

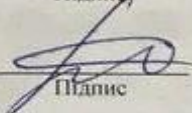


## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

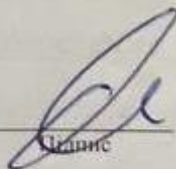
на тему Метод автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками

Галузь знань 12 – Інформаційні технології  
Шифр і назва галузі знань  
Спеціальність 122 – Комп'ютерні науки  
Шифр і назва спеціальності  
Освітня програма Комп'ютерні науки  
Назва освітньої програми

Виконав: студентка 4 курсу, група КН-18-1  О.О. Залуцька  
Курс, група виконавця Підпис Ініціали, прізвище  
Керівник: викладач кафедри КН  М.О. Молчанова  
Науковий ступінь, посада Підпис Ініціали, прізвище  
Нормоконтроль: к.т.н., доцент кафедри КН  Р.О. Багрій  
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор

 О.В. Бармак  
Підпис Ініціали, прізвище

1 червня 2022 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

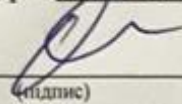
Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук



(підпис)

д.т.н., професор О.В. Бармак

«25» березня 2022 року

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Метод автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками»

2. Завдання видано студентці Залуцькій Ользі Олександрівні

(прізвище, ім'я, по батькові)

3. Керівник роботи викладач кафедри КН Марина Олексіївна Молчанова

(посада, прізвище, ім'я, по батькові)

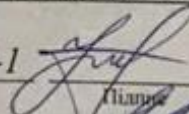
4. Затверджено наказом університету від «01» березня 2022 р. № 18

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – розробка методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах та його програмна реалізація. Система має виконувати ряд функцій, а саме: обрахунок оцінки семантичної важливості обраного унікального слова, визначення множин слів, які містять семантичне забарвлення, обрахунок числових оцінок психоемоційної тональності текстового повідомлення та їх візуальне відображення психоемоційної тональності слів повідомлення.

Виконавець: студентка 4 курсу, група КН-18-1

Курс, група виконавця



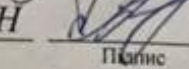
Підпис

О.О. Залуцька

Ініціали, прізвище

Керівник: викладач кафедри КН

Науковий ступінь, посада



Підпис

М.О. Молчанова

Ініціали, прізвище

## Анотація

Тема кваліфікаційної роботи бакалавра: «Метод автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками»

Виконавець кваліфікаційної роботи бакалавра: студентка групи КН-18-1 Залуцька Ольга Олександрівна

Керівник кваліфікаційної роботи бакалавра: викладач кафедри КН Марина Олексіївна Молчанова

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
56	30	14	37	3

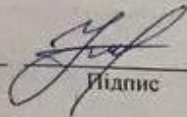
Метою кваліфікаційної роботи бакалавра є розробка методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками, та його програмна реалізація на базі системи аналізу повідомлень на платформі .NET. Для розробки інформаційної системи було використано мову програмування C#, а також систему керування базами даних MS SQL Server.

Розроблена система призначена для працівників сфер, що мають досліджувати емоційне забарвлення текстів у соціально-орієнтованих сервісах, таких як Twitter, Facebook, Instagram та інших. Реалізована система допомагатиме визначити емоційну тональність текстів, що допоможе підвищити продуктивність роботи в тих сферах, де це необхідно.

Ключові слова: NLP, емотивний аналіз тексту, визначення тональності, TF-IDF, YAKE, TextRank.

Виконавець: студентка 4 курсу, група КН-18-1

Курс, група виконавця

  
Підпис

О.О. Залуцька  
Ініціали, прізвище

## Зміст

Перелік скорочень .....	6
Вступ.....	7
Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій	
1.1 Аналіз інформаційних моделей .....	9
1.2 Огляд теоретичних підходів до розв’язку подібних задач .....	12
1.3 Аналіз існуючих програмних рішень.....	16
1.4 Аналіз сучасних засобів створення програмного забезпечення .....	20
1.5 Мета, задачі та вимоги до реалізації інформаційної системи .....	23
Розділ 2 Проектування інформаційної системи	
2.1 Аналіз та автоматизація обробки потоків даних .....	24
2.1.1 Методи автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками.....	24
2.1.2 Математично-алгоритмічне забезпечення .....	25
2.2 Інформаційна структура системи .....	27
2.2.1 Проектна архітектура системи та взаємозв’язок компонентів.....	27
2.2.2 Інформаційна модель.....	29
2.3 Вибір засобів розробки інформаційної системи .....	36
2.3.1 Вибір мови програмування .....	36
2.3.2 Вибір фреймворку .....	37
2.3.3 Вибір редактору програмного коду .....	38
2.3.4 Вибір СКБД .....	39
Розділ 3 Програмна реалізація інформаційної системи	
3.1 Структура та функціональне призначення програмних складових системи.....	42
3.2 Особливості реалізації програмних складових системи .....	45
3.3 Тестування інформаційної системи .....	49
3.4 Інструкція користувача.....	54
3.5 Вимоги до розгортання інформаційної системи.....	59

Висновки .....	61
Перелік посилань.....	63
Додатки	

## Перелік скорочень

<b>Скорочення, термін, позначення</b>	<b>Пояснення</b>
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
ІС	Інформаційна система
ІТ	Інформаційні технології
БД	База даних
ХНУ	Хмельницький національний університет
TF	Term Frequency
TF-IDF	Term Frequency – Inverse Document Frequency
СКБД	Система керування базами даних
MS	Microsoft
SQL	Structured Query Language
VS	Visual Studio
NLP	Natural language processing

## Вступ

Збільшення обсягів інформації та щоденний розвиток інформаційних технологій, беззаперечно, надають людству незліченну кількість переваг, роблячи знання доступними для кожного. Однак чи в абсолютній більшості випадків користувачеві Інтернету вдається знайти необхідний матеріал, а що найголовніше – верифіковане й надійне джерело? Звідси і виникає проблема виокремлення корисного, достовірного матеріалу серед колосальних обсягів науково-технічної літератури, документів та статей. Окрім того, часто шуканий матеріал не має чітко визначеної мети чи змісту, що змушує виконувати зайву роботу, витрачаючи власний час.

За допомогою аналізу граматичних структур, знаходячи відношення між словами у певному визначеному контексті, методи семантичного аналізу тексту надають можливості обчислювальній машині розпізнавати та розуміти морфемні конструкції: слова, речення, окремі цілісні документи – тобто всю цифрову текстову інформацію, якою оточена людина.

Недосконалість або ж цілковита відсутність сервісів та технологій для автоматизованої обробки текстової інформації за різними критеріями – одна із найбільших проблем ІТ, що досі вимагають вирішення. Для змістовної обробки текстової інформації необхідно реалізувати комплекс методів, що забезпечують реалізацію алгоритмів лінгвістичного аналізу текстів, що в свою чергу, ще більше ускладнює вищенаведену проблему.

Задля реалізації автоматизованого смислового аналізу цифрової інформації мають бути впроваджені високопродуктивні способи організації обчислювального процесу, адже відсутність науково-методичного базису призводить до технологічних обмежень та низької ефективності в системах збереження та аналізу інформації.

За допомогою методів та систем семантичного аналізу можуть бути вирішені задачі інтерпретації слів та фраз, що можуть створити неточності для сприйняття обчислювальною машиною, цим забезпечивши продуктивний аналіз

цифрової інформації. Окрім того, автоматичне вилучення інформації, що може представляти цінність та її обробка можуть допомогти в роботі різних компаній: наприклад, із електронних листів та відгуків клієнтів можна визначити емотивне забарвлення цих повідомлень, що є важливим зрізом інформації для керівників бізнесу.

Аналіз емотивної тональності тексту – це задача обробки природної мови, що широко використовується в таких областях, як соціологія, політологія, маркетинг, медицина та психологія. Щодня ці методи та моделі вдосконалюються, однак досі повноцінної системи аналізу емоційного забарвлення текстів українською мовою досі не розроблено, що робить тему кваліфікаційної роботи бакалавра більш актуальною.

В процесі виконання кваліфікаційної роботи бакалавра об'єктом дослідження стане процес визначення психоемоційного забарвлення цифрових текстових повідомлень, предметом – технології, методи, алгоритми та засоби створення систем для визначення емоційного контексту повідомлень, твітів, дописів та інших текстів, метою – створення методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками, та його програмна реалізація на платформі .NET.

## Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій

### 1.1 Аналіз інформаційних моделей

В сучасному науковому просторі значна кількість уваги відведена аналізу електронного тексту[1], адже саме він є основним джерелом інформації. Існує кілька підходів до його аналізу. За допомогою методів семантичного аналізу можна визначати ідею, тему текстів, оцінювати смислове навантаження чи визначати тему, із якою пов'язаний текст: математика, соціологія, маркетинг, література, тощо [2].

Щодня користувачі Інтернету залишають сотні тисяч відгуків, коментарів, твітів, рецензій, тощо. Переважно, ця інформація у неструктурованому форматі, потребує обробки та залучення високопродуктивних методів та ресурсів [3]. Деякі статистичні дані щодо публікацій у соціально-орієнтованих сервісах наведено у таблиці 1.1.

Соціально-орієнтований сервіс [4] – інтернет-спільнота користувачів, об'єднаних за певною ознакою на базі певного сайту чи застосунку.

Таблиця 1.1 – Статистичні дані щодо кількості публікації в секунду у популярних сервісах [4].

<b>Назва соціально-орієнтованого сервісу</b>	<b>Кількість публікацій в секунду</b>
Facebook	55017
Twitter	8965
Instagram	7719
Reddit	39

Без залучення методів штучного інтелекту та аналізу даних неможливо опрацювати такі обсяги інформації, що генерується чи не щосекунди, саме тому

значна частка задач на ринку ІТ пов'язана із NLP, семантичним аналізом та визначенням емоційного забарвлення цифрових текстів.

Сьогодні існує чимало проєктів для дослідження психоемотивного забарвлення текстів для аналізу відгуків про готелі, банківські установи, ресторани, кінотеатри, товари, події, тощо [5]. Для забезпечення їх роботи використовуються методи, засновані на словниках, підходах, що базуються на правилах чи використанні машинного навчання [6]. Та особливо важливим є використання цих методів при опрацюванні текстів українською мовою: досліджень на цю тематику небагато, що і формує особливу актуальність роботи.

Мова – досить складне утворення, тому в комп'ютерній лінгвістиці розвиваються різноманітні напрямки, такі як [7]:

- автоматизований аналіз текстів;
- створення, підтримка автоматично створених словників;
- створення автоматизованих інформаційно-пошукових систем;
- переклад із залученням машинного навчання;
- дешифрування та кодування текстів;
- підтримка лінгвоморфемних баз даних;
- реалізація програмних продуктів для рішення задач теоретичної та прикладної лінгвістики.

При автоматизованому аналізі тексту можна стикнутись із явищами, що ускладнюють ці процеси. Унікальні мовленнєві явища, такі як полісемія, омонімія, тощо, вносять в мову неточності, незрозумілі для обчислювальної машини, проте ясні для носія мови. Виняткові особливості мови ускладнюють встановлення правильної семантично-синтаксичної структури тексту в його формальне логічне представлення.

Необхідно розглянути ці поняття детальніше для чіткішого окреслення предметної області. Полісемія [8] – морфологічне явище, що означає «багатозначність», вважається, що семантична одиниця має декілька значень. Омонімія [9] – явище, за якого два слова, зберігають однакове звучання, але набувають різного значення.

Незважаючи на стрімкий розвиток ІТ індустрії, в сфері NLP є великий пласт невирішених досі задач, зокрема питань української мови. Однією з причин є ізольованість процесів аналізу на етапах дослідження тексту. Обчислювальна машина не може оцінювати інформацію як людський мозок, адже вихідні дані одного процесу є вхідними для наступного, в той час, як людина здатна синхронно оцінювати декілька явищ, формулюючи унікальний висновок.

Важливим аспектом при аналізі тексту є виокремлення ключових слів. Ключове слово [10] – одиниця природної мови, що використовується для вираження певного аспекту текстового матеріалу, мають велике смислове значення і формують в сукупності множину понять, що містяться в матеріалі.

Властивості ключових термінів:

1. ключові терміни – найважливіші слова, що характеризують предмет, явище, про яке йдеться в тексті.
2. Не мають оцінки, є нейтральними за стилістикою.
3. Мають семантичний зв'язок між собою
4. Зазвичай ключові терміни формують множину розмірністю 5-15 слів.

За останнє десятиліття інтерес до галузі аналізу емоційної тональності текстів сильно зріс. Однак, незважаючи на постійний розвиток рішень у цій галузі, невирішених проблем досі багато. Аналіз емоційного забарвлення тексту ускладнюється не тільки через проблему виділення одиниць оцінки тональності, а також через неоднозначність емоційної складової лексичних компонентів [10]. Наприклад, у межах одного ж тексту «висока ціна» – може бути негативною оцінкою товару, тоді як «висока якість» – позитивною.

Для аналізу тональності тексту використовуються статистичні методи, але їх основний недолік у тому, що для коректної роботи необхідно забезпечити наявність корпусу слів мови, що трудомістким та дорогим засобом.

Призначення методів аналізу емоційного забарвлення тексту полягає у виявленні в текстах лексики, що містить емоційне забарвлення, визначення

особистісної оцінки автора щодо об'єктів, про які йдеться в тексті. Текстові матеріали можна розділити на дві категорії: факти та думки.

Окрема галузь в сфері ІТ – NLP, мета якої обробка предметної мови, знаходження емотивного забарвлення текстів.

Українська мова є унікальним явищем, адже залежно від порядку слів у реченні може кардинально змінитись і його зміст, і його емотивність. Отож при проведенні семантичного аналізу можуть виникнути труднощі, на які варто зважати при побудові методів аналізу та реалізації програмного продукту.

Методи NLP можуть бути застосовані при дослідженні громадської думки щодо різноманітних товарів, послуг, політичні та інші події. Такі дані можуть використовуватись в маркетингових дослідженнях, аналізі зв'язків з громадськістю, тощо. Дані можна використати в комерційних дослідженнях: маркетинговий аналіз – лише єдина сфера, де вже необхідно створювати відповідні рішення.

Таким чином, NLP, методи визначення емотивного забарвлення тексту, семантичний аналіз – сфера ІТ, що потребує постійної підтримки та введення нових методів, адже за допомогою автоматизованого аналізу тексту можна вирішити чимало задач.

## **1.2 Огляд теоретичних підходів до розв'язку подібних задач**

Аналіз тональності текстів знаходить своє застосування, починаючи з оцінки якості товарів та послуг та закінчуючи складанням текстів із задалегідь заданими емоційними характеристиками. Об'єктом емоційного оцінювання може бути власне ім'я, назва продукту, організації, послуги чи професії, стосовно яких виражається думка [11]. Проте щоденна кількість публікованих відгуків у соціальних мережах досягає величезної кількості, тому обробка відгуків вручну виявляється неможливою і потребує автоматизації. Тому аналіз тональності текстів з кожним роком стає все більш актуальним завданням як з теоретичної,

так і з практичної точки зору. Це переважно пов'язано з розвитком інтернету та зі зміною у форматі комунікацій між людьми.

Визначення емоційної полярності речення починається з аналізу речення на суб'єктивність. У випадку, якщо пропозиція має суб'єктивний характер, то можна вважати, що у ньому виражається авторська думка. Потім визначають тональність речення: негативне воно чи позитивне та на скільки відсотків вірогідна така оцінка.

Існує чотири основних підходи для аналізу тексту на емоційне забарвлення [12]:

- підхід на правилах;
- підхід із словником;
- машинне навчання;
- навчання без вчителя.

Підхід на словниках (lexicon-based approach) заснований на застосуванні набору правил, використовуючи які система формує висновок про тональність тексту [13].

Було розглянуто простий приклад: речення «Я люблю вчитися». Можна застосувати таке правило: якщо присудок «люблю» входить у позитивний набір дієслів (як-от «люблю», «подобається», «схвалюю») і в реченні немає заперечень, то тональність класифікується як позитивна.

Підхід, заснований на словниках вимагає оформлення бази правил, що є досить трудомістким та дороговартісним процесом. У більшості випадків правила пов'язані із певною тематикою.

Цей підхід полягає в аналізі тональності слів та фраз, представлених у тексті. Найчастіше аналіз засновується на використанні прикметників, як показників тональності («хороший», «ненависний», «любий») або ж поєднанні прикметника та прислівника [13].

