	
Date	date (dd-mm-yyyy)	Ідентифікатор роботодавця	
Time	time (hh-mm)	Ідентифікатор категорії вакансії	
IdWord	integer(8)	Ідентифікатор слова	

Результатом проведення частотного аналізу є набір пар «слово»-«кількість входжень у тексти». Такі пари зберігаються у таблиці «FrequencyAnalysis» (табл. 4.6). У результаті проведення частотного аналізу формується багато таких пар, тому таблиці «Seance» та «FrequencyAnalysis» мають відношення один-до-багатьох.

Таблиця 4.6 – БД: Реляційна таблиця FrequencyAnalysis

Поле	Тип даних	Опис	Ключ
IdWord	integer(8)	Ідентифікатор слова	PK
Word	varchar (255)	Назва категорії робочої посади	
Count	varchar (255)	Опис категорії робочої посади	Nullable

За результатами проведення частотного аналізу, які переглядаються користувачами,можуть бути сформовані списки з певних слів. Користувач сам за власним бажанням обирає такі слова з отриманого результату частотного аналізу. Ці слова можуть бути у подальшому використані ним для ефективного вивчення англійської мови саме для розуміння обробленого тексту (чи сукупності текстів), або ж для створення семантичного ядра сайту за тематиками, що містяться у текстах, та ін.

Слова зберігаються у текстових файлах для можливості їх подальшого перегляду чи інтеграції у будь-яку систему. Кожен користувач може мати декілька списків, тому між сутностями «SystemUsers» та «ListOfWords» існує відношення один-до-багатьох.

Структура таблиці «ListOfWords» наведена у табл. 4.7.

Таблиця 4.7 – БД: Реляційна таблиця ListOfWords

Поле	Тип даних	Опис	Ключ
IdList	integer(8)	Ідентифікатор списку	PK
IdUser	integer (8)	Ідентифікатор користувача системи	
Path	string (1000)	Шлях до папці, у якій збережуються списки слів	
Name	varchar (255)	Ім’я файлу	

ER-діаграма розроблюваного у роботі програмного частотного аналізатора наведена на рис. 4.8.

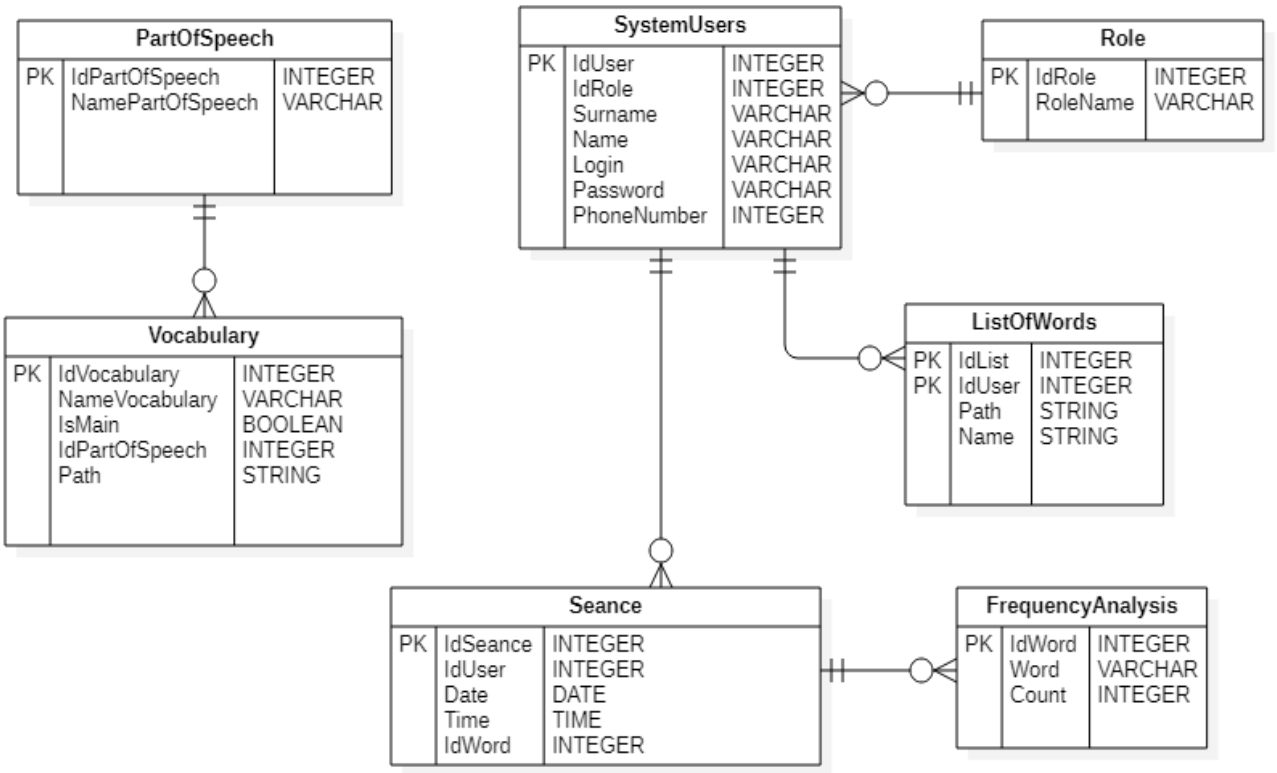

Рисунок 4.8 – Реляційна модель даних

Діаграма містить усі описані таблиці та їх взаємозв’язки.

4.5 Проектування структури програмних класів

На рис. 4.9 наведена діаграма програмних класів розроблюваного аналізатора.

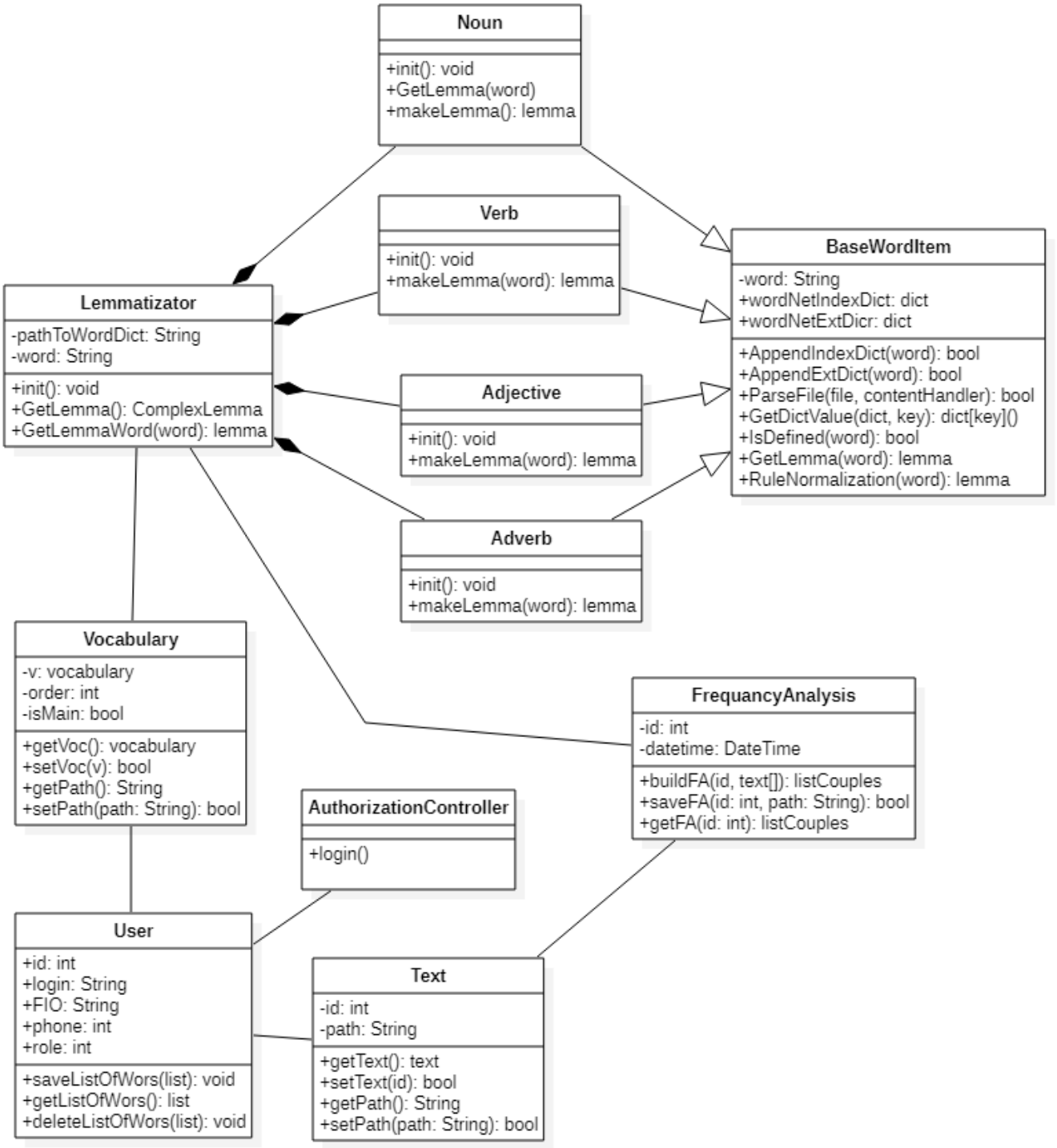

Рисунок 4.9 – Діаграма програмних класів

Розглянемо більш детально створені програмні класи.

Клас Lemmatizator застосовується для отримання леми слова.

Рядок pathToWordDict – шлях до словників,

рядок word – початкове слово.

Методи:

init() – початкова ініціалізація;

GetLemma(word) – повертає лемму слова (яке може бути складеним);

GetLemmaWord(word) – повертає нормалізовану форму слова.

З класом Lemmatizator відношенням композиції пов’язани чотири класи: Noun, Verb, Adjective та Adverb.

Клас Noun – клас для нормалізації слів, що є іменниками.

Методи:

init() – початкова ініціалізація класу Noun, створюються правила роботи з закінченнями іменників;

GetLemma(word) – перевизначений метод для нормалізації;

makeLemma() – отримання леми.

Клас Verb – клас для нормалізації слів, що є дієсловами.

Методи:

init() – початкова ініціалізація класу Verb, створюються правила роботи з закінченнями дієслів;

makeLemma() – отримання леми.

Аналогічною є структура класів Adjective та Adverb.

Клас BaseWordItem – головний клас для нормалізації слів.

word – поточне слово.wordNetIndexDict – базовий словник.

wordNetExtDicr – словник виключень.

Методи:

AppendIndexDict(word): bool – підключення словника з базовими словами;

AppendExtDict(word): bool – для розбору рядка з певного файлу використовуються нормалізовані форми виключень та їх ненормалізовані форми.

ParseFile(file, contentHandler): bool – розбір файлу рядок за рядком, викликання для кожного такого рядка обробник в залежності від частини мови;

GetDictValue(dict, key): dict[key] – пошук слова у словнику чи декількох словниках;

IsDefined(word): bool – перевірка того, що слово існує;

GetLemma(word): lemma – повертання леми (нормалізованого слова або виключення);

RuleNormalization(word): lemma – приведення слова за правилами (відповідно до частини мови) до нормалізованої форми.

Клас Vocabulary – для роботи зі словниками, в тому числі власними з певних предметних областей.

v: vocabulary – певний словник;

order: int – порядковий номер, за яким система шукає слово саме у цьому словнику;

isMain: bool – чи є він словником базових слів чи слів-виключень;

getVoc(): vocabulary – отримати словник;

setVoc(v): bool – встановити словник;

getPath(): String – отримати шлях до словника;

setPath(in path:String): bool – встановити шлях до словника.

Клас User – використовується для роботи з користувачами системи.

id: int – ідентифікатор користувача;login: String – логін користувача;

FIO: String – прізвище та ім’я користувача;phone: int – номер телефону;

role: int – роль (адміністратор чи звичайний користувач).

Методи:

saveListOfWors(list): void – збереження списку слів за вимогою користувача;

getListOfWors(): list – отримання списку слів, що був створений певним користувачем;

deleteListOfWors(list): void – видалення непотрібного списку слів.

Клас AuthorizationController – використовується для реєстрації та авторизації користувача системи.login() – головний метод.

Клас Text – використовується для работі з текстами, зміст яких буде основою для частотного аналізу.

id: int – ідентифікатор тексту;path: String – повний шлях до тексту.

Методи:

getText(): text – отримати файл з текстом, визначений користувачем;

setText(id): bool – завантажити у систему файл з текстом;

getPath(): String – отримати шлях до файлу з текстом;

setPath(path:String): bool – встановити чи змінити шлях до файлу з текстом.

Клас FrequancyAnalysis – використовується для проведення частотного аналізу.

id: int – номер сеансу обробки текстів;

datetime: DateTime – дата та час проведення сеансу обробки;

buildFA(id, text[]): listCouples – побудова пар «слово»-«кількість» для усіх текстів text[];

saveFA(id:int, path:String): bool – збереження результатів;

getFA(id:int): listCouples – отримати результати попередньо проведеного частотного аналізу.

4.6 Висновки до розділу

У четвертому розділі обрано архітектурний патерн MVC для проектування частотного аналізатору текстів. Визначено структури даних словників, спосіб та послідовність використання цих словників для роботи лематизатора. Створено діаграми послідовностей та діяльності для виконання основних функцій програми. Розроблено структуру бази даних та створено її реляційну модель. Наприкінці розроблено структуру програмних класів системи та описано класи та зв’язки між ними.

5 ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗРОБЛЮВАНОЇ СИСТЕМИ

5.1 Особливості створення програмних модулів з урахуванням мови програмування

При розробці програмноїсистеми для проведення частотного аналізу технічних текстів англійською мовою було використано сучасну мову програмування Python, безкоштовне інтегроване середовище PyCharm 2021.1.2 та систему контролю версій Git.

Для організації інтерфейсу використано бібліотеку TKinter. Вона дозволяє створювати багатовіконні застосування та перемикати управління між вікнами програми. Для кожного вікна можна встановити ширину і висоту за допомогою screen_width та screen_height. Для того, щоб ці значення розраховувались виходячи з розмірів екрану, використовуються методи winfo_screenwidth() та winfo_screenheight(). До кожного вікна можуть бути прив’язані кнопки (метод Button()), радіо кнопки (метод Radiobutton()), зображення (методImageTk.PhotoImage()). Для створення меню використовується метод Menu(). Команди додаються до пунктів меню за допомогою add_command(label="Назва команди"). Якщо потрібні каскадні меню, викликається метод add_cascade(label=" Назва команди", menu=”Назва пункту меню”).

Для зберігання даних використовується база даних MySQL [13]. Для того, щоб зв’язати MySQL та Python, потрібно встановити модуль Python SQL з використанням pip:

pip install mysql-connector-python

Для роботи з сервером MySQL потрібно імпортувати такі модулі:

import mysql.connector

from mysql.connector import Error

Для підключення до серверу MySQL необхідно задати ім’я хосту, користувача та пароль користувача. Після цього повертається об’єкт connection.

Для створення бази даних потрібно передати об’єкт connection та запит на створення query.

Для можливості нормалізації слів використовуються словники базиWordNet, у яких слова розмічені відповідно до частин мови. Для кожної частини мови виділено по два файла: index.pos та data.pos, де pos - це noun, verb, adj та adv.Словники представлені у текстовому форматі ASCII.

Для перекладу збережених користувачем слів та, фактично, формування власного словника англійських слів та їх перекладів російською використовано електронний словник StarDict [14]. Він має відкритий код та може використовувати пошук у багатьох необхідних онлайн-словниках.