Саме такі слова в поєднанні з їх емоційною забарвленістю формують тональний словник [14]. Словники формуються як вручну, так і автоматично, в

звичайному вигляді словники тональності представляють собою певний список слів із значенням рівня тональності для кожного окремого слова.

У таблиці 1.2 представлений приклад слів, перекладених українською, з нормативними емоційними оцінками з бази ANEW – Affective Norms for English Words – набір із 1034 слів, що характеризуються афективними параметрами валентності.

Таблиця 1.2 – Тональність слів із бази ANEW

Слово	Валентність
Щасливий	8,21
Хороший	7,47
Нудний	2,95
Сердитий	2,85
Сумний	1,61

Як видно з таблиці 1.1, для аналізу тексту використовується наступний алгоритм: кожному слову в тексті надається значення тональності, де число означає валентність від 1 до 9; далі обчислюється загальна тональність всього тексту, наприклад, за допомогою середнього арифметичного всіх значень. Чим вище набуте значення, тим позитивніше вважається тональність. Чим нижче значення, тим негативнішим емоційним забарвленням володіє текст. Використання методів лінгвістичного аналізу із застосуванням тональних словників є простим у застосуванні і має високий результат у визначення емоційного забарвлення тексту.

Для того, щоб перейти до визначення тональності тексту, спершу необхідно визначити ключові слова. Дослідимо методи для визначення таких термінів.

Алгоритм TextRank [15]. Цей алгоритм походить від відомого PageRank, але є більш вдосконаленим. Схожість цих двох методів полягає в наступному:

оцінки подібності зберігаються у квадратній матриці, подібної до матриці, що використовується для PageRank.

TextRank – це техніка узагальнення тексту без контролю. Хід алгоритму TextRank, якого дотримуються при його використанні:

- Виділення необхідного тексту та його об'єднання.
- Текст розбивається на окремі речення.
- Нормалізація тексту, приведення його в нижній регістр, відсікання символів, чисел, зайвих пробілів, тощо.
- Векторизація тексту, обчислення кількості слів, їх частоти повтору.
- Подібність між векторами речень обчислюється і зберігається в матриці.
- Перетворення матриці подібності на граф, з реченнями як вершинами і оцінками подібності як ребрами, для обчислення рангу пропозиції.
- Нарешті, певна кількість речень з найвищим рейтингом утворюють остаточне представлення.

Наступний метод для виділення ключових термінів тексту – YAKE [16]. YAKE – це легкий неконтрольований метод автоматичного вилучення ключових слів, який базується на статистичних характеристиках тексту, витягнутих з окремих документів, для вибору найважливіших ключових слів тексту.

Системи, що засновані на методі YAKE не потребують навчання на певному набору, не залежать від словників, зовнішнього корпусу, розмірів тексту, мови чи типу документа.

Порівнявши метод YAKE найсучаснішими неконтрольованими підходами (TF-IDF, KP-Miner, RAKE, TextRank, SingleRank, ExpandRank, TopicRank, і MultipartiteRank), експериментально було доведено, що метод YAKE значно перевершує найсучасніші методи семантичного аналізу текстової інформації.

Переваги методу YAKE:

- підхід без нагляду;
- корпус-незалежний;

– незалежний від домену та мови.

Для вирішення задачі пошуку ключових слів існує вже багато методів, що різняться між собою за швидкістю роботи, ефективністю та метою використання. Серед них є як і багато спільного, так і відмінного, що доцільно використати в кваліфікаційній роботі бакалавра, забезпечивши цим її ефективність та унікальність.

### 1.3 Аналіз існуючих програмних рішень

Для семантичного аналізу тексту та дослідження емотивного забарвлення розроблено чимало систем, однак в кожній з них є свої недоліки й неточності. Для роботи із українською мовою складно знайти валідований застосунок, що забезпечував би коректну роботу із текстовою інформацією.

Найбільш популярні системи для роботи із антиплагіатом, SEO-аналізatori текстів, що визначають різні властивості матеріалу, як-от ключові терміни, частоту повторів слів, відсоток води, нудоту тексту, словник ядра.

Розглянемо одні із найпопулярніших системи аналізу тексту онлайн.

Було перевірено текст, що розміщений на головній сторінці сайту кафедри КН ХНУ [17] за допомогою онлайн-сервісу «Miratext» [18]. Цей застосунок виконує аналіз тексту, розміщеного на сайті за наступним параметрами:

- перелік ключових термінів;
- топ-20 найбільш уживаних слів;
- хмару слів за різними параметрами;
- найбільш вживані словосполучення з двох та з трьох слів;
- нудоту тексту;
- аналіз тексту по закону Ципфа.

Було наведено результати роботи аналізатору.

На рисунку 1.1 представлено таблицю із результатами загального аналізу сайту.



Ще однією функцією, що надає ця система, є визначення нудоти тексту (рисунок 1.4).

Нудота

Пошук

Слово	Повторень	Щільність, %
наука	20	0,67
комп	15	0,5
доцент	12	0,4
кафедрувати	12	0,4
ютерний	12	0,4
встановити	11	0,37

Рисунок 1.4 – Висновок щодо нудоти тексту платформою «Miratext» [18]

Також було наведено результат роботи системи щодо аналізу тексту за законом Ципфа (рисунок 1.5).

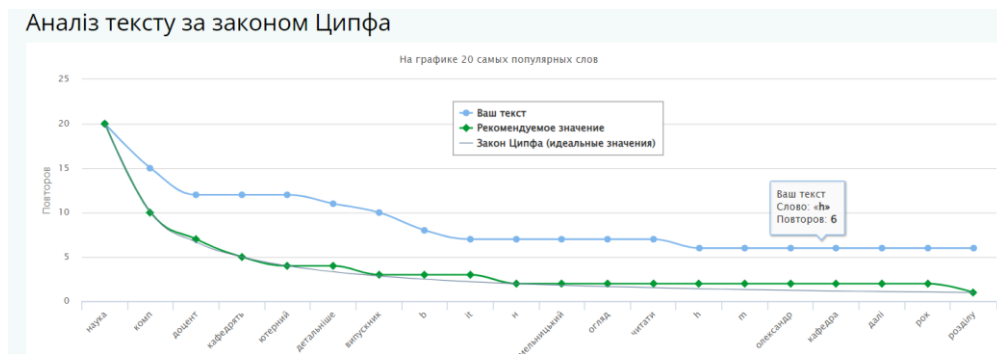


Рисунок 1.5 – Результат аналізу за законом Ципфа платформою «Miratext» [18]

Окрім виділення ключових слів, дослідимо програмні продукти, що допомагають визначити емоційне забарвлення тексту. Розглянемо сайт «MonkeyLearn» [19]. На комерційній основі він надає можливість не лише визначити емоційний тон повідомлень, а й сформуванати статистику для підприємств та установ, що працюють із відгуками та коментарями споживачів.

Для того, щоб проаналізувати текст – введемо його у відповідне поле «Enter your text here» та натиснемо кнопку «Classify text». Введений текст можна проаналізувати за трьома характеристиками: neutral, positive та negative (пер. нейтральний той, позитивний та негативний). Результат виводиться у вікні, що поруч, у відсотках. Переглянемо дію цього застосунку на практиці. Проаналізуємо наступне повідомлення «He is such an amazing friend» (пер. він – чудовий друг), що має позитивне забарвлення (рисунок 1.6).

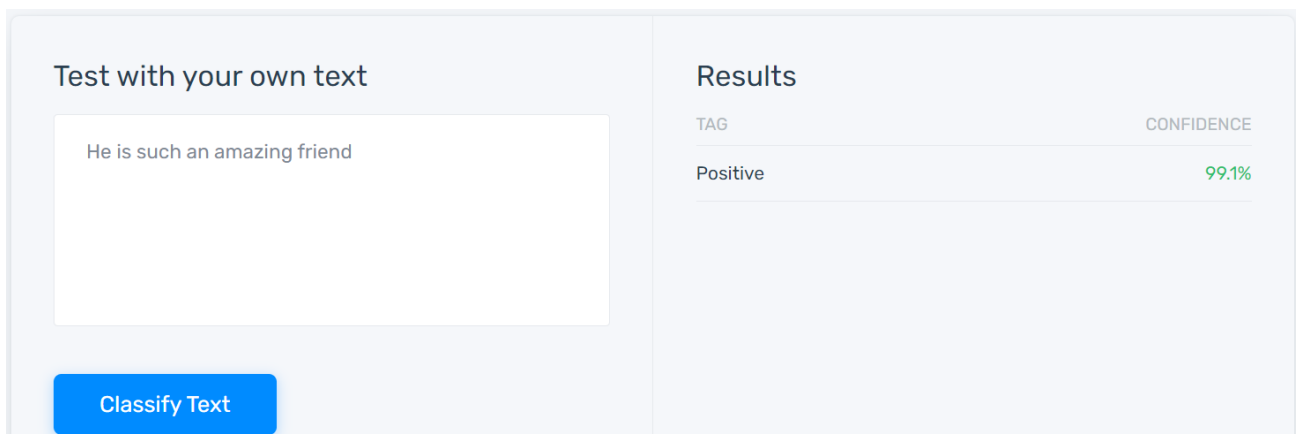


Рисунок 1.6 – Результати аналізу тексту сайтом «MonkeyLearn» [19]

Проаналізувавши текст, бачимо результат «Позитивний 99.1%», що відповідає дійсності. Спробуємо перевірити ще одне повідомлення «it was a terrible accident: the passengers were badly injured» (пер. це була жахлива подія: пасажери сильно постраждали) (рисунок 1.7).

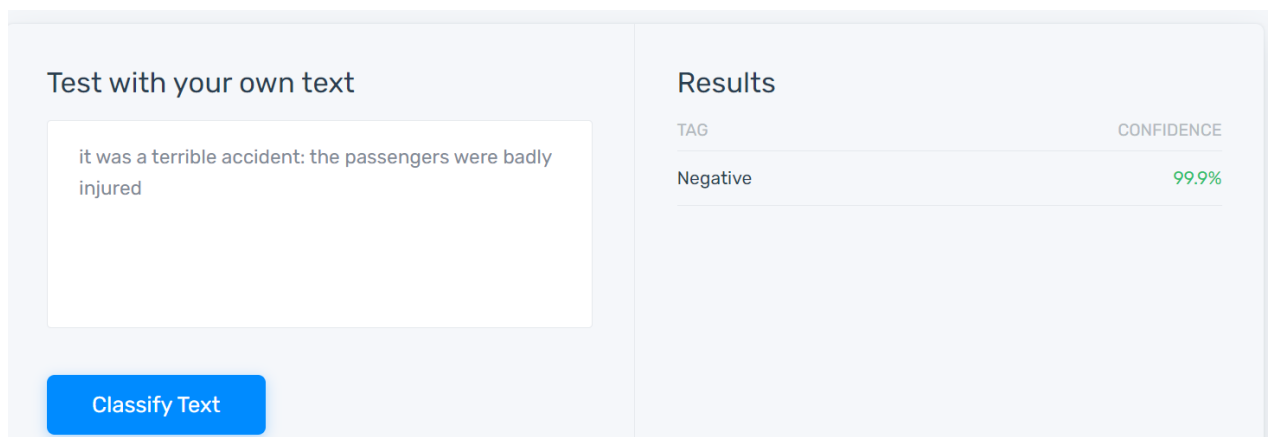


Рисунок 1.7 – Результати аналізу тексту сайтом «MonkeyLearn» [19]

Відповідь наступна: негативний 99.9%, що теж відповідає дійсності.

Недоліком цього сервісу є погана швидкість його роботи, час очікування на деякі повідомлення, особливо ті, що містять орфографічні помилки складає більше хвилини.

Як підсумок, проаналізовано популярні системи семантичного аналізу тексту. Було встановлено, що в кожного з сайтів є свої недоліки й переваги, що обов'язково варто врахувати під час реалізації власного методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками та створенні програмного продукту.

#### **1.4 Аналіз сучасних засобів створення програмного забезпечення**

Для створення інформаційної системи інтелектуальної асоціації текстових повідомлень за семантичними ознаками було проаналізовано платформи .NET та Java та доступні фреймворки, що можна використати під час створення програмного продукту.

Microsoft .NET [20] – платформа, створена для спрощення та пришвидшення розробки програмних застосунків. За допомогою цієї платформи можна створювати безліч додатків, як і крос-платформенні, так і націлені на конкретну операційну систему.

Сфери застосування .NET:

- мікросервіси;
- машинне навчання;
- розробка ігор;
- програми для інтернету речей;
- хмарні застосунки;

Окрім цього, платформа надає можливість користувачам створювати додатки, призначені для серверних платформ Windows, Linus та MAC.

Розробка під платформу .NET користується великим попитом через свої численні переваги. Отож враховуючи що у рамках теми КРБ буде створення саме windows-застосування, у якості фреймворку буде використовуватись .NET Framework.

Java [21] – платформу обчислень, що була вперше випущена «Sun Microsystems» 1995 р. Сьогодні більшість веб-застосувань працюють з допомогою цієї платформи. Володіє високою швидкістю, надійним рівнем захисту.

Однією з найбільших переваг цієї платформи є її здатність легко переходити з однієї комп'ютерної системи на іншу та можливість запускати одну програму на багатьох різних системах.

Для реалізації методу визначення емотивної тональності тексту було обрано платформу .NET, адже вона володіє наступними перевагами, невластивими Java [22]:

- .NET, як платформа, створена спершу саме для роботи із Windows, тому найбільш сумісна поставленою задачею.
- Платформа дозволяє використовувати різні мови програмування: C#, VB.NET, C++).
- Легке налагодження проекту та налаштування під нові версії
- Застосування AJAX
- Зручні та потужні інструменти для створення інтерфейсу користувача
- Використання хмарних технологій Azure для інтернету речей та великих даних.

Зважаючи на вищеописані переваги та недоліки, було обрано платформу .NET в якості засобу розробки для реалізації програмного продукту.

Також необхідно проаналізувати доступні фреймворки, для створення застосунку. Було проведено порівняння таких фреймворків: .NET Core .NET Framework.

.NET Core [23] заснований на архітектурі .NET Framework. Проте між ними є ряд відмінностей:

- платформа має можливість використання хмарних технологій;
- функціонал для кроссплатформенного програмування та модульності;
- У версії Core середовища виконання бібліотеки відділене.

Зважаючи на те, що кожен компонент оновлюється окремо, це дозволяє програмі працювати з окремими модулями та не залежати від оновлення всієї платформи.

Доповнення було розроблено корпорацією Microsoft. Також важливою перевагою є те, що інструмент був і залишається безкоштовним для використання.

.NET Framework 4.5 [24] можна використовувати із мовами програмування C#, Visual Basic та F#, також користуючись перевагами розширеного функціоналу цих мов, що забезпечує цей фреймворк. Переваги використання .NET Framework 4.5 [25]:

- можливість спрощення створення асинхронного коду;
- забезпечення змішання потоку управління в синхронному коді;
- значна швидкість відгуку інтерфейсу користувача;
- поліпшена масштабованість веб-додатків.

.NET Framework 4.5 також реалізує суттєві вдосконалення в інших функціональних галузях, таких як ASP.NET, Managed Extensibility Framework, Windows Communication Foundation, Windows Workflow Foundation та Windows Identity Foundation. .NET Framework 4.5 забезпечує підвищення продуктивності, надійності та безпеки.

Зважаючи на поставлену задачу, було обрано платформу .NET та фреймворк .NET Framework 4.5 для реалізації програмного продукту методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками.

## 1.5 Мета, задачі та вимоги до реалізації інформаційної системи

Метою кваліфікаційної роботи бакалавра є розробка методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками, та його програмна реалізація на базі системи аналізу повідомлень на платформі .NET. Система аналізу повідомлень має виконувати ряд функцій, а саме:

- перегляд доступних повідомлень та текстових одиниць для аналізу;
- обрахунок оцінки семантичної важливості обраного унікального слова, що присутнє в обраному повідомленні;
- визначення множин слів текстового повідомлення, які містять семантичне забарвлення різних психоемоційних категорій;
- автоматизоване візуальне відображення психоемоційної тональності слів повідомлення;
- обрахунок числових оцінок психоемоційної тональності текстового повідомлення;
- автоматизоване візуальне відображення психоемоційної тональності кожного текстового повідомлення.

Для досягнення мети необхідно здійснити наступні етапи: характеризувати предметну область, розглянути теоретичні підходи для вирішення задач та існуючі програмні застосунки, спроектувати інформаційну систему, обрати засоби її розробки, програмно реалізувати усі необхідні модулі системи, здійснити їх тестування та валідувати створений програмний продукт.