Для роботи з системою користувач повинен реєструватись та авторизуватись як адміністратор чи користувач. При цьому персональні дані адміністраторів та користувачів повинні бути захищеними. Для цього виконується хешування з використанням алгоритму sha1. В Python хеш-функція приймає вступну послідовність зі змінною довжиною в байтах і конвертує її в послідовність з фіксованою довжиною. Ця функція потребує підключення бібліотеки hashlib: import hashlib. Для хешування рядку str1 виклик функції:

result = hashlib.sha1(str1.encode('ascii')).hexdigest()

Ця функція є односторонньою, тобто з нею можна можна отримати хеш по паролю, але, маючи сам хеш, не можна відтворити пароль. Тому введені користувачем дані хешуються та отримані хеші порівнюються з відповідними користувачам.

5.2 Реалізація інтерфейсу користувачів системи

Наведемо приклади реалізації інтерфейсу програми для проведення частотного аналізу слів в англійських текстах та покажемо додаткові можливості програми. На рис. 5.1 показано загальний вигляд головного вікна програми. Для вибору режимів адміністратора та звичайного користувача потрібно обрати відповідний перемикач та натиснути кнопку Start.

Рисунок 5.1 – Загальний вигляд програми

Після цього відкривається форма для реєстрації чи авторизації. На рис. 5.2 показан приклад форми для авторизації адміністратора.

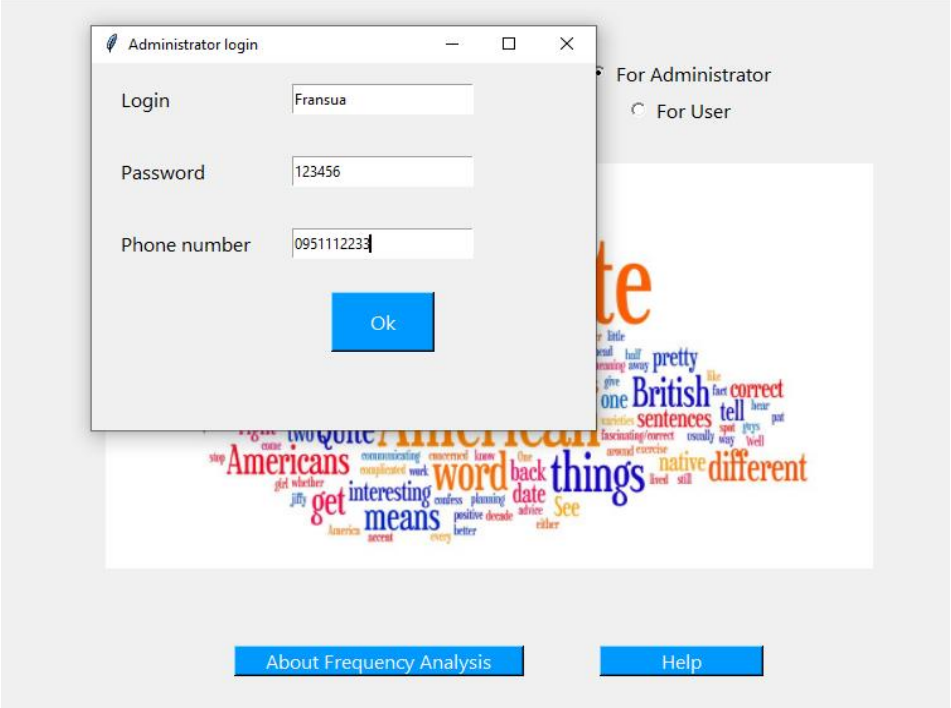

Рисунок 5.2 – Приклад форми авторизації адміністратора

На рис. 5.3 наведено вікно для реєстрації та авторизації користувача. Ім’я та прізвище користувача задаються тільки на етапі реєстрації, для авторизації потрібні лише логін та пароль.

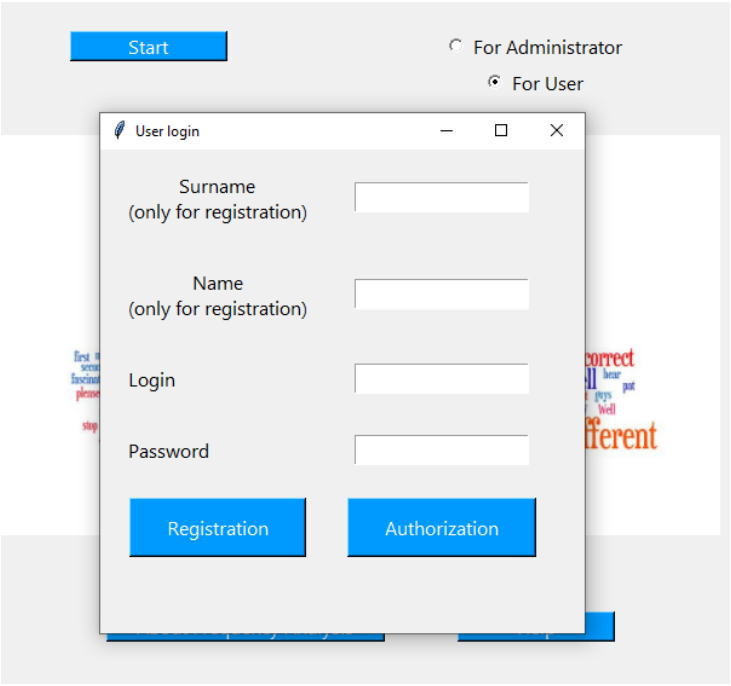

Рисунок 5.3 – Вікно для реєстрації та авторизації користувача

При роботі у режимі адміністратора є можливість додавання та видалення основних та спеціальних словників (рис. 5.4). Основні словники містять слова різних частин мови усередені певного словника. Спеціальні словники містять тільки іменники з заданої предметної області: економіка, хімія, фізика, легка промисловість, тощо.

Для додавання нового словника потрібно натиснути кнопку «Load the dictionary» та обрати потрібний словник. Зазвичай словники містяться у папці Vocabulary (рис. 5.5).

Для видалення словника потрібно обрати його та натиснути кнопку «Delete the dictionary».

На рис. 5.6 можна побачити виконання основного функціоналу програми – обчислення частот слів.

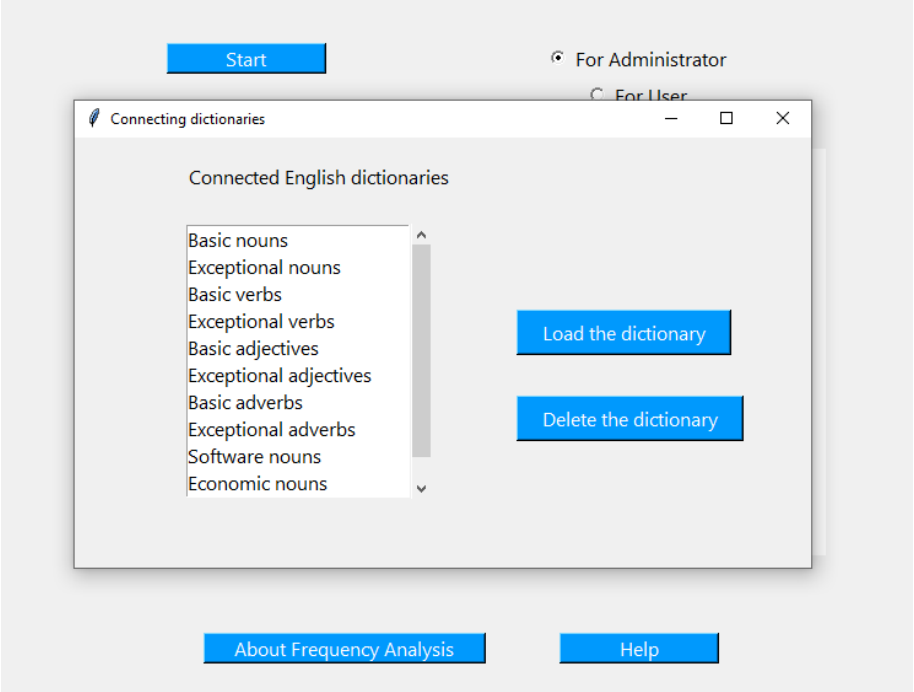

Рисунок 5.4 – Додавання та видалення словників

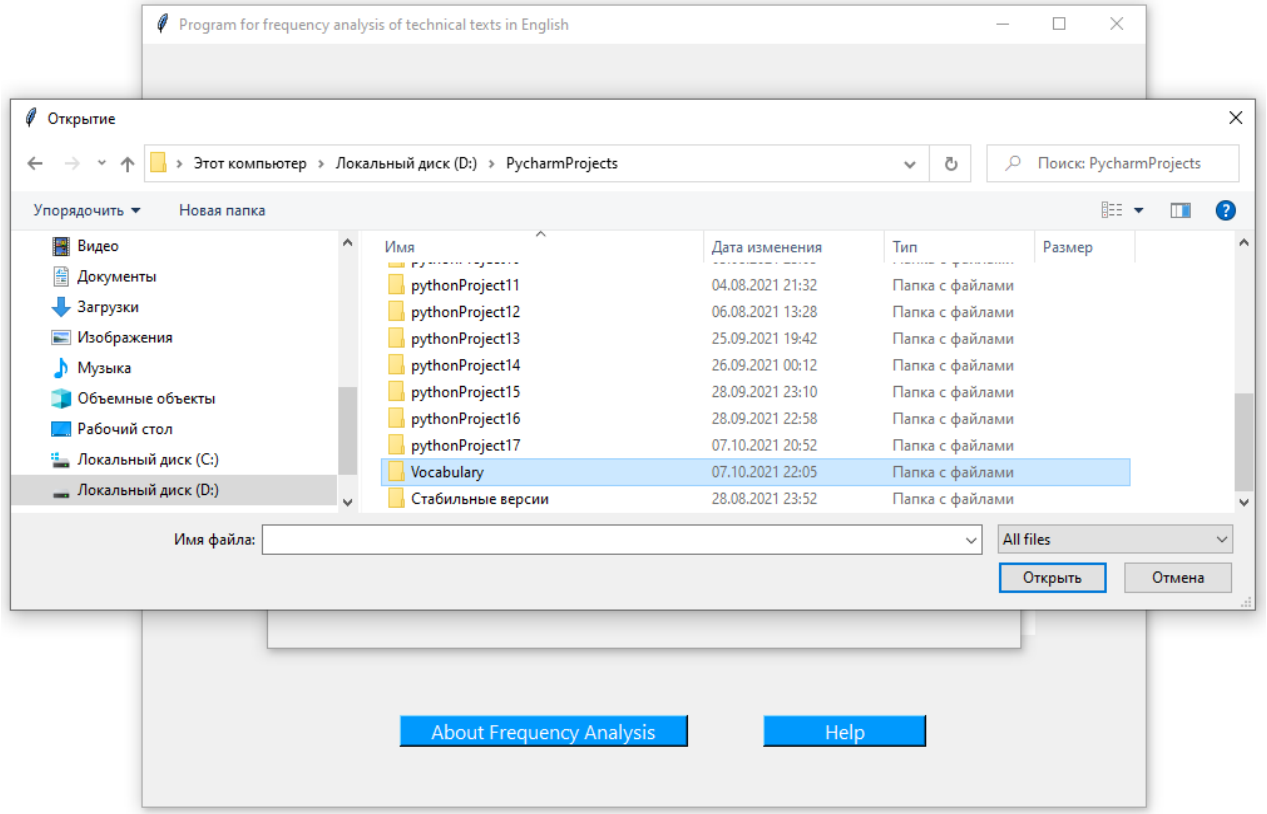

Рисунок 5.5 – Зберігання словників

У фрейм «English text»завантажується текст для частотного аналізу.

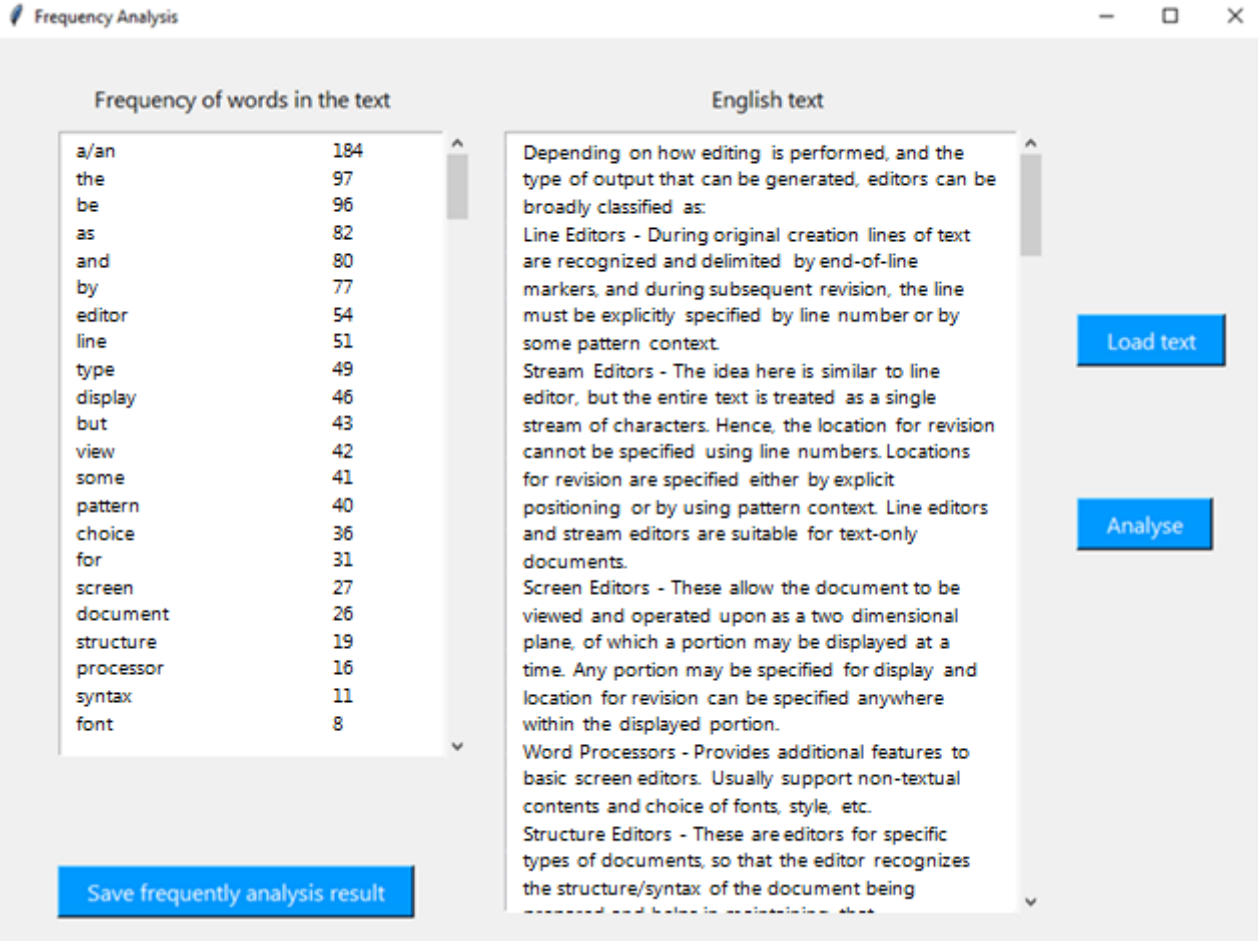

Рисунок 5.6 – Обчислення частот слів

Для завантаження тексту потрібно натиснути кнопку «Load text»та обрати файл у форматі txt. На рис. 5.6 завантажено фрагмент тексту за тематикою інформаційних технологій.

При натисканні кнопки «Analyse» з тексту виділяються слова, приводяться до нормальної форми та підраховується кількість входжень у текст кожного слова. Потім слова вопрядковуються у порядку спадання кількості входжень. У фреймі «Frequency of words in the text»відображуються слова у нормальній формі та кількості їх входжень з можливістю скроллінгу. Якщо у тексті будуть зустрічатись слова з іншої мови, то програма проігнорує їх. Слова частин мови, крім іменника, дієслова, прикметника та прислівника вже є у нормальній формі, їх перетворення не потрібно. Для збереження результатів частотного аналізу у базі даних потрібно натиснути кнопку «Save frequently analysis result».