## Розділ 2 Проектування інформаційної системи

### 2.1 Аналіз та автоматизація обробки потоків даних

#### 2.1.1 Методи автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками

Схему методу автоматизованого визначення психоемоційної тональності текстових повідомлень за семантичними ознаками наведено на рисунку 2.1.

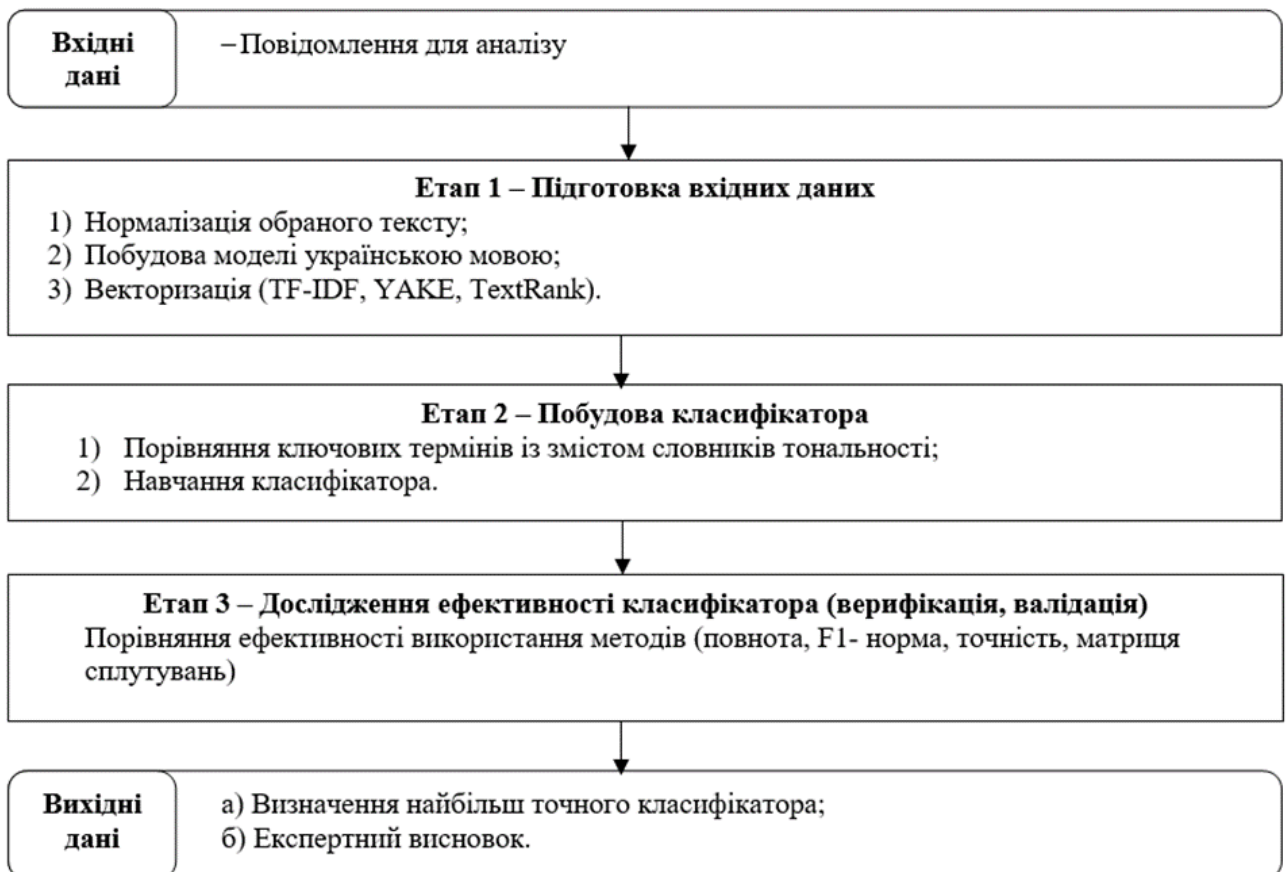


Рисунок 2.1 – Схема методу автоматизованого визначення психоемоційної тональності текстових повідомлень за семантичними ознаками

Робота методу починається із підготовки вхідних даних: було обрано повідомлення (текст), який необхідно проаналізувати.

Перший крок під час семантичного дослідження будь-якого тексту – це переведення його в цифрову форму, очищення від символів, чисел, зайвих пробілів та переведення в нижній регістр. Саме ці дії надають можливість комп'ютеру опрацювати інформацію друкованого, усного чи звукового матеріалу й проводити необхідні маніпуляції з ним.

Після того, як користувач обрав необхідний для подальшого аналізу текст, відбувається приведення тексту до «нормальної» форми тобто з повідомлення вилучаються пунктуаційні знаки, числа, різного роду символи, слова переводяться в нижній регістр. Після цього формуються послідовні множини слів, що відділені одне від одного.

Наступний крок – векторизація тексту. За допомогою методів TF-IDF, YAKE та TextRank відбираються ключові терміни повідомлення. За кожним методом окремо формуються множини ключових слів.

Після проходження вищеописаних етапів відбувається пошук емотивної лексики тих термінів, що були виокремлені методами пошуку ключових слів та обчислення певної оцінки повідомлення, що пізніше буде конвертована в рівень емотивного забарвлення у відсотках (позитивна, негативна, нейтральна).

Тобто, в сформованій множині ключових слів кожному терміну присвоюється певна оцінка (якщо слово присутнє в словнику тональності), а потім осилюється загальна тональність всього повідомлення, шляхом підсумування значення тональностей кожного окремого терміну.

Далі, із отриманих результатів формується таблиця приналежності повідомлення до кожної із емотивних категорій (позитивна, негативна, нейтральна) та формується експертний висновок. Окрім того, передбачається можливість перегляду оцінок кожного ключового слова у повідомленні.

Таким чином, було представлено схему роботи методу визначення психоемоційного забарвлення у текстових повідомленнях.

## **2.1.2 Математично-алгоритмічне забезпечення**

В роботі використовуватиметься декілька методів пошуку ключових термінів, тому необхідно детально описати кожен з них.

TF-IDF [26] – метод оцінки, що складається із частотної оцінки TF[27] (term frequency) (2.2), та оберненої частоти IDF[27] (inverse document frequency) (2.3). Оцінка є обчисленим значенням частоти появи певного слова ( $x$ ) у множині повідомлень, також показує, наскільки слово є вартісним для окремого повідомлення. Кожне із значень обчислюється наступним чином:

$$TF_x = \frac{n(x)}{N}, \quad (2.1)$$

де  $n(x)$  – кількість появ слова  $x$  у множині повідомлень,  $N$  – загальна кількість слів у множині повідомлень.

$$IDF_x = \log \frac{|D|}{|(d_i \supset x_i)|}, \quad (2.2)$$

де  $|D|$  – кількість множин повідомлень,  $|(d_i \supset x_i)|$  – кількість множин повідомлень, де зустрічається слово  $x_i$ , за умови, що  $i \neq 0$ .

Значення TF-IDF (2.3), що є результатом роботи методу метод є добутком двох вищеописаних значень – TF та IDF.

$$TF - IDF_x = TF_x \cdot IDF_x, \quad (2.3)$$

де  $TF_x$  – частота певного слова множині повідомлень, а  $IDF_x$  – обернена частота певного слова у множині повідомлень.

Наступний метод – YAKE [28] полягає в наступних кроках:

1. Спершу текст нормалізується та генеруються терміни-кандидати [28]. Проводиться розділення на рівні речення, що далі будуть розбиватись на терміни.

2. Визначаються особливості термінів. Цей етап полягає в тому, що кожний окремий термін оцінюється за допомогою спеціальних функцій.

3. Підрахунок балів за кожен окремий термін. Обчислення проводиться за формулою 2.4 [28].

$$S(w) = \frac{W_{unique} \cdot W_{pos}}{W_{regist}} + \left( \frac{W_{frequency}}{W_{unique}} + \frac{W_{sent}}{W_{unique}} \right), \quad (2.4)$$

де  $W_{regist}$  – важливість великих літер і скорочень,  $W_{pos}$  – позиція слова в документі: більше значення надається словам, що розташовані на початку документа,  $W_{frequency}$  – частота появи слів,  $W_{unique}$  – перевіряє унікальність та різноманітність контексту, у якому зустрічається певне слово,  $W_{sent}$  – визначається, як часто певне слово (слово-кандидат) зустрічається в різних реченнях. Вищий бал отримують ті слова, що часто зустрічаються в інших реченнях.

4. Останній етап – об'єднання значень оцінок подібних морфологічно слів. Що менша оцінка – то більша вірогідність підібрати краще ключове слово.

Таким чином, було досліджено методи аналізу тексту та виділення ключових термінів з повідомлень.

Для реалізації методу автоматизованого визначення психоемоційної тональності текстових повідомлень за семантичними ознаками, та реалізації програмного продукту із функціями, описаними в розділі 1.5 будуть використані формули та методи, описані вище.

## **2.2 Інформаційна структура системи**

### **2.2.1 Проектна архітектура системи та взаємозв'язок компонентів**

Система автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками складається із трьох підсистем та бази даних (рисунок 2.2).

Першою є підсистема, що необхідна для роботи із базою даних, у якій зберігаються текст для подальшого аналізу та інформація про них. Ця підсистема виконує такі функції для роботи із БД як: вивід даних з БД, їх редагування та видалення. Окрім того, БД зберігатиме і проміжні результати для подальшої роботи із текстовими повідомленнями.

Друга підсистема призначена для формування множин ключових термінів, зокрема, виконує наступні функції:

- очищення тексту від знаків пунктуації, символів та чисел;
- формування множини ключових термінів за методами TF-IDF, YAKE та TextRank (векторизація);
- передача даних в БД та заповнення відповідних таблиць.



Рисунок 2.2 – Структура компонентів автоматизованої системи визначення психоемоційної тональності текстових повідомлень

Третя підсистема призначена для визначення психоемоційного забарвлення тексту, виконуються наступні функції:

- пошук збігів слів із сформованої множини ключових термінів та словників тональності;
- присвоєння емотивних оцінок ключовим термінам текстового повідомлення;

- обчислення результатів та формування оцінки психоемоційного забарвлення повідомлення за семантичними ознаками;
- формування експертного висновку та виведення результатів користувачеві.

Таким чином, представлена структура компонентів методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками виконує поставлене завдання.

### **2.2.2 Інформаційна модель**

Робота застосунку не може бути повноцінною без належно створеної бази даних, що не лише зберігатиме вхідні дані, а й проміжні результати обчислень. Необхідно створити структуру інформаційної системи, що забезпечить надійні зв'язки між даними та безперебійну роботу програми.

База даних – це організована структура, призначення якої є збереження, редагування та обробка взаємозалежної інформації [29]. Для розуміння структури БД було створено діаграму даних, що відображає список таблиць, їх полів та зв'язків між ними.

Відповідно до поставленого завдання, а саме створення методу визначення психоемоційної тональності текстових повідомлень за семантичними ознаками, було спроектовано структуру бази даних, що дозволяє зберігати матеріал для аналізу, а саме цифрові текстові повідомлення, та проводити ефективний обрахунок результатів аналізу. Структура розробленої БД представлена на рисунку 2.3.

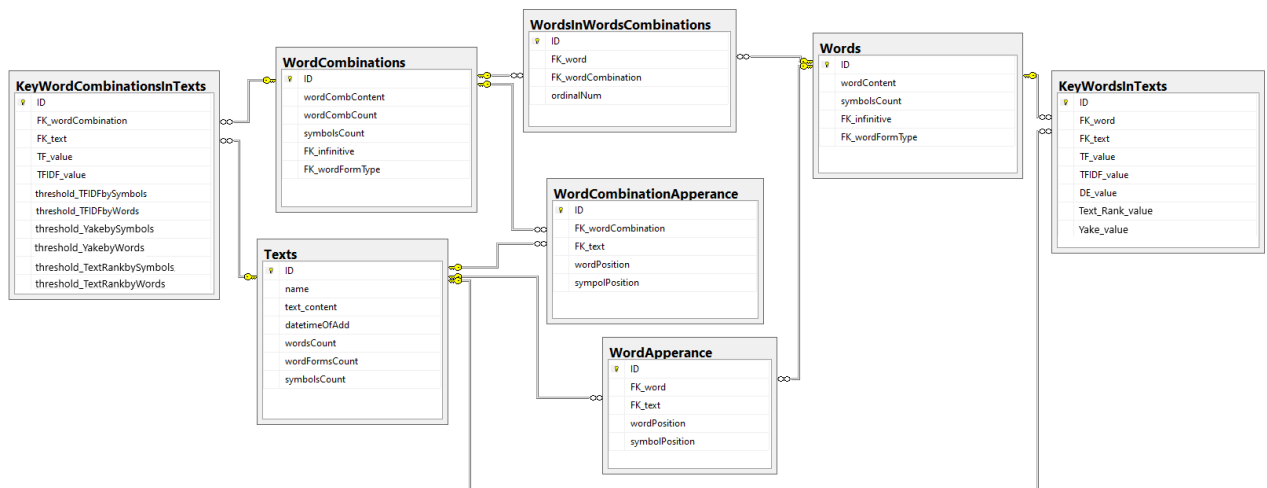


Рисунок 2.3 – Структура бази даних інформаційної системи інтелектуальної асоціації текстових повідомлень за семантичними ознаками

Згідно із поставленим завданням, необхідно реалізувати таблиці бази даних, що відповідає заданій структурі та заповнити їх початковими даними для подальшого аналізу.

Таблиця «Texts» (таблиця 2.1) зберігає цифрові текстові повідомлення, що будуть проаналізовані системою. Таблиця містить наступні поля: ID, name, textContent, dateTime, wordsCount, symbCount.

Таблиця 2.1 – Атрибути таблиці «Texts»

№П/П	Назва атрибуту	Тип даних	Опис
1	ID	int	Цілочисельний ідентифікатор записів, первинний ключ для визначення запису таблиці.
2	name	nvarchar(50)	Назва повідомлення.
3	textContent	nvarchar(MAX)	Контент, наповнення повідомлення.
4	dateTime	datetime	Дата й час додання повідомлення до БД
5	wordsCount	int	Кількість слів у текстовому повідомленні.
6	symbCount	int	Кількість символів у текстовому матеріалі

Таблиця «Words» (таблиця 2.2) призначена для зберігання інформації про слово, що зустрічається в тексті. Кожне слово тут зберігається як незалежна семантична одиниця, визначається його словоформа, інфінітив та кількість символів, що містить слово.

Таблиця 2.2 – Атрибути таблиці «Words»

№П/П	Назва стовпця	Тип даних	Призначення
1	ID	int	Цілочисельний ідентифікатор записів, первинний ключ для визначення запису таблиці.
2	wordContent	varchar(50)	Текст слова.
3	symbolsCount	int	Кількість символів, що містить слово.
4	FK_infininit	int	Вторинний ключ, посилання на запис таблиці «Infinitives» для співставлення з відповідним записом в таблиці для визначення інфінітиву слова.
5	FK_wFormType	int	Вторинний ключ, посилання на запис «FormTypes» для співставленням із типом слова

Таблиця «WordCombs» (таблиця 2.3) створена для збереження деталей щодо словосполучень. Таблиця містить наступні поля: ID, wCombContent та wCombCount.

Таблиця 2.3 – Атрибути таблиці «WordCombs»

№П/П	Назва стовпця	Тип даних	Призначення
1	ID	int	Цілочисельний ідентифікатор записів, первинний ключ для визначення запису таблиці.
2	wCombContent	varchar(50)	Зміст словосполучення.
3	wCombCount	int	Числове значення кількості зустрічей словосполучення в тексті.
4	symbolsCount	int	Кількість символів у словосполученні, що розглядається.

5	FK_infinite	int	Вторинний ключ, посилання на таблицю «Infinite» для співставлення з відповідним інфінітивом слова.
6	FK_wordFormType	int	Вторинний ключ, посилання на запис таблиці «WFormTypes» для співставлення з відповідною словоформою

Таблиця «WordsInCombinations» (таблиця 2.4) зберігає дані щодо слів, котрі складають певне словосполучення. Містить наступні поля: дані про слова з яких складається словосполучення та його порядковий номер у текстовому повідомленні.