Результати частотного аналізу можуть бути використані для формування власного словника, що допомагає читати та розуміти певний текст. На рис. 5.7 показано 2 фрейми. У лівий фрейм завантажено певний результат частотного аналізу за допомогою кнопки «Load words' frequency».У правий фрейм завантажено слова, що зараз містяться у власному словнику користувача. Для більш зручного користування зверху над правим фреймом розташовано випадаючий список, з якого можна обрати першу букву слів словника. Поряд з кожним словом у власному словнику розташовано переклад російською мовою, який спочатку визначається за допомогою стандартних словників-перекладачів StarDict.

Для того, щоб додати слово у власний словник, потрібно обрати його зі фрейму «Frequency of words in the text» та натиснути кнопку «Add».

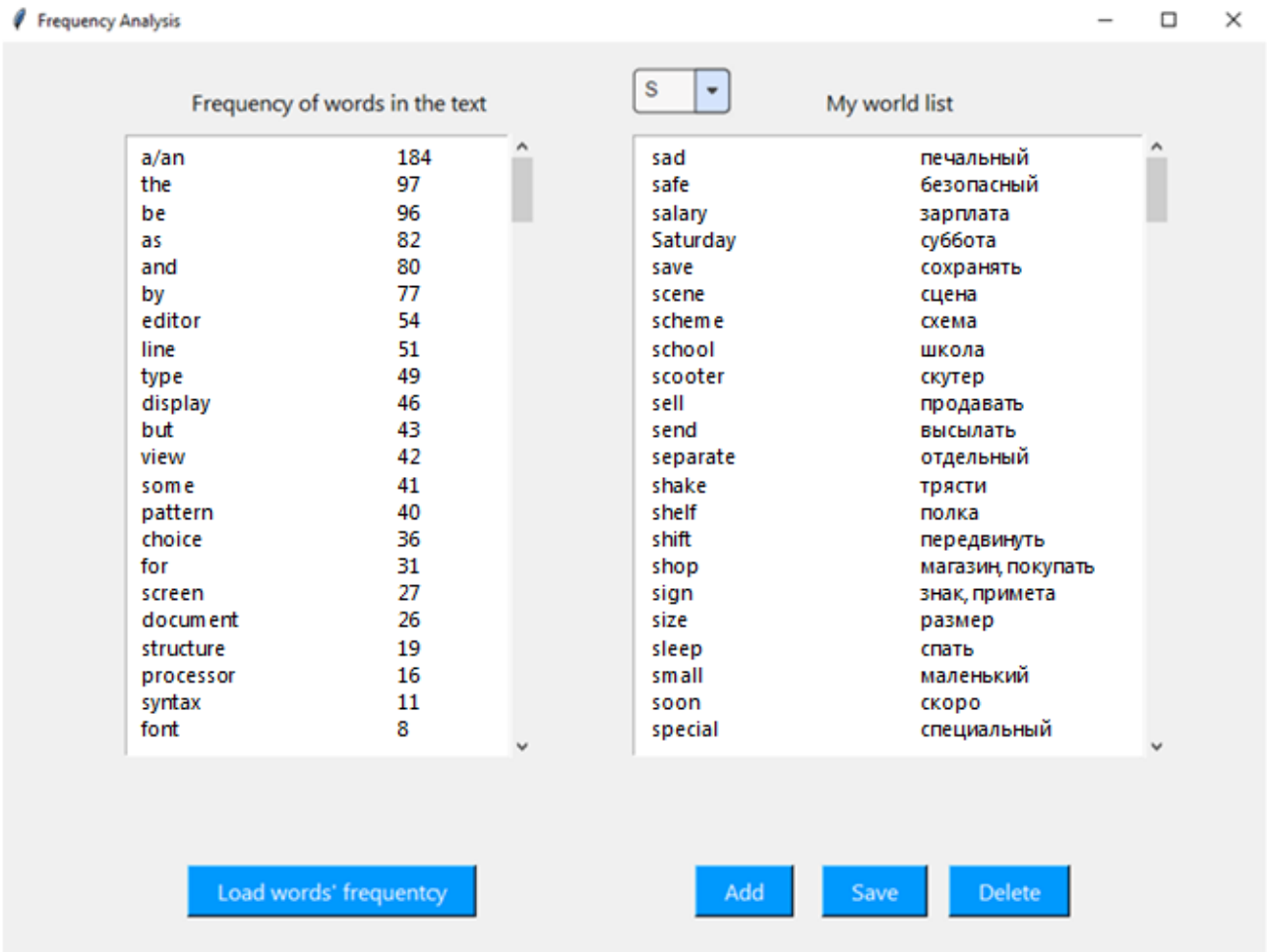

Рисунок 5.7 – Відображення власного словника

На рис. 5.8 показано результат додавання у власний словник слова «screen». Зазначимо, що при додаванні слів у словник вони упорядковуються відповідно до лексикографічного порядку.

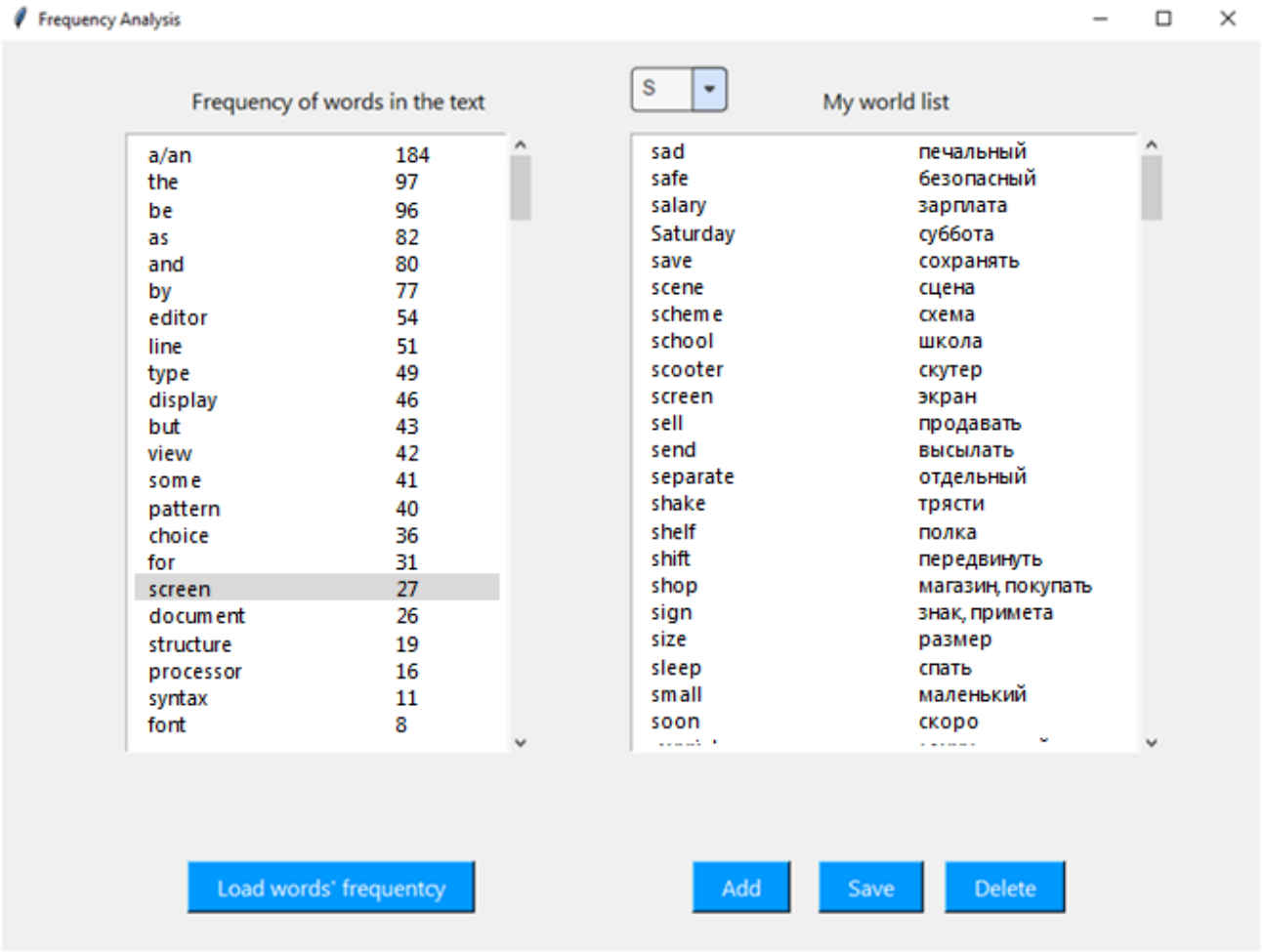

Рисунок 5.8 – Приклад додавання слів із списку

Слова у власному словнику можна редагувати. Для цього достатньо двічі клацнути мишею по слову, відредагувати його та натиснути «Enter».Для видалення слова з власного словника його потрібно обрати та натиснути кнопку «Delete».

На рис. 5.9 показано результат змін у словнику:

– редагування: для слова «salary»додано альтернативний переклад;

– видалення: слова «Saturday» та «save» видалено зі словнику.

Для збереження змін у словнику потрібно натиснути кнопку «Save».

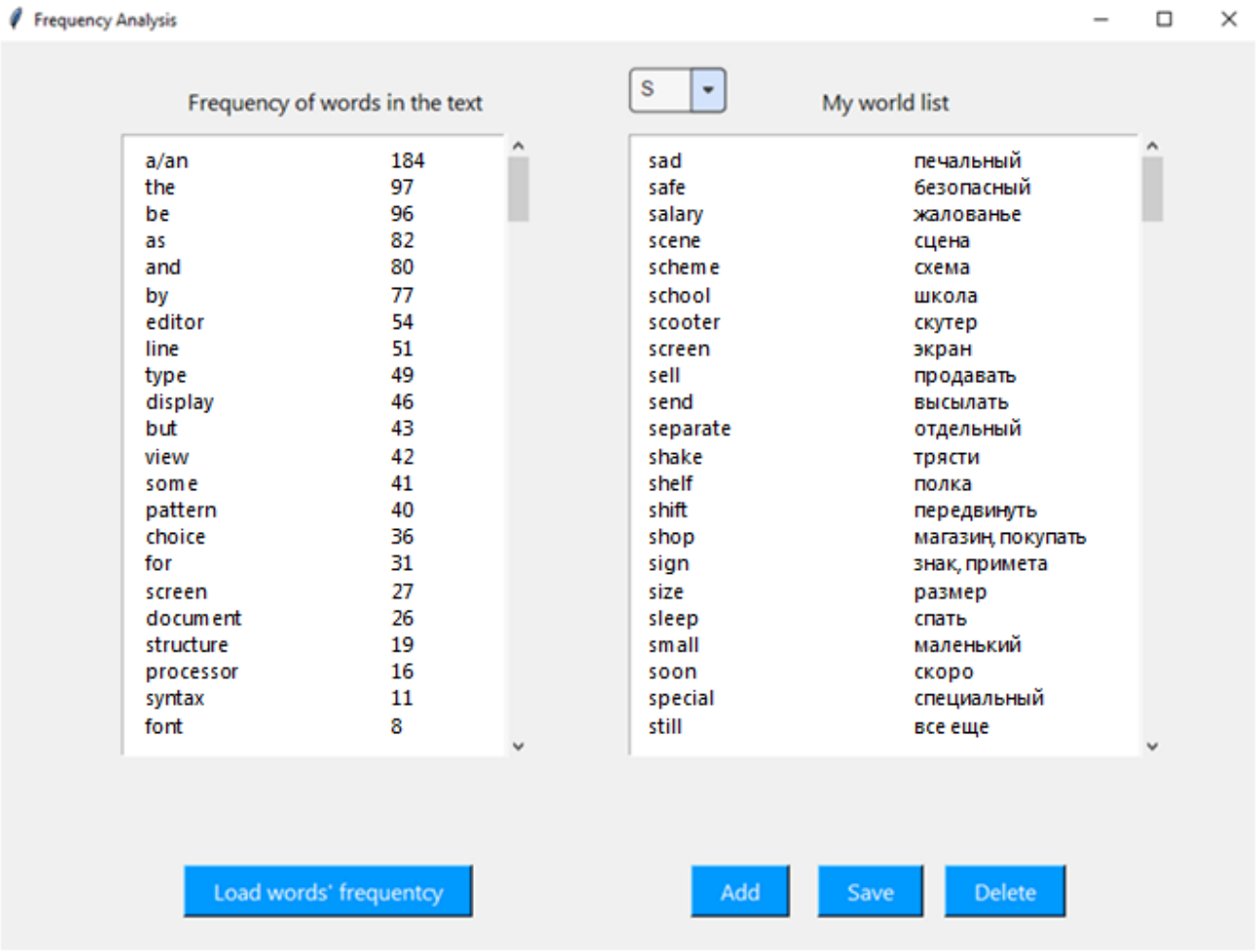

Рисунок 5.9 – Модифікація власного словника

5.3 Висновки до розділу

У розділі визначено програмні інструменти для розробки програмного засобу. Описано використання словників бази WordNet та електронного словнику StarDict.

Наведено приклади інтерфейсу для проведення частотного аналізу слів англійського тексту та для подальшого використання результатів цього аналізу для створення та модифікації власного словника.

6 ВИЗНАЧЕННЯ ВЛАСТИВОСТЕЙ ЧАСТОТНОГО АНАЛІЗАТОРА

6.1 Тестування моделі лематизатора

Визначимо метрики верифікації длянайбільш складної частини системи – лематизатора. Лематизація слова, що відноситься до будь-якої з розглянутих частин мови (дієслово, іменник, прислівник чи прикметник) передбачає отримання нормальної форми слова на основі той форми, що зустрілась в аналізованому тексті. Оберемо для тестування досить великий текстовий фрагмент.

Результат «True Positive» означає, що нормальну форму слова отримано, і нормальна форма є правильною. Результат «True Negative» означає, що нормальну форму слова отримати неможливо (наприклад, слово записано помилково та відсутнє у словниках, чи воно є власним ім’ям), і цей висновок є правильним. Результат «False Positive» (помилка 1 роду) означає, що нормальну форму слова отримано, і нормальна форма є неправильною. Результат «False Negative» (помилка 2 роду) означає, що нормальна форма слова існує, але програма не визначила її.

Результати визначення правильних і помилкових процедур лематизації для слів в залежності від їх частин мови, наведені у табл. 6.1 – 6.4.

Таблиця 6.1 – Верифікація лематизації дієслів

	True	False
True	528	18
False	15	13

Таблиця 6.2 – Верифікація лематизації іменників

	True	False
True	1927	35
False	38	63

Таблиця 6.3 – Верифікація лематизації прислівників

	True	False
True	187	19
False	21	15

Таблиця 6.4 – Верифікація лематизації прикметників

	True	False
True	394	27
False	42	32

У табл. 6.5 наведені значення точності лематизатора для слів з кожної частини мови. Як можна побачити, точність є досить високою, тому можна вважати, що лематизатор розроблено якісно.

Таблиця 6.5 – Верифікація моделі лематизатора

№ п/п	Назва частини мови	Точність= TP / (TP + FP)
1.	дієслово	0.97
2.	іменник	0.98
3.	прислівник	0.89
4.	прикметник	0.90

6.2 Тестування з діаграмою причино–наслідкових зв’язків

Виконаємо тестування за допомогою діаграми причино-наслідкових зв’язків [15] для сценаріїв варіантів використання «Завантаження словників», «Завантаження текстів» та «Відбір слів».