Таблиця 2.4 – Атрибути таблиці «WordsInCombinations»

№П/П	Назва стовпця	Тип даних	Призначення
1	ID	int	Цілочисельний ідентифікатор записів, первинний ключ для визначення запису таблиці.
2	FK_word	int	Вторинний ключ, посилання на таблицю «Words» для співставлення з відповідним словом в словосполученні.
3	FK_wordComb	int	Вторинний ключ, посилання на запис «WordCombinations» для співставлення з відповідним словосполученням тексту.
4	wNumber	int	Порядковий номер словосполучення у тексті

Таблиця «WordCombApperance» (таблиця 2.5) зберігає дані щодо появи словосполучень в тексті. Таблиця містить поля: ID, FK\_wordComb, FK\_text, wordPos, symbolPos.

Таблиця 2.5 – Атрибути таблиці «WordCombApperance»

№П/П	Назва стовпця	Тип даних	Призначення
1	ID	int	Цілочисельний ідентифікатор записів, первинний ключ для визначення запису таблиці.

2	FK_wordComb	int	Вторинний ключ, посилання на запис «WordCombs» для співставлення із відповідним словосполученням, що зустрічається в тексті.
3	FK_text	int	Вторинний ключ, посилання на запис із таблиці «Texts» для співставлення із відповідним досліджуваним текстовим повідомленням.
4	wordPos	int	Значення положення слова в тексті по словах.
5	sympolPos	int	Значення положення слова в тексті по символах.

Таблиця «WordApperance» (таблиця 2.6) створена для збереження даних щодо появи слів у тексті, таблиця містить поля для збереження назви повідомлення, що досліджується, змісту слів, їх позицію.

Таблиця 2.6 – Атрибути таблиці «WordApperance»

№П/П	Назва стовпця	Тип даних	Призначення
1	ID	int	Цілочисельний ідентифікатор записів, первинний ключ для визначення запису таблиці.
2	FK_word	int	Вторинний ключ, посилання на таблицю «Words» для співставлення з відповідним словом в словосполученні.
3	FK_text	int	Вторинний ключ, посилання на запис із таблиці «Texts» для співставлення із відповідним досліджуваним текстовим повідомленням.
4	wordPosition	int	Позиція слова у повідомленні по словах
5	sympolPosition	int	Позиція слова у повідомленні по символах

Таблиця «KeyWordsInMessages» (таблиця 2.7) створена для зберігання даних щодо ключових термінів та значень проміжних математичних обчислень, що здійснюються під час аналізу повідомлення.

Таблиця 2.7 – Атрибути таблиці «KeyWordsInMessages»

№П/П	Назва стовпця	Тип даних	Призначення
1	ID	int	Цілочисельний ідентифікатор записів, первинний ключ для визначення запису таблиці.
2	FK_word	int	Вторинний ключ, посилання на таблицю «Words» для співставлення з відповідним словом в словосполученні.
3	FK_text	int	Вторинний ключ, посилання на запис із таблиці «Texts» для співставлення із відповідним досліджуваним текстовим повідомленням.
4	TFIDF_value	float	Обчислене значення TF-IDF для ключових термінів повідомлення.
5	threshold_TFIDFbySymbols	float	Порогове значення входження слова слова за TF-IDF по символах.
6	threshold_TFIDFbyWords	float	Порогове значення входження слова за TF-IDF по словах
7	threshold_YakebySymbols	float	Порогове значення входження слова слова YAKE по символах.
8	threshold_YakebyWords	float	Порогове значення входження слова за YAKE по словах
9	threshold_TextRankbySymbols	float	Порогове значення входження слова слова за TextRank по символах.
10	threshold_TextRankbyWords	float	Порогове значення входження слова за TextRank по словах

Таблиця «KeyWordCombsInTexts» (таблиця 2.8) зберігає дані щодо ключових словосполучень в повідомленнях та проміжні обчислення під час аналізу тексту.

Таблиця 2.8 – Атрибути таблиці «KeyWordCombinationsInTexts»

№П/П	Назва стовпця	Тип даних	Призначення
1	ID	int	Цілочисельний ідентифікатор

			записів, первинний ключ для визначення запису таблиці.
2	FK_wordComb	int	Вторинний ключ, посилання на запис «WordCombs» для співставлення з відповідним словосполученням повідомлення, що досліджується.
3	FK_text	int	Вторинний ключ, посилання на запис із таблиці «Texts» для співставлення із відповідним досліджуваним текстовим повідомленням.
5	threshold_TFIDFbySymbols	float	Порогове значення входження слова слова за TF-IDF по символах.
6	threshold_TFIDFbyWords	float	Порогове значення входження слова за TF-IDF по словах
7	threshold_YakebySymbols	float	Порогове значення входження слова слова YAKE по символах.
8	threshold_YakebyWords	float	Порогове значення входження слова за YAKE по словах
9	threshold_TextRankbySymbols	float	Порогове значення входження слова слова за TextRank по символах.
10	threshold_TextRankbyWords	float	Порогове значення входження слова за TextRank по словах
11	threshold_TFIDFbySymbols	float	Порогове значення входження слова слова за TF-IDF по символах.
12	threshold_TFIDFbyWords	float	Порогове значення входження слова за TF-IDF по словах
13	threshold_YakebySymbols	float	Порогове значення входження слова слова YAKE по символах.

В результаті виконання розділу було створено структуру бази даних та відповідні таблиці. Створена БД дозволяє зберігати усю необхідну інформацію щодо повідомлень, їх змісту та результатів аналізу, потрібну для створення технології за методом визначення психоемоційної тональності тексту за семантичними ознаками.

## 2.3 Вибір засобів розробки інформаційної системи

Запорукою створення програмного продукту, що правильно виконуватиме поставлені задачі є доцільно підібрані засоби розробки інформаційної системи.

Метод автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками передбачає постійний зв'язок між БД та застосунком. Тому, необхідно визначити, із якою мовою програмування буде здійснюватиметься робота, СКБД, фреймворк та редактор програмного коду.

### 2.3.1 Вибір мови програмування

Для створення додатку методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками було проведено порівняння C# та Java.

C# – об'єктно-орієнтована мова програмування [30], створена у 2000 році компанією Microsoft. Здобула популярність через свою властивість гнучкості та кросплатформенності. Мову можна використати при створенні додатків будь-якого типу призначення. Важливою особливістю цієї мови програмування є її статична типізація.

Окрім того, існує ряд переваг цієї мови програмування, невластивих іншим мовам [31]:

- легкість прочитання та розуміння;
- підтримка ООП;
- статична типізація;
- мова підтримує поліморфізм;
- створені компоненти можуть використовуватись повторно;

Мова С# буде безпосередньо розпізнана будь-яким, хто добре знайомий із С, С++ або Java. Розробники, що знайомі з будь-якою із цих мов, зазвичай можуть почати ефективно працювати на С # за менший час.

Окрім цього, мова С# має ряд додаткових функцій [32]

- автоматичний збір сміття;
- універсальна проста у використанні;
- стандартна бібліотека;
- монтажні версії;
- керування делегатами та подіями.

Також було розглянуто переваги та недоліки мови програмування Java. Java [33] є однією з найпоширеніших і найефективніших мов програмування протягом тривалого часу. Причина в тому, що вона має багато переваг, які допомагають програмістам легко вирішувати складні реальні проблеми. Java також допомагала в розробці різного програмного забезпечення в минулому і продовжує допомагати в розробці в даний час. Однак варто зауважити, що мова також має багато мінус, що було досліджено:

- низька продуктивність через значне споживання оперативної пам'яті;
- несучасний графічний інтерфейс та застарілі конструктори форм;
- нульові можливості для резервного копіювання даних користувачів;
- складний, в порівнянні з С# синтаксис;
- високовартісна комерційна ліцензія.

Проаналізувавши недоліки та переваги описаних мов програмування, для написання програми методу визначення психоемоційного забарвлення текстових повідомлень за семантичними ознаками була обрана мова програмування С#.

### **2.3.2 Вибір фреймворку**

Для подальшої роботи було обрано фреймворк .NET Framework 4.5. вибір обґрунтовано рядом переваг, що має цей засіб розробки, а саме:

- заснований на принципах ООП;

- відмінно налагоджена система кешування
- середовище розробки програмного коду visual studio, що ідеально сумісний з .NET;
- доступність кросплатформної розробки;
- легкість в розгортанні та обслуговуванні.

Для реалізації програмного продукту методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками було обрано фреймворк.NET Framework 4.5.

### **2.3.3 Вибір редактору програмного коду**

Не менш важливим кроком під час створення програмних продуктів є вибір середовища написання програмного коду, адже розробник більшу частину часу проводить саме у ньому. Отже якість коду певною мірою залежить від зручності редактору та його потужності.

Було розглянуто два найпопулярніших редактори на 2022 рік: Visual Studio та Eclipse IDE.

Eclipse [34] – модульне інтегроване середовище розробки програмного забезпечення. Підтримка та створення оновлень забезпечується Eclipse Foundation і включає проекти, такі як платформа Eclipse, набір інструментів для програмістів на мові Java, системи контролю версій, конструктори GUI тощо. Написаний в основному на Java, може бути використаний для розробки застосунків на Java та на інших мовах програмування, включаючи Ada, C, C++, C#, та інші. Переважно, цей редактор коду використовують для написання Java-скриптів.

Однак середовище має ряд мінусів:

- складний провідник проєктів;
- тривале налаштування при першому встановленні;
- використовує значний запас оперативної пам'яті.

MS Visual Studio [35] – продукт Майкрософт, що має інтегроване середовище розробки програмного забезпечення та низку інших засобів. Дає змогу розробляти як консольні програми, так і програми з графічним інтерфейсом, включно з підтримкою технології Windows Forms, а також вебсайти, вебзастосунки, вебслужби що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Переваги VS:

- висока гнучкість та продуктивність роботи;
- наявність інструментарію для розробки, орієнтованої на хмарні сервіси;
- наявність безкоштовної підписки;
- інтуїтивний інтерфейс.

Ці два середовища пропонують безліч функцій, які гарантують їх зручність використання. Однак Visual Studio має перевагу в будь-якій з параметрів оцінки.

Тому, для написання програмного коду методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками було обрано середовище розробки MS Visual Studio 2017.

#### **2.3.4 Вибір СКБД**

Окрім цього необхідно обрати систему керування базами даних. Для справної роботи інформаційної системи необхідно забезпечити коректний зв'язок між базою та додатком. Тому, для надійної роботи методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками було проаналізовано наступні СКБД: MS SQL Server та MySQL.

MySQL – досить популярна та потужна СКБД SQL з відкритим вихідним кодом, що підтримується корпорацією Oracle [36]. Застосунок часто використовують у веб-розробці, однак СКБД має ряд недоліків:

- неефективно підтримує бази даних великого обсягу;
- не передбачена можливість підтримки ROLE, COMMIT;
- транзакції не виконуються так швидко та ефективно, як у MS SQL Server;
- часті збої роботи;
- гірше масштабування продуктивності, ніж у MS SQL Server.

MS SQL Server [37] – це система управління реляційними базами даних, розроблена компанією Microsoft. Включає мову SQL і T-SQL, можливість оголошення змінних і збережених процедур. Механізм бази даних поділено на два сегменти:

- реляційний механізм, який використовується для обробки команд та запитів;
- механізм зберігання, призначений керувати різними функціями бази даних, як-от таблиці, сторінки, файли, індекси і транзакції.

#### Переваги SQL Server:

- Висока продуктивність роботи: через виклики процедур не виникає затримок, оскільки в SQL Server передбачена можливість одноразової компіляції процедур й збереження їх у формі, що виконується.
- Масштабування: якщо необхідно внести певні зміни, необхідно лише замінити збережену процедуру, не змінюючи код програми.
- Надійність: якщо необхідно обмежити доступ до даних, можна написати збережену процедуру для доступу до даних і дозволити цьому користувачеві використовувати лише доступні для нього таблиці.

Зважаючи на предметну область кваліфікаційної роботи та на вимоги до СКБД, а також враховуючи усі вищеписані недоліки та переваги, було прийнято рішення використовувати SQL Server у якості СКБД.

Підсумовуючи вищесказане, в цьому розділі було проаналізовано варіанти СКБД, мов програмування та фреймворків, доступних для реалізації програмного продукту.

Для розробки інформаційної технології інтелектуальної асоціації текстових повідомлень за семантичними ознаками було обрано платформу .NET, мову програмування C#, СКБД Microsoft SQL Server та середовище розробки Visual Studio 2017.

## Розділ 3 Програмна реалізація інформаційної системи

### 3.1 Структура та функціональне призначення програмних складових системи

Для структуризації системи VTN JL було створено діаграму класів, зображену нижче, на рисунку 3.1.

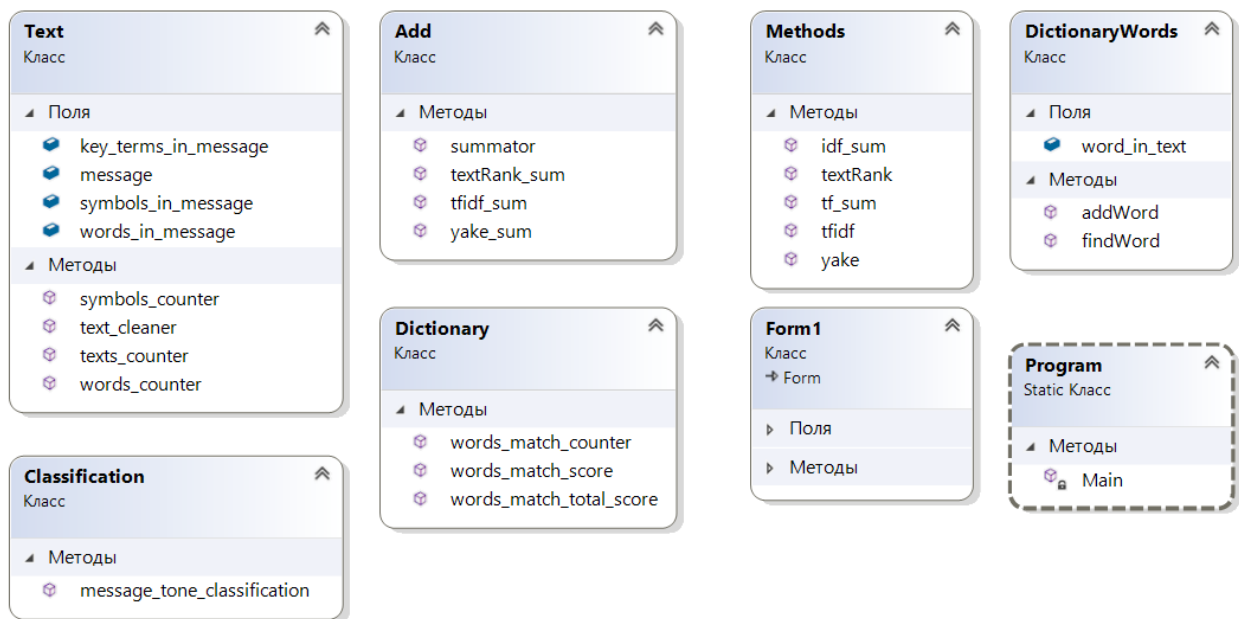


Рисунок 3.1 – Діаграма класів програмного застосунку на базі методу визначення тональності повідомлення за семантичними ознаками

Робота програми розпочинається із класу Program. На Form1 передбачень різні вкладки для роботи із текстовими повідомленням та їх подальшим аналізом, та є основним графічним інтерфейсом, з яким працюватиме користувач. Реалізовано наступні вкладки:

- База повідомлень.
- Появи семантичних одиниць.
- Аналіз повідомлень.

Клас “Text” містить методи, що допомагають обробити текст для подальшої векторизації, тобто нормалізують його. Метод text\_cleaner

призначений для видалення чисел, символів із тексту та його переведення у нижній регістр. Метод `texts_counter` призначений для обчислення кількості доступних текстів. Метод `words_counter` створений для обчислення кількості слів у кожному окремому повідомленні. Метод `symbols_counter` – обчислює кількість символів в повідомленні. Таким чином, цей клас – є початковим для роботи нашої системи, адже саме з нього розпочинається робота: завантаження текстів та їх попередня обробка.

Клас “Methods” призначений для роботи методів, за якими будуть визначатись ключові терміни. Метод `tf_sum` обчислює значення TF в тексті, `idf_sum` – відповідно, IDF. В результаті обчислення цих двох значень, метод `tf_idf` повертає значення TF-IDF. Метод `yake` та `textRank` визначають вектор ключових термінів відповідно. Таким чином, в результаті роботи цих методів відбувається векторизація текстового повідомлення.