Сценарій «Завантаження словників»

Причини:

1) Роль користувача – Administrator
2) Користувачеві доступен словник
3) Технічна умова: словник у правильному форматі

Наслідки:

101) Успішне завантаження

102) Помилка «невірний формат словника»

103) Помилка «словник не знайдено»

На рис. 6.1 показана функціональна діаграма причино-наслідкових зв’язків для сценарію «Завантаження словників».

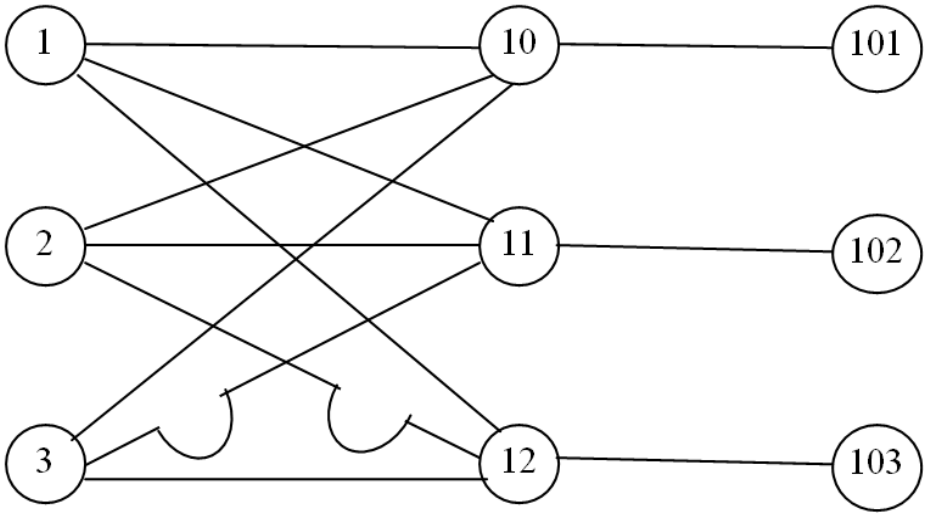

Рисунок 6.1 – Функціональна діаграма для сценарію «Завантаження словників»

Далі у табл. 6.6 побудовано таблицю рішень.

Визначимо тестові набори

1) Користувач є адміністратором, йому доступне словник, словник має коректний формат.

2) Користувач є адміністратором, йому доступне словник, але словник має невірний формат.

Таблиця 6.6– Таблиця рішень для сценарію «Завантаження словників»

Причини		1	2	3
	1	1	1	1
	2	1	1	0
	3	1	0	1
Наслідки	101	1		
	102		1	
	103			1

3) Користувач є адміністратором, йому недоступен словник (наприклад, він знаходиться на іншому носії), формат словника при цьому коректний.

Сценарій «Завантаження текстів»

Причини:

1) Роль користувача – User
2) Обрана команда для завантаження тексту
3) Обрано певний текст

Наслідки:

101) Успішне завантаження

102) Помилка «Не обрано текст»

103) Помилка «Неавторизований користувач»

На рис. 6.2 та у табл. 6.7 показана функціональна діаграма причино-наслідкових зв’язків та таблиця рішень для сценарію «Завантаження текстів».

Визначимо тестові набори для даного сценарію.

1) Користувач є User, він обрав команду для завантаження та певний текст.

2) Користувач є User, він обрав команду для завантаження, але не визначив текст для завантаження.

3) Користувач не є авторизованим User.

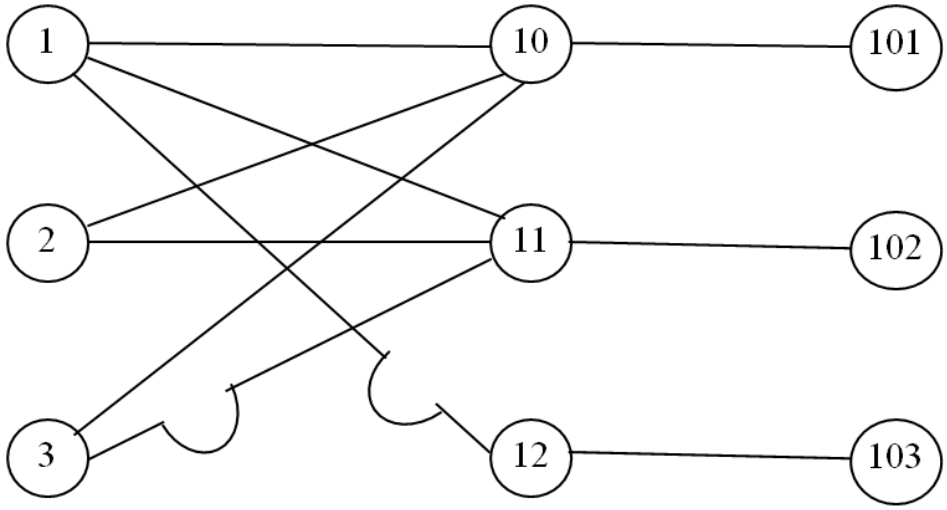

Рисунок 6.2 – Функціональна діаграма для сценарію «Завантаження текстів»

Таблиця 6.7– Таблиця рішень для сценарію «Завантаження текстів»

Причини		1	2	3
	1	1	1	0
	2	1	1	X
	3	1	0	X
Наслідки	101	1		
	102		1	
	103			1

Сценарій «Відбір слів»

Причини:

1) Частотний аналіз проведений.

2) Слово для додавання у власний список обрано.

Наслідки:

101) Успішне додавання

102) Помилка додавання

На рис. 6.3 та у табл. 6.8 показана функціональна діаграма причино-наслідкових зв’язків та таблиця рішень для сценарію «Відбір слів».

Визначимо тестові набори для даного сценарію.

1) Частотний аналіз проведено та слово для додавання у список обрано.
2) Частотний аналіз не проведено.
3) Не обрано слово для додавання у список.

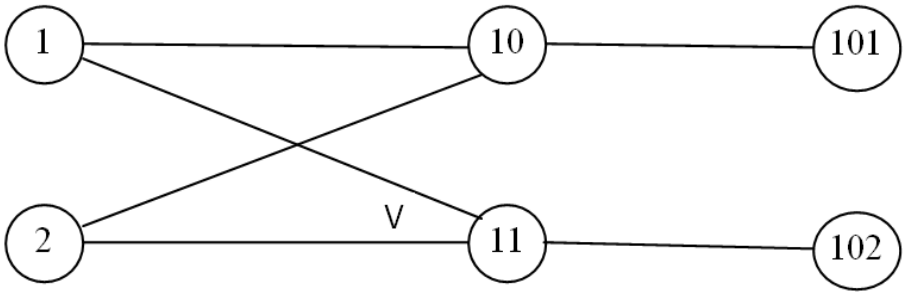

Рисунок 6.3 – Функціональна діаграма для сценарію «Відбір слів»

Таблиця 6.8 – Таблиця рішень для сценарію «Відбір слів»

Причини		1	2	3
	1	1	0	X
	2	1	X	0
Наслідки	101	1		
	102		1	1

6.3 Тестування ступеня зручності використання частотного аналізатора

Проведено тестування зручності використання частотного аналізатора. Зручність оцінювалась 5 користувачами з приблизно однаковим рівнем досвіду та освіти за 10-бальною шкалою.

На рис. 6.4 графічно представлені кількісні результати тестувань зручності процесу проведення частотного аналізу технічних текстів англійською мовою з використанням розробленої програми та без неї (з використанням будь-яких інших засобів).