Клас “Dictionary” містить у собі ряд методів, що необхідні під час роботи із словниками тональності. Методи `words_match_counter`, `words_match_score` та `words_match_total_score` спершу шукають збіги між сформованою множиною ключових слів та словником, привласнюють оцінки кожному ключовому слову згідно із словником тональності та підсумовують результати відповідно. Таким чином, кожному повідомленню буде присвоєно певну оцінку.

Заключний етап – формування експертного висновку. В цьому і полягає задача наступного методу – “Classification”. Метод `message_tone_classification` конвертує оцінку, що повертає метод `words_match_total_score` класу “Dictionary” в відсоткову приналежність повідомлення до певної категорії: позитивна, нейтральна, негативна та на основі неї формує експертний висновок.

На рисунку 3.2 зображено перший модуль системи, що призначений для початкової роботи із БД. Саме тут користувач може обрати текстове повідомлення, яке необхідно проаналізувати чи додати нове. Зокрема, повідомлення можна переглянути перед запуском аналізу, відредагувати його чи видалити, якщо потрібно. Також, в цьому модулі можна одразу переглянути нормалізований текст повідомлення.

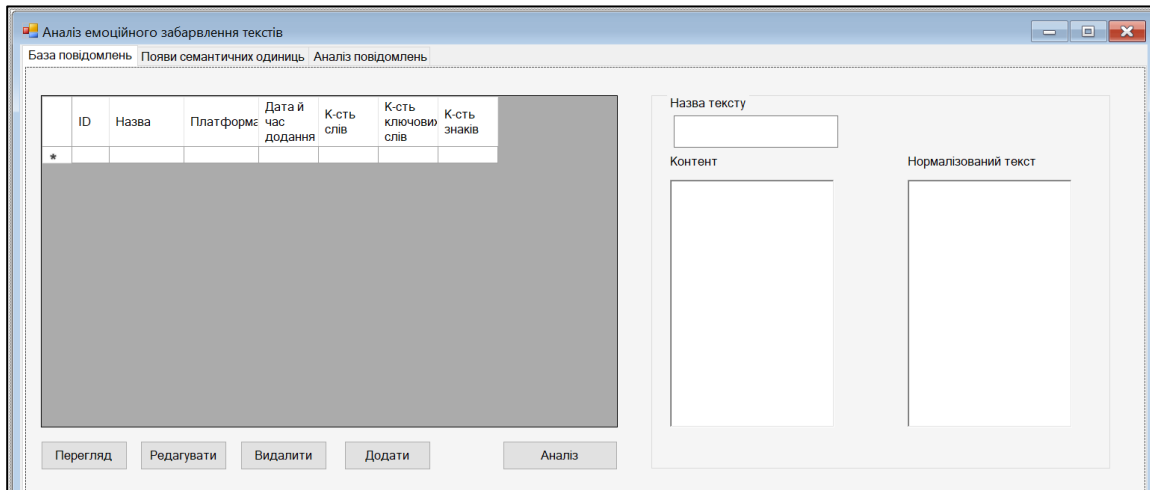


Рисунок 3.2 – Інтерфейс модулю програмного застосунку, призначеного для роботи з текстовими повідомленнями

На рисунку 3.3 представлено форму, що виводить користувачеві основну інформацію по ключових термінах та відображає наступну інформацію:

- Появи слів тексту. Користувач може переглянути «нормалізоване» повідомлення та усю інформацію по появі слів у ньому, вивести повідомлення на екран.
- Появи ключових термінів тексту. Користувач може переглянути списки ключових термінів та вивести їх на екран.

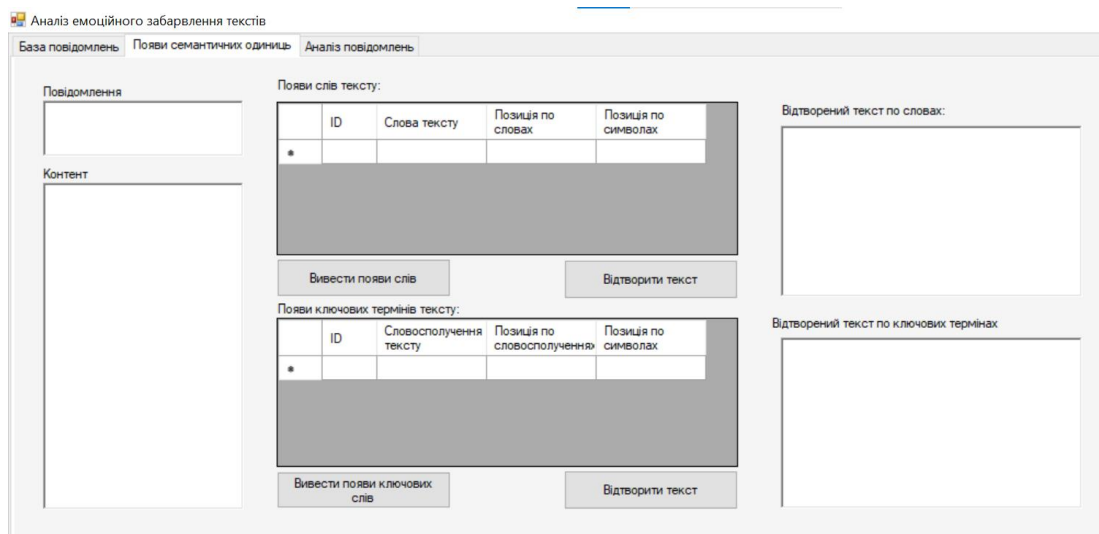


Рисунок 3.3 – Інтерфейс модуля застосунку, призначеного для відображення інформації про появи слів у тексті

На рисунку 3.4 представлено модуль системи, що відповідає за відображення головної інформації – приналежності повідомлення до певної емотивної категорії та формування експертного висновку щодо нього.

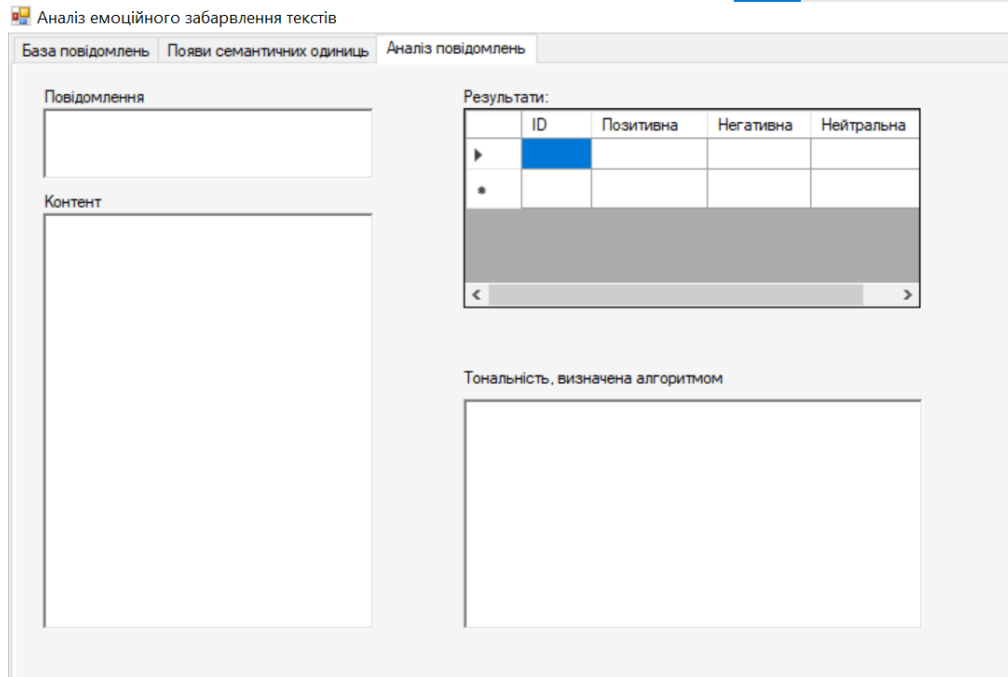


Рисунок 3.4 – Інтерфейс модулю аналізу повідомлень системи

Таким чином, в результаті розробки методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками було отримано вищеописані модулі інтерфейсу, з якими працюватиме користувач та класи програмного продукту, що реалізують необхідні функції.

### 3.2 Особливості реалізації програмних складових системи

Перший модуль програмного коду, із якого розпочинає свою роботу система – завантаження бази текстових повідомлень в додаток, тому було реалізовано метод «LoadTextBase» для виведення даних на форму застосунку. В окрему таблицю типу «dataGridView» виводяться наступні поля таблиці:

- ID
- назва повідомлення;
- платформа, на якій було розміщено чи отримане повідомлення;
- дата й час додання в БД;
- кількість слів, що містить повідомлення;
- кількість символів в повідомленні;

Фрагмент програмного коду наведено нижче:

```
public void loadTextBase()
{
    string connectString = "Server=.\SQLEXPRESS;Initial Catalog=nlp;Integrated
Security=true;";
    SqlConnection myConnection = new SqlConnection(connectString);
    myConnection.Open();
    string query = "SELECT * FROM TextBase ORDER BY ID";
    SqlCommand command = new SqlCommand(query, myConnection);
    SqlDataReader reader = command.ExecuteReader();
    List<string[]> dataa = new List<string[]>();
    while (reader.Read())
    {
        dataa.Add(new string[10]);
        dataa[dataa.Count - 1][0] = reader[0].ToString();
        dataa[dataa.Count - 1][1] = reader[1].ToString();
    }
    reader.Close();
    myConnection.Close();
    foreach (string[] s in dataa)
        dataGridView1.Rows.Add(s);
}
```

Результат виконання програмного коду можна побачити на рисунку 3.5

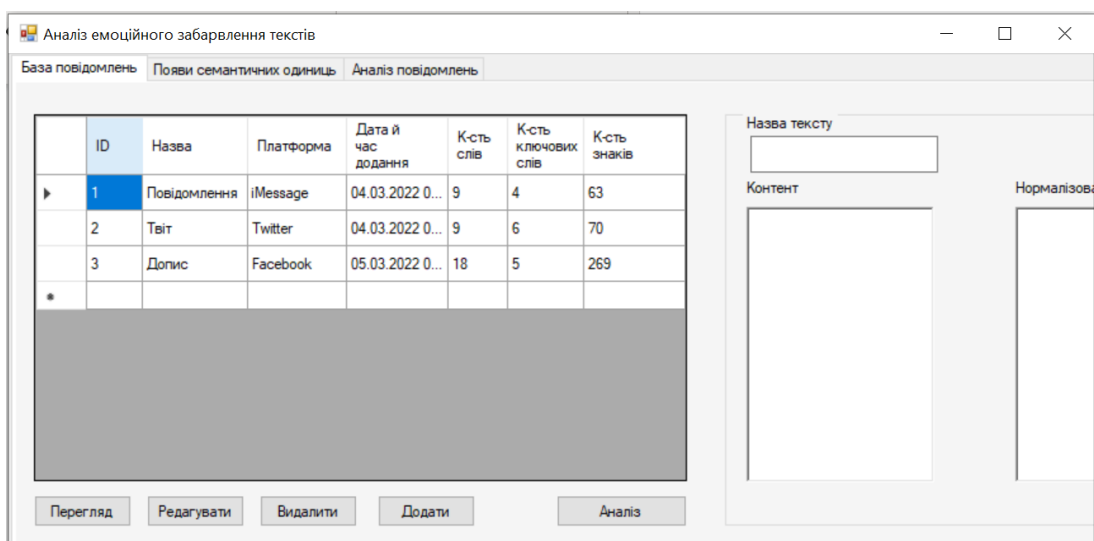


Рисунок 3.5 – Результат роботи методу «LoadTextBase»

Для того, щоб система могла працювати із текстом, необхідно нормалізувати текст. Фрагмент програмного коду наведено нижче:

```
var str = project.Variables[].Value.ToLower();
if (str.Length > 1)
{
    return char.ToUpper(str[0]) + str.Substring(1);
}
return str.ToUpper();
```

В результаті виконання програмного коду (рисунок 3.6) отримуємо текст, що переведений в нижній регістр та не має в собі знаків, тощо.

Аналіз емоційного забарвлення текстів

База повідомлень | Появи семантичних одиниць | Аналіз повідомлень

ID	Назва	Платформа	Дата й час додання	К-сть слів	К-сть ключових слів
1	Повідомлення	iMessage	04.03.2022 0...	9	4
2	Твіт	Twitter	04.03.2022 0...	9	6
3	Допис	Facebook	05.03.2022 0...	18	5

Назва тексту: Допис

Контент:

Друзі, сьогодні о 15 години проведемо зустріч в нашому молодіжному центрі. Тема "Як захистити себе від негативного впливу інтернету". Приводьте друзів та знайомих, будемо раді. Орієнтовний час півтори години. Дякуємо за лайк та поширення на ваших сторінках. До зустрічі!

Нормалізований текст:

друзі сьогодні о години проведемо зустріч в нашому молодіжному центрі тема як захистити себе від негативного впливу інтернету приводьте друзів та знайомих будемо раді орієнтовний час півтори години дякуємо за лайк та поширення на ваших сторінках до зустрічі

Перегляд | Редагувати | Видалити | Додати | Аналіз

Рисунок 3.6 – Результат виконання методу нормалізації тексту

Після нормалізації тексту слідує крок його векторизації. Наведемо приклад роботи методу TF-IDF, що визначає ключові терміни текстів. Фрагмент наступного програмного коду – реалізація методу TF-IDF.

```

private static double[][] TransformToTFIDFVectors(List<List<string>> stemmedDocs,
Dictionary<string, double> vocabularyIDF)
{
    // Transform each document into a vector of tfidf values.
    List<List<double>> vectors = new List<List<double>>();
    foreach (var doc in stemmedDocs)
    {
        List<double> vector = new List<double>();

        foreach (var vocab in vocabularyIDF)
        {
            // Term frequency = count how many times the term appears in this document.
            double tf = doc.Where(d => d == vocab.Key).Count();
            double tfidf = tf * vocab.Value;

            vector.Add(tfidf);
        }

        vectors.Add(vector);
    }

    return vectors.Select(v => v.ToArray()).ToArray();
}

```

В результаті виконання програмного коду, отримуємо ключові терміни повідомлення, що відобразяться на наступному модулі застосунку (рисунок 3.7).

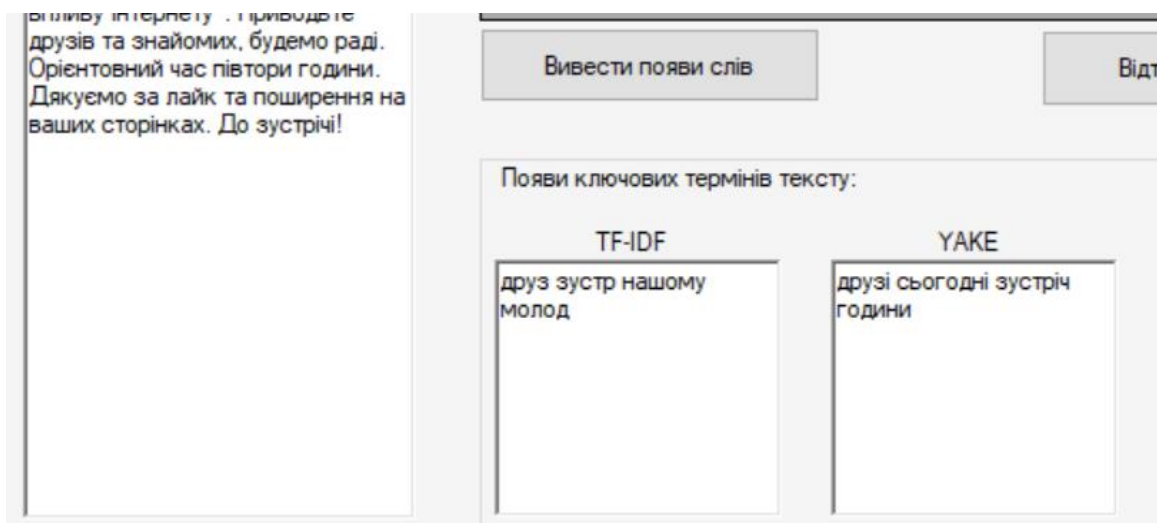


Рисунок 3.7 – Результат виконання методу «tf\_idf»

Таким чином, в результаті розробки програмного застосунку на базі методу визначення психоемоційної тональності тексту за семантичними ознаками було реалізовано програмний продукт, що виконує всі поставлені вимоги, має зрозумілий інтерфейс та високу швидкодію роботи.

### 3.3 Тестування інформаційної системи

Важливий крок – верифікація програмного продукту, тобто необхідно пересвідчитись, що ми створили програмний продукт правильно, реалізували повідомлення для користувача у випадках некоректно ведених даних, тощо.

Для перевірки правильності роботи програмного застосунку методу визначення психоемоційної тональності тексту за семантичними ознаками було проведено ряд тест кейсів та UI тестування системи.

Було проведено перший тест-кейс (таблиця 3.1), що перевіряє попередження системи про невведений текст в БД.

Нехай користувач при завантаженні в базу даних повідомлення, не ввів його контент. При виборі такого повідомлення необхідно вивести відповідне попередження.

Таблиця 3.1 – Тест-кейс АТ0001

<b>Тест-кейс ID:</b> АТ0001	<b>Пріоритет:</b> 2	<b>Створено:</b> 12.05.2022, Залуцька О.
<b>Назва:</b> Перевірка повідомлення про вибір тексту без введеного до БД контенту. <b>Вхідні дані:</b> Текстове повідомлення .txt		
<b>Кроки</b>	<b>Очікуваний результат</b>	
<p><i>Передумова:</i> на вкладці «База повідомлень» користувач має обрати рядок текстового повідомлення, якому не введений контент та натиснути кнопку «Аналіз».</p> <ol style="list-style-type: none"> <li>Запустити додаток та обрати вкладку «База повідомлень»</li> <li>Обрати рядок повідомлення із відсутнім контентом в БД (рядок 4)</li> <li>Натиснути кнопку «Аналіз»</li> <li>Порівняти фактичний результат з очікуваним</li> <li>Закрити діалог із помилкою</li> <li>Обрати рядок 3, де всі поля заповнені коректно</li> <li>Натиснути кнопку «Аналіз»</li> <li>Порівняти фактичний результат з очікуваним</li> </ol>	<p>Діалогове вікно із помилкою: «Відсутній текст для подальшого аналізу!».</p> <p>Рядок обрано, текст успішно проаналізовано.</p>	

**Результат виконання тест-кейсу: пройдено успішно**

Результат виконання тест-кейсу наведено на рисунку 3.8.

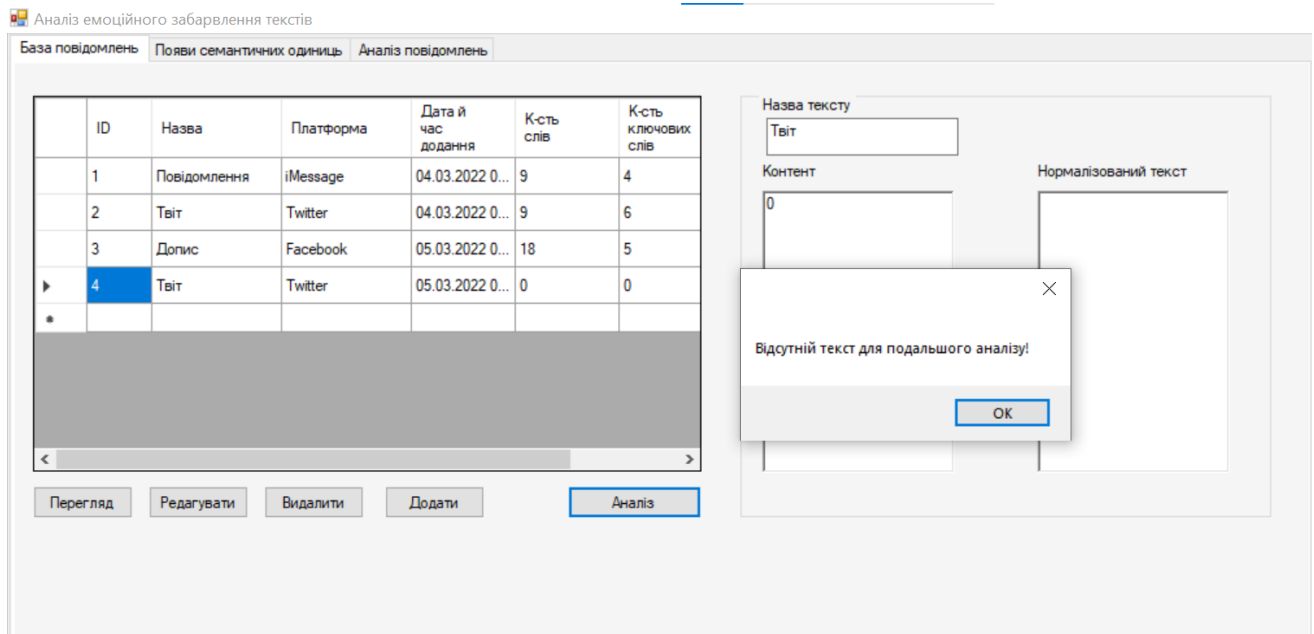


Рисунок 3.8 – Результат виконання тест-кейсу АТ0001

Було проведено тестування підсистеми, що нормалізує текст, видаляє числа, символи, зайві пробіли з тексту та переводить його в нижній регістр. Тест-кейс наведено нижче (таблиця 3.2).

Таблиця 3.2 – Тест-кейс АТ0002

<b>Тест-кейс ID:</b> АТ0002	<b>Пріоритет:</b> 1	<b>Створено:</b> 12.05.2022, Залуцька О.
<b>Назва:</b> Перевірка правильності методу нормалізації тексту		
<b>Вхідні дані:</b> Текстове повідомлення .txt		
<b>Кроки</b>	<b>Очікуваний результат</b>	
<p><i>Передумова:</i> на вкладці «База повідомлень» користувач має обрати рядок текстового повідомлення, подальший аналіз якого потрібно перевести, зокрема перевірити якісь роботи методу нормалізації тексту.</p> <ol style="list-style-type: none"> <li>Запустити додаток та обрати вкладку «База повідомлень»</li> <li>Обрати рядок повідомлення із</li> </ol>	Текст «тестування рядка»	

<p>відсутнім контентом в БД (рядок 5 із текстом «ТесТуванНя Рядка 1234 %:.*( ,...»)</p> <p>3. Натиснути кнопку «Аналіз»</p> <p>4. Порівняти фактичний результат з очікуваним</p>	Текст «тестування рядка»
<b>Результат виконання тест-кейсу:</b> пройдено успішно	

Результат проведення тест-кейсу наведено на рисунку 3.9.

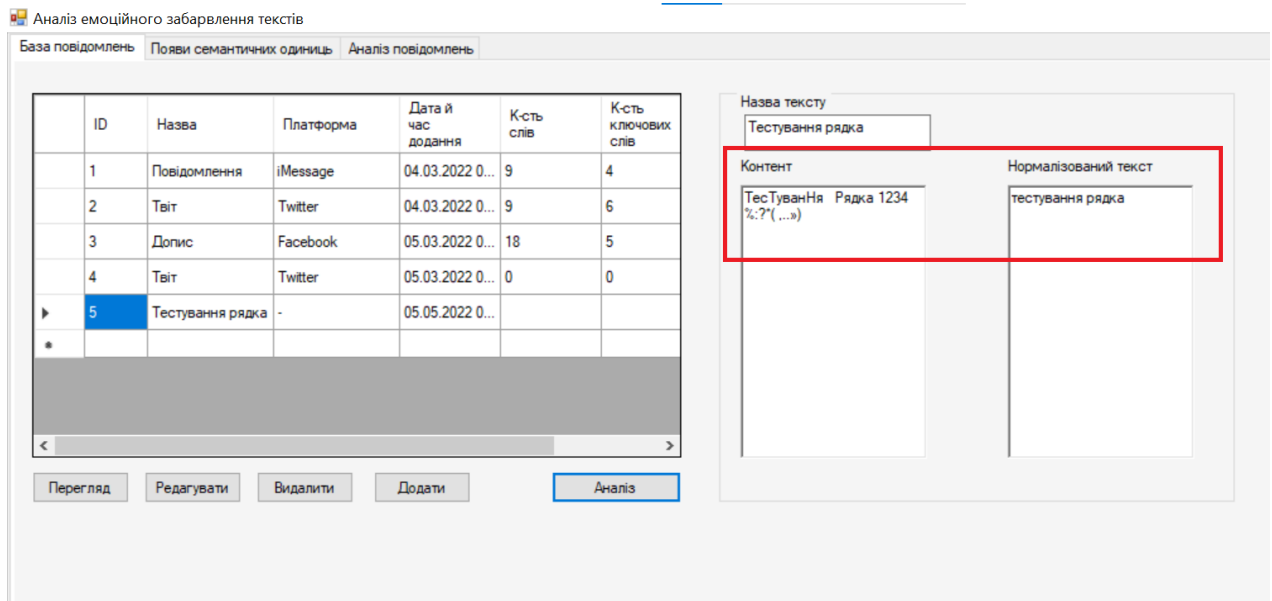


Рисунок 3.9 – Результат перевірки роботи методу нормалізації тексту

Часто користувачі можуть не звертати увагу на вид тексту, що подається для аналізу, тому варто передбачити і випадки, коли текст введений в базу некоректно, попередивши про це користувача.

Було проведено відповідний тест-кейс (таблиця 3.3).

Таблиця 3.2 – Тест-кейс АТ0003

<b>Тест-кейс ID:</b> АТ0003	<b>Пріоритет:</b> 2	<b>Створено:</b> 12.05.2022, Залуцька О.
<b>Назва:</b> Перевірка правильності завантаження тексту із БД		
<b>Вхідні дані:</b> Текстове повідомлення .txt		
<b>Кроки</b>	<b>Очікуваний результат</b>	
<i>Передумова:</i> на вкладці «База повідомлень» користувач має обрати		

<p>рядок текстового повідомлення, подальший аналіз якого неможливий через некоректний вид запису в БД.</p> <ol style="list-style-type: none"> <li>1. Запустити додаток та обрати вкладку «База повідомлень»</li> <li>2. Обрати рядок повідомлення із некоректно введеним текстом повідомлення, наприклад, двійковим кодом (рядок 5 із текстом «001110101010101010101010»)</li> <li>3. Натиснути кнопку «Аналіз»</li> <li>4. Порівняти фактичний результат з очікуваним</li> </ol>	<p>Поява діалогового вікна із текстом «Неможливо зчитати текст»</p> <p>Поява діалогового вікна із текстом «Неможливо зчитати текст»</p>
<p><b>Результат виконання тест-кейсу: пройдено успішно</b></p>	

Результати виконання тест-кейсу наведено на рисунку 3.10.

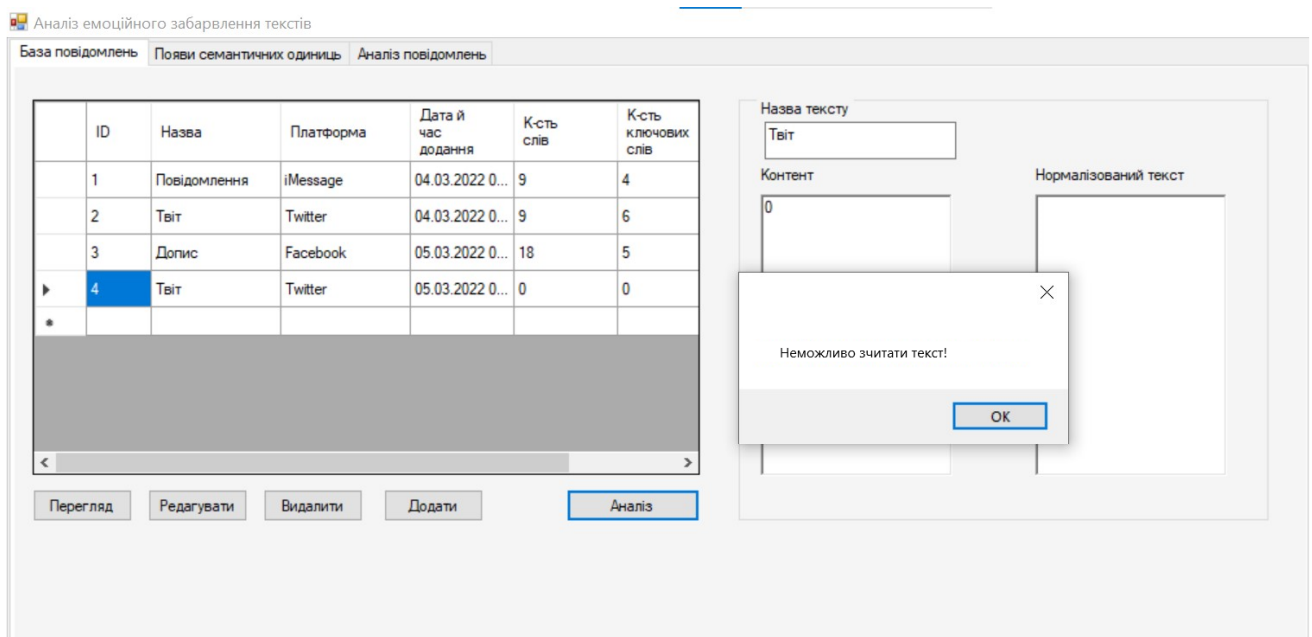


Рисунок 3.10 – Результат виконання тест-кейсу

Також було виконано тестування системи шляхом внесення в базу даних повідомлення негативного емотивного забарвлення. Відповідно до поставленого завдання сформовано тест-кейс (таблиця 3.11)

Таблиця 3.2 – Тест-кейс АТ0004

Тест-кейс ID: АТ0004	Пріоритет: 1	Створено: 12.05.2022, Залуцька
----------------------	--------------	--------------------------------

		О.
<b>Назва: Перевірка правильності методу визначення психоемоційного забарвлення</b>		
<b>Вхідні дані:</b> Текстове повідомлення .txt		
<b>Кроки</b>	<b>Очікуваний результат</b>	
<p><i>Передумова:</i> на вкладці «База повідомлень» користувач має обрати рядок текстового повідомлення, подальший аналіз якого відбуватиметься. Повідомлення містить явно негативний характер, що необхідно довести.</p> <ol style="list-style-type: none"> <li>1. Запустити додаток та обрати вкладку «База повідомлень»</li> <li>2. Обрати рядок текстового повідомлення з негативним емоційним забарвленням.</li> <li>3. Натиснути кнопку «Аналіз»</li> <li>4. Порівняти фактичний результат з очікуваним</li> </ol>	<p>Система має довести, що повідомлення містить негативне забарвлення із вірогідністю, більшою за 80%.</p> <p>За результатами аналізу повідомлення із вірогідністю 97.4% містить негативне забарвлення.</p>	
<b>Результат виконання тест-кейсу:</b> пройдено успішно		

Результат виконання тест-кейсу наведено на рисунку нижче (рисунк 3.11).

Аналіз емоційного забарвлення текстів

База повідомлень | Появи семантичних одиниць | Аналіз повідомлень

Повідомлення  
Відгук

Контент  
Гальмівний жахливо. Може сам вимкнутись. Відбиток пальця взагалі може не спрацьовувати, а про ігри можна не говорити. Це викинуті на вітер гроші. Купував на офіційному сайті, але більше не кочу і дивитись в бік цього виробника, неякісний, погана збірка.

Результати:

	ID	Позитивна	Негативна	Нейтральна
▶	1	2.6	97.4	0
*				

Тональність, визначена алгоритмом

Назва повідомлення: "Відгук"  
Платформа, на якій розміщено "Відгук" - Rozetka

За результатами дослідження, цей текст із вірогідністю 97.4% містить негативне забарвлення

Із вірогідністю 2.6 є позитивним

Рисунок 3.11 – Результат виконання тест-кейсу

Окрім тест-кейсів проведено UI тестування (рисунок 3.12), задля визначення потенційних місць, де може виникнути помилка.