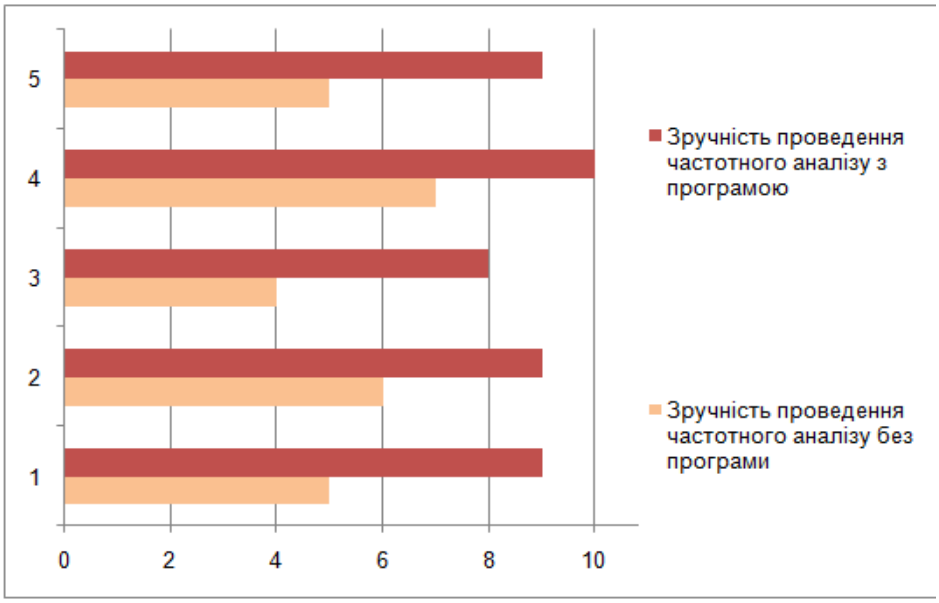

Рисунок 6.4 – Результати тестування зручності проведення частотного аналізу

Аналіз усереднених результатів показав, що зручність була підвищена приблизно у 1.7 разів.

6.4 Висновки до розділу

У шостому розділі виконано тестування моделі лематизатора. Визначено, що точність лематизації належить діапазону від 0.89 до 0.98, що можна вважати якісним результатом.Виконано тестування за допомогою діаграми причино-наслідкових зв’язків. Експериментальна перевірка зручності проведення частотного аналізу показала, що використання розробленого програмного засобу дозволяє підвищити зручність приблизно у 1.7 разів.

ВИСНОВКИ

У роботі розроблено програмний застосунок для проведення частотного аналізу технічних текстів англійською мовою. Експериментальна перевірка зручності проведення частотного аналізу з використанням розробленого програмного засобу показала, що зручність підвищено приблизно у 1.7 разів.

Критичний аналіз існуючих рішень містить огляд засобів для обробки неструктурованого тексту та частотного аналізу слів, що містяться у ньому. Розглянуто існуючі проблеми, практичні задачі, для яких корисно застосовувати обробку тексту. Виконано огляд програмних аналогів для підрахунку частотності слів. Далі наведено принципи роботи частотного програмного аналізатору. Побудовано модель нормалізації англійських слів для таких частин мови як: дієслово, іменник, прислівник, прикметник. Розглянуто можливості використання засобів NaturalLanguageProcessing. Визначено спосіб верифікації результатів роботи частотного аналізатора. У розділі специфікації вимог до програмного продукту детально описано функціонал програми для виконання частотного аналізу з використанням UseCase. Наведено вимоги до нефункціональних характеристикпрограмного продукту. У розділі проєктування частотного аналізатору визначено архітектуру програмного продукту, створено UML-діаграми для опису роботи аналізатору. Виконано концептуальне та детальне проектування структури та організації класів. Далі наведено опис програмної реалізації застосування, аналіз та обґрунтування вибору технологій та мови програмування, та вигляд інтерфейсу роботи програми. Для розробленого застосування виконано функціональне тестування. Проведено експеримент для оцінки зручності користування продуктом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Krisilov, V. A. & Komleva, N. O. (2019). “Analysis and Evaluation of Competence of Information Sources in Problems of Intellectual Data Processing”. [Analiz i ocenka kompetentnosti istochnikov informacii v zadachah intellektual'noj obrabotki dannyh]. Problemele Energeticii Regionale, Vol. 1-1(40), pp. 91-104 DOI: 10.5281/zenodo.3239184 (in Russian).

2. Tyshchenko, A. I., Onishchenko, T. V. & Pysarenko, K. O. “The Web-Interface Availability Modelfor People with Disabilities”. Herald of Advanced Information Technology. Publ. Science i Technical.2019; Vol.2 No.3:p.206–214. Odesa. Ukraine. DOI: https://doi.org/10.15276/hait.03.2019.4

3. Komleva, N. O., Liubchenko, V. V. & Zinovatnaya S. L. “Methodology of Information Monitoring and Diagnostics of Objects Represented by Quantitative Estimates Based on Cluster Analysis”. Applied Aspects of Information Technology. Publ. Nauka i Tekhnika. Odessa: Ukraine. 2020; Vol. 3 No. 1: 376–392. DOI: https://doi.org/10.15276/aait.01.2020.1

4. Liubchenko, V., Komleva, N., Zinovatna, S. & Pysarenko, K. “Framework for Systematization of Data Science Methods”. Applied Aspects of Information Technology. Publ. Nauka i Tekhnika. Odessa: Ukraine. 2021; Vol.4 No.1: 80–90. DOI: https://doi.org/10.15276/aait.01.2021.7

5. Komleva N.O., Cherneha K.S., Tymchenko B.I., Komlevoy O.M. Intellectual approach application for pulmonary diagnosis. Proceedings of the 2016 IEEE 1st International Conference on Data Stream Mining and Processing, DSMP, 2016, Article № 7583505, pp. 48-52.

6. Комлевая Н.О., Комлевой А.Н., Тимченко Б.И. Сравнительный анализ двух подходов при решении задачи классификации. – Науко-технічний журнал "Радіоелектронні і комп'ютерні системи". – Харьков, 2014. – № 6(70). –С. 115 – 119.

7. Лейн Х., Хапке Х., Ховард К. Обработка естественного языка в действии. – СПб: Питер, 2020. – 503 с.

8. Wordstarter для WordStat [Електронний ресурс]: – Режим доступу: https://chrome.google.com/webstore/detail/wordstater-%D0%B4%D0%BB% D1%8F-

wordstat-%D1%81/hjlgbdmfljafjdkpgdiefkplpkcjlphh?hl=ru. – Загл. з екрану: дата звернення 15.11.2021.

9. Pymorphy2 [Електронний ресурс]: – Режим доступу: https://pymorphy2.readthedocs.io/en/stable/. – Загл. з екрану: дата звернення 15.11.2021.

10. StatSoftText-Analizer [Електронний ресурс]: – Режим доступу: http://statsoft.ru/solutions/ready_solutions/another/text_analizer.php#tab-examples-link. – Загл. з екрану: дата звернення 15.11.2021.

11. TextOBRAZ [Електронний ресурс]: – Режим доступу: http://www.sbup.com/seo-forum/seo_servisy__seo_instrumenty_i_seo_utility/ textobraz_2_1_bloknot_dlya_optimizatora/. – Загл. з екрану: дата звернення 15.11.2021.

12. WordNet: A Lexical Database for English [Електронний ресурс]: – Режим доступу: https://wordnet.princeton.edu/. – Загл. з екрану: дата звернення 15.11.2021.

13. Шварц Б., Зайцев П., Ткаченко В. MySQL по максимуму: оптимизация, репликация, резервное копирование. – СПб: Питер, 2019. – 243 с.

14. Stardict [Електронний ресурс]: – Режим доступу: https://soft.mydiv.net/win/download-Stardict.html. – Загл. з екрану: дата звернення 15.11.2021.

15. Канер С., Фолк Дж., Нгуен Е. К. Тестирование программного обеспечения. – Киев: ДиаСофт, 2019. – 315 с.

16. Комлева Н.О., Вонгуе Нгангом Франсуа Жералдін. Проведення частотного аналізу технічних текстів англійською мовою // Topical issues of modern science, society and education. Proceedings of the 5th International scientific and practical conference. SPC “Sci-conf.com.ua”. Kharkiv, Ukraine. 2021. Pp. 493-499. URL: https://sci-conf.com.ua/v-mezhdunarodnaya-nauchno-prakticheskaya-konferentsiya-topical-issues-of-modern-science-society-and-education-28-30-noyabrya-2021-goda-harkov-ukraina-arhiv/