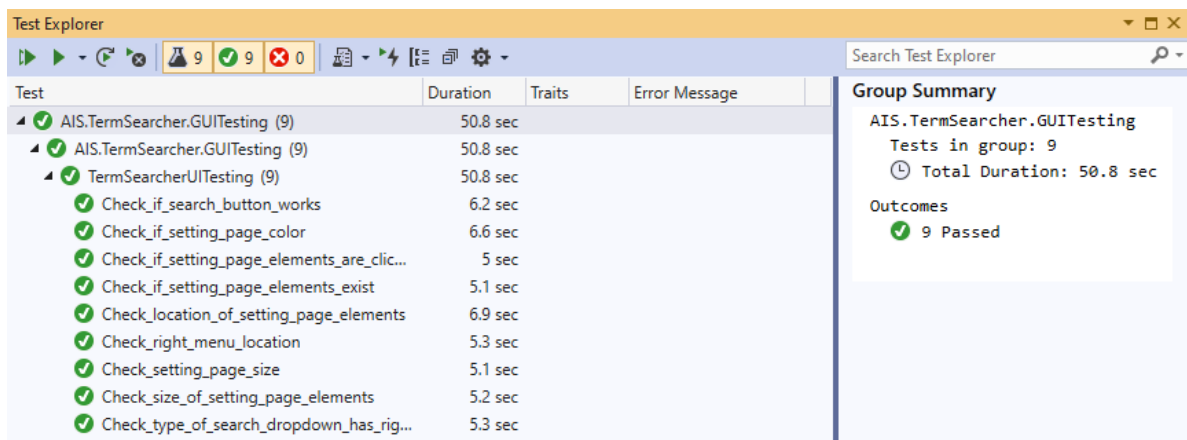


Рисунок 3.12 – Результати проведення UI тестування

Як висновок, було проведено ряд тестувань задля верифікації системи. На прикладі тест-кейсів було продемонстровано правильність роботи основних функцій, а UI тестування засвідчує, що інтерфейс користувача є стабільним та надійним. Для забезпечення ефективності роботи застосунку тестування проводилось після кожної зміни в програмному коді.

### 3.4 Інструкція користувача

Робота користувача із методом визначення психоемоційної тональності тексту за семантичними ознаками розпочинається з першого модуля «База повідомлень», в якому можна виконати наступні дії:

- обрати текстове повідомлення для подальшого аналізу для перегляду;
- редагувати існуюче текстове повідомлення;
- видалити існуюче текстове повідомлення.

Для вибору повідомлення для подальшого аналізу необхідно обрати рядок із відповідним записом та натиснути кнопку «Аналіз» (рисунок 3.13).

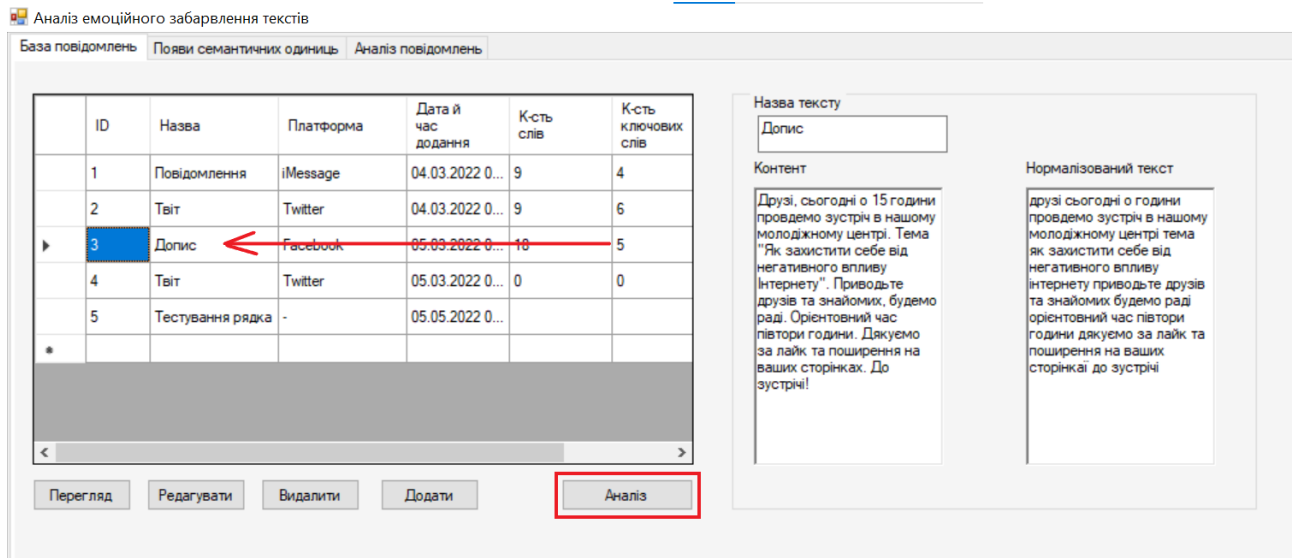


Рисунок 3.13 – Робота із вибором текстового повідомлення для аналізу

На цій формі також можна переглянути весь текст повідомлення та його вигляд у нормалізованій формі (рисунок 3.14).

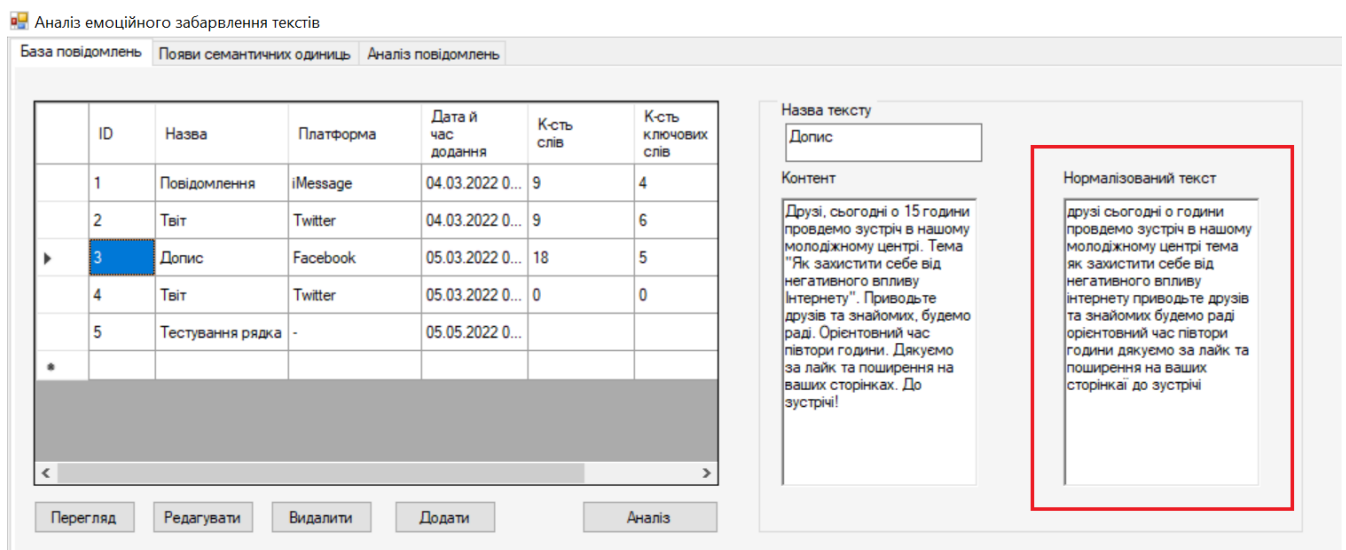


Рисунок 3.14 – Перегляд тексту, приведеного до нормальної форми

Обравши повідомлення, користувач переходить до наступної вкладки – «Появи семантичних одиниць». На цій вкладці спочатку виводиться лише назва та контент повідомлення (рисунок 3.15).

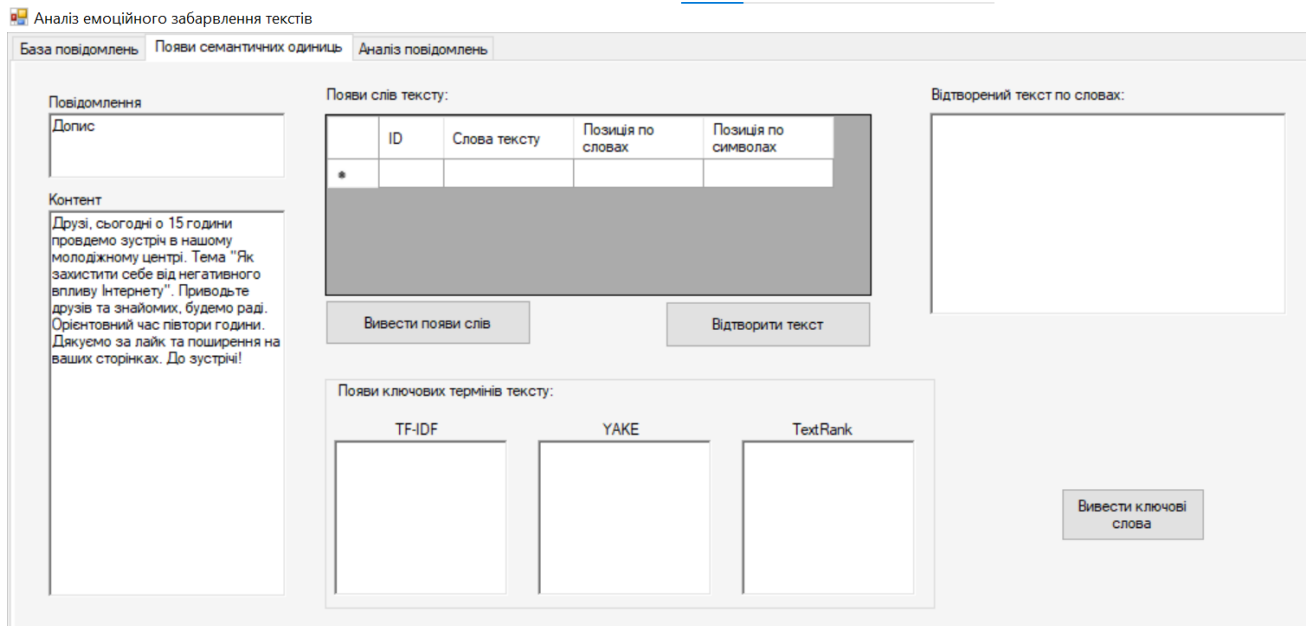


Рисунок 3.15 – Початок роботи із вкладкою «Появи семантичних одиниць»

Залежно від мети роботи користувач обирає із запропонованих дій: «Вивести появи слів», «Відтворити текст» та «Вивести ключові слова».

Якщо користувач обирає «Вивести появи слів» - з'явиться таблиця із словами, що містить повідомлення та інформація про них (рисунок 3.16).

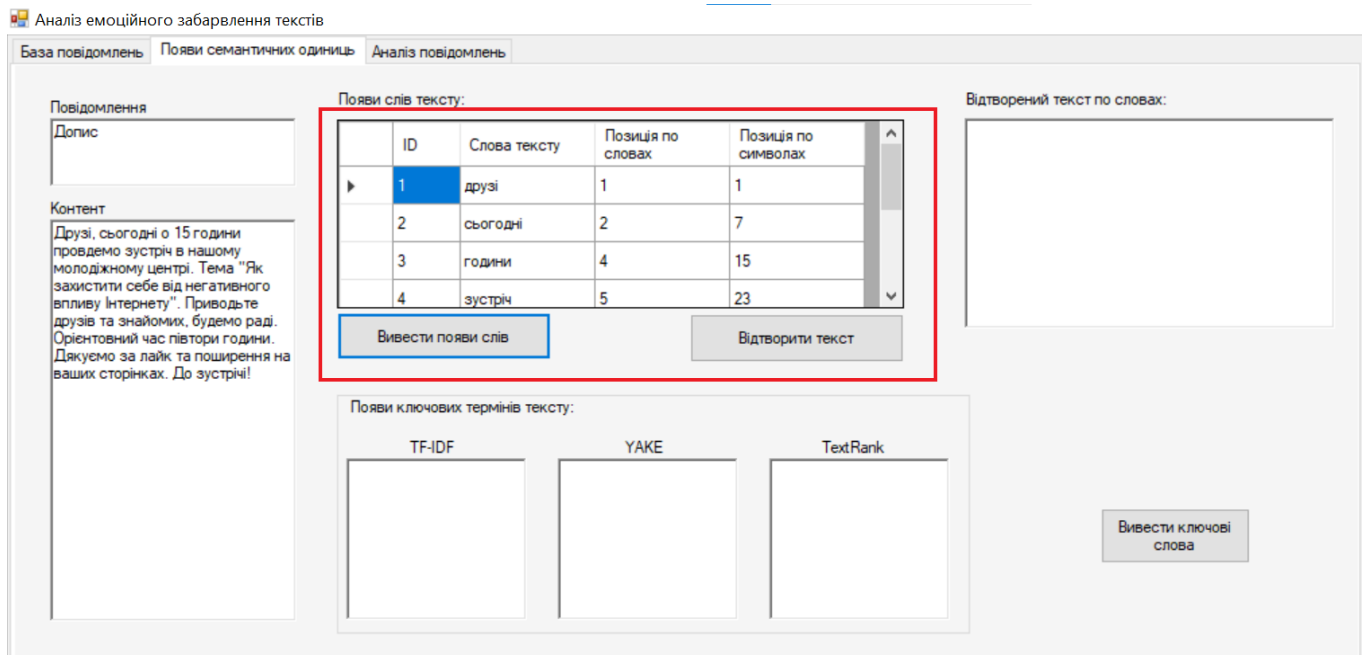


Рисунок 3.16 – Результат роботи методу «Вивести появи слів»

Також, можна відтворити текст по словах, натиснувши кнопку «Відтворити текст» (рисунок 3.17).

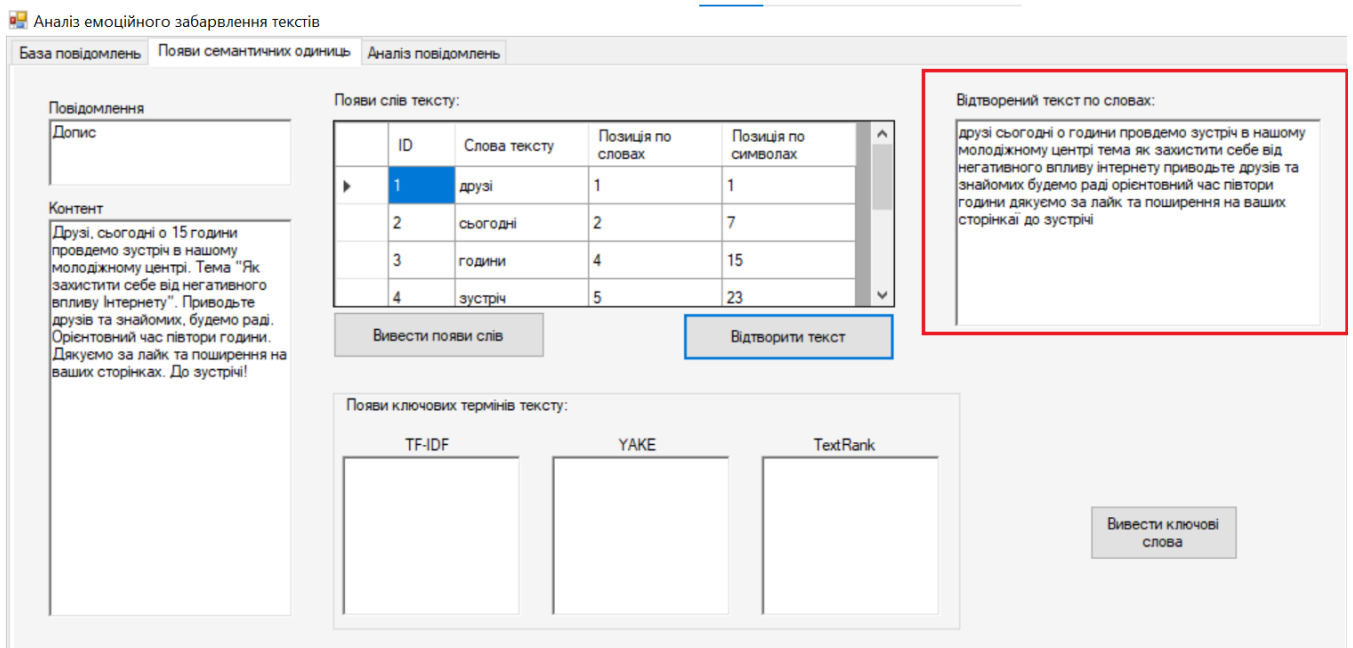


Рисунок 3.17 – Результат роботи методу для відтворення тексту повідомлення по словах

Також, існує можливість одразу переглянути ключові слова, визначені наступними методами: TF-IDF, YAKE, TextRank. Для цього користувачеві необхідно натиснути кнопку «Вивести ключові слова» (рисунок 3.18).

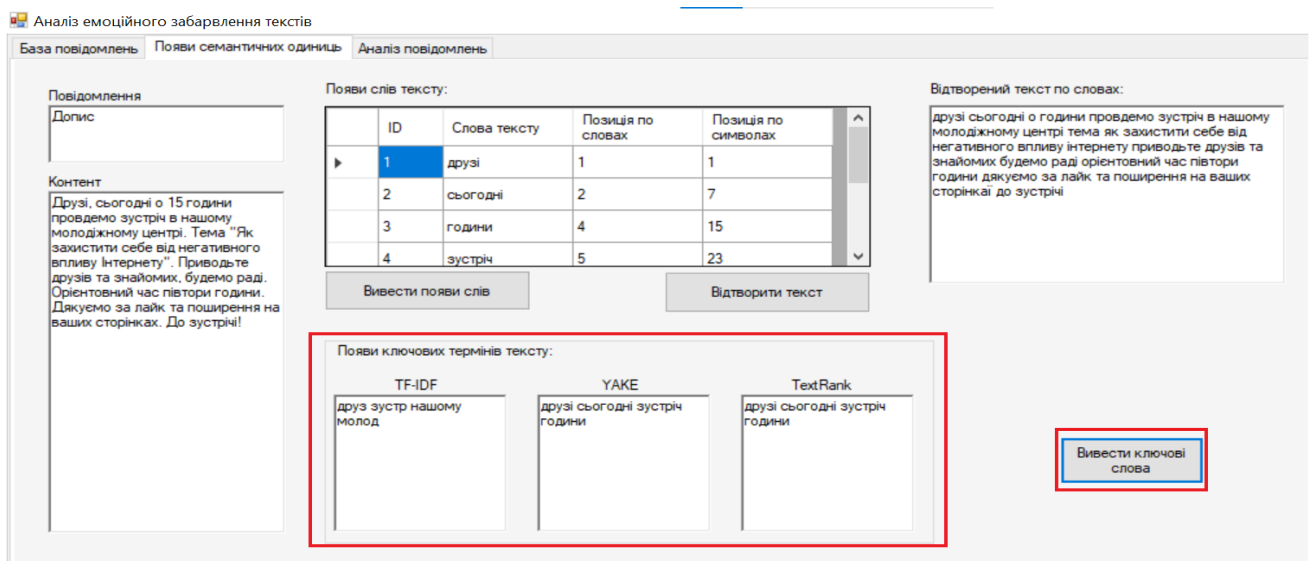


Рисунок 3.18 – Результат виведення ключових термінів на екран

Далі користувач переходить до вкладки «Аналіз повідомлень», що є головним у реалізованому застосунку. На цій вкладці виконуються наступні функції:

- повторний перегляд повідомлення;
- результати аналізу емоційного забарвлення за категоріями (позитивний, негативний, нейтральний);
- висновок щодо емоційного забарвлення тексту.

Результати на цій вкладці генеруються автоматично після натиснення кнопки «Аналіз» на вкладці «База повідомлень».

Користувачеві представлено до уваги таблицю із даними щодо аналізу, відсоткові значення приналежності текстового повідомлення до кожної з емотивних категорій.

Користувач може переглянути результати аналізу у наступній таблиці (рисунок 3.19).

Аналіз емоційного забарвлення текстів

База повідомлень | Появи семантичних одиниць | **Аналіз повідомлень**

Повідомлення  
Допис

Контент  
Друзі, сьогодні о 15 години проведемо зустріч в нашому молодіжному центрі. Тема "Як захистити себе від негативного впливу Інтернету". Приводьте друзів та знайомих, будемо раді. Орієнтовний час півтори години. Дякуємо за лайк та поширення на ваших сторінках. До зустрічі!

Результати:

ID	Позитивна	Негативна	Нейтральна
1	95.2 %	3.16 %	1.64 %
*			

Тональність, визначена алгоритмом

Назва повідомлення: "Допис"  
Платформа, на якій розміщено "Допис" - Facebook

За результати дослідження, цей текст із вірогідністю 95.2 % містить позитивне забарвлення  
Із вірогідністю 3.16 % є негативним

Рисунок 3.19 – Результати аналізу тексту у табличній формі

Також, сформовано висновок щодо повідомлення. Результати наведено на рисунку 3.20.

Аналіз емоційного забарвлення текстів

База повідомлень    Появи семантичних одиниць    Аналіз повідомлень

Повідомлення

Допис

Контент

Друзі, сьогодні о 15 години проведемо зустріч в нашому молодіжному центрі. Тема "Як захистити себе від негативного впливу Інтернету". Приводьте друзів та знайомих, будемо раді. Орієнтовний час півтори години. Дякуємо за лайк та поширення на ваших сторінках. До зустрічі!

Результати:

ID	Позитивна	Негативна	Нейтральна
1	95.2 %	3.16 %	1.64 %

Тональність, визначена алгоритмом

Назва повідомлення: "Допис"  
Платформа, на якій розміщено "Допис" - Facebook

За результати дослідження, цей текст із вірогідністю 95.2 % містить позитивне забарвлення

Із вірогідністю 3.16 % є негативним

Рисунок 3.20 – Результати аналізу повідомлення, відображені на вкладці «Аналіз повідомлень»

Таким чином було продемонстровано, що інтерфейс користувача є зрозумілим та простим у використанні. Сформовано інструкцію користувача.

### 3.5 Вимоги до розгортання інформаційної системи

Задля коректної роботи методу визначення психоемоційної тональності тексту за семантичними ознаками рекомендовано нижченаведені технічні засоби.

Мінімальні вимоги до апаратних засобів:

- Процесор: AMD Atlon 640 DualCore.
- RAM: 4 Gb.
- Вільний дисковий простір: 128 Gb.

Рекомендовані вимоги до апаратних засобів:

- Процесор: Intel Core I7 8<sup>th</sup> Gen;
- RAM: 8 Gb;
- Вільний дисковий простір: 128 Gb.

Вимоги до програмних засобів:

- ОС: Windows 7/10.
- СКБД MS SQL Server (версія від 2012 року).
- .NET Framework 4.5.

## Висновки

В результаті виконання кваліфікаційної роботи бакалавра, було реалізовано метод автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками та його програмна реалізація на базі системи аналізу повідомлень на платформі .NET.

Система виконує усі необхідні функції, а саме:

- перегляд доступних повідомлень та текстових одиниць для аналізу;
- обрахунок оцінки семантичної важливості обраного унікального слова, що присутнє в обраному повідомленні;
- визначення множин слів текстового повідомлення, які містять семантичне забарвлення різних психоемоційних категорій;
- автоматизоване візуальне відображення психоемоційної тональності слів повідомлення;
- обрахунок числових оцінок психоемоційної тональності текстового повідомлення;
- автоматизоване візуальне відображення психоемоційної тональності кожного текстового повідомлення.

Робота повністю відповідає поставленому завданню, було перевірено роботу системи низкою тестувань.

Програмний продукт можна використати в установах, мета роботи яких – визначення громадської думки, дослідження коментарів, відгуків, тощо. Звісно, програмний продукт можна вдосконалювати: забезпечити можливість автоматизованого визначення тексту на сторінці та його аналіз, додати більшу кількість методів пошуку ключових термінів, під'єднати новіші, потужніші словники для пошуку емотивної лексики.

Також для досягнення мети було проведено: аналіз та характеристика предметної області, дослідження існуючих теоретичних підходів та програмного

забезпечення для вирішення задачі визначення емотивного забарвлення текстів, зокрема програмна реалізація власного продукту.

Основні наукові та практичні результати доповідались в наступних наукових публікаціях:

– Залуцька О. О. Інформаційний портрет ключових термінів у цифрових навчальних матеріалах, О. О. Залуцька, О. В. Мазурець. Матеріали III Міжнародної науково-практичної конференції «Сучасні інформаційні технології та інноваційні методики навчання: досвід, тенденції, перспективи», Тернопіль, 2019, С. 120-122.

– Параметри моделі тестового завдання при автоматизованому формуванні тестів, Ю. Р. Придачук, О. О. Залуцька, Я. О. Збірник наукових праць за матеріалами XII Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2020». Хмельницький, 2020, с. 229-231.

– Войчишин О. О., Залуцька О. О., Попов Ю. М., Купрійчук В. О. Інформаційна технологія автоматизованого формування семантичного ядра цифрових текстів. Збірник наукових праць за матеріалами XIII Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2021». Хмельницький, 2021. с. 298-305.

## Перелік посилань

1. Крак Ю. В., Бармак О. В., Мазурець О. В. – Практична реалізація інформаційної технології автоматизованого визначення множини семантичних термінів в контенті навчальних матеріалів. Проблеми програмування, 2018, 2-3: 245-254.
2. Nicolai C., Piazza M. – The implicit commitment of arithmetical theories and its semantic core. Erkenntnis. 2019. Т. 84. №. 4. С. 913-937
3. Lerman K, Gilder A, Dredze M, Pereira F. Reading the markets: forecasting public opinion of political candidates by news analysis. In: Proceedings of the 22nd international conference on computational. – Linguistics 1, 2008. – С. 473–480
4. Chaffey D. Global social media research summary 2020. URL: <https://www.smartinsights.com/social-media-marketing/socialmedia-strategy/new-global-social-media-research>.
5. The Best and Most Current of Modern Natural Language Processing. URL: <https://medium.com/huggingface/the-best-and-most-current-ofmodern-natural-language-processing-5055f409a1d1>.
6. Устенко, М. О., Руських А. О. – Діджиталізація: основа конкурентоспроможності підприємства в реаліях цифрової економіки. Вісник економіки транспорту і промисловості 68, 2019
7. Ворон, А. А., В. А. Солопенко. "Українська мова." *Профільний рівень: підруч. для 10.*
8. Соколовська С. Ф. Полісемія в тексті: типи реалізації та функції. Вісник Житомирського державного університету імені Івана Франка, 2003, 11: 201-205.
9. Kantemir S. Полісемія vs. Омонімія: труднощі розмежування. Наукові записки Національного університету «Острозька академія»: Серія «Філологія», 2012, 1.24: 104-107.
10. Залуцька О. О., Мазурець О. В. Інформаційний портрет ключових термінів у цифрових навчальних матеріалах. 2019.

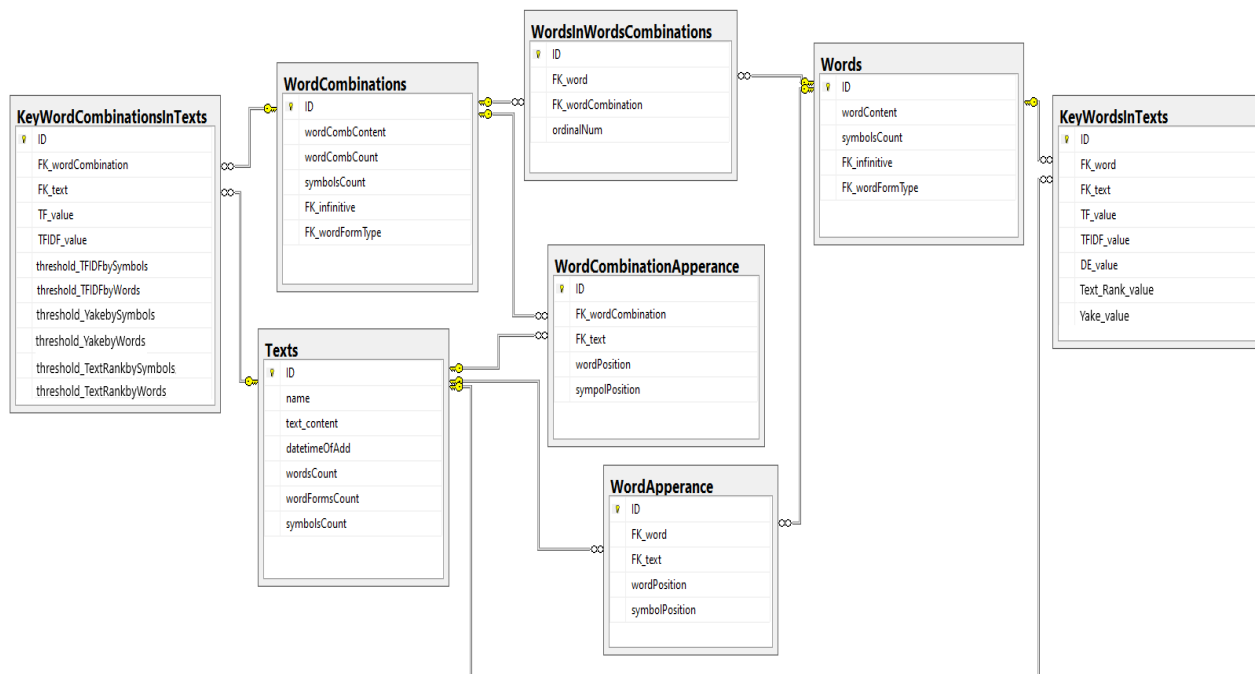
11. Sassi I. B., Yahia S. B., Mellouli S. Fuzzy classification-based emotional context recognition from online social networks messages. 2017 IEEE International Conference on Fuzzy Systems. 2017. С. 1-6.
12. Su Q. Motivations, methods and metrics of misinformation detection: an NLP perspective. Natural Language Processing Research. 2020. №. 1-2. С. 1-13.
13. Старко, В., Андрій Рисін. "Великий електронний словник української мови (ВЕСУМ) як засіб NLP для української мови. 2020.
14. Біленький В. С., Кобзєв В. Г., Матвєєв Д. І. Методи Оцінювання Тонального Забарвлення Тексту. 2020. С. 89.
15. Huang Z., Xie Z. A patent keywords extraction method using TextRank model with prior public knowledge. Complex & Intelligent Systems. 2021. С. 1-12.
16. Campos R. et al. YAKE! Keyword extraction from single documents using multiple local features. Information Sciences. 2020. С. 257-289.
17. Сайт кафедри комп'ютерних наук ХНУ. URL: <https://kn.khnu.km.ua/page.aspx>
18. Сайт «Miratext». URL: [https://miratext.ru/seo\\_analiz\\_text/](https://miratext.ru/seo_analiz_text/)
19. Сайт «MonkeyLearn». URL: <https://monkeylearn.com/sentiment-analysis-online/>
20. Microsoft .NET. URL: <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>
21. Java benefits. URL: <https://www.ibm.com/docs/en/aix/7.1?topic=monitoring-advantages-java>
22. Benefits of .NET. URL: <https://scand.com/company/blog/net-vs-java-comparison/>
23. .NET Core. URL: <https://www.xelent.en/blog/>
24. .NET Framework 4.5. URL: <https://www.microsoft.com/en-en/download/details.aspx?id=30653>
25. .NET Framework 4.5 Benefits. URL: <https://www.csharpens.com/articles/what-is-net-framework-uses-and-benefits-1035/>

26. Мазурець О. В., Залуцька О. О. Інформаційний портрет ключових термінів у цифрових навчальних матеріалах, 2019.
27. Qaiser S., Ali R. Text mining: use of TF-IDF to examine the relevance of words to documents, International Journal of Computer Applications, №. 1, С. 25-29.
28. Campos R, YAKE! Keyword extraction from single documents using multiple local features, Information Sciences, 2020, С. 257-289.
29. В.О. Брацький, О.М. М'якшило, «Дослідження особливостей застосування реляційних і нереляційних баз даних на прикладі SQL Server та MongoDB», Наукові праці Національного університету харчових технологій, Т. 22, № 5, с. 15-24, 2016
30. C# Language. URL: <https://www.codeguru.com/csharp/benefits-of-c/>
31. Benefits of C#. URL: <https://www.bairesdev.com/technologies/csharp/>
- 32 Bishop J., Horspool R. N., Worrall B. Experience in integrating Java with C# and .NET. Concurrency and Computation: Practice and Experience, 2005, Т. 17, №. 5-6, С. 663-680.
33. Benefits of using Java. URL: <https://data-flair.training/blogs/pros-and-cons-of-java/>
34. Eclipse IDE. URL: <https://uk.wikipedia.org/wiki/Eclipse>
35. Visual Studio. URL: <https://www.microsoft.com/rus/msdn/vs/default.aspx>
36. MySQL. URL: <https://www.mysql.com/>.
37. MS SQL Server. URL: <https://www.microsoft.com/ru-ru/download/details.aspx?id=35579>

# ДОДАТКИ

## Додаток А

## Структура бази даних інформаційної технології інтелектуальної асоціації текстових повідомлень за семантичними ознаками



## Додаток Б

### Програмні коди

#### Лістинг Form1.cs:

```
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

    }
    private void button1_Click(object sender, EventArgs e)
    {
        візитиBindingSource.AddNew();
    }
    private void button2_Click(object sender, EventArgs e)
    {
        візитиBindingSource.EndEdit();
        візитиTableAdapter.Update(dataSet1);
    }
    private void button3_Click(object sender, EventArgs e)
    {
        Form2 form2 = new Form2();
        this.Hide();
        form2.ShowDialog();
        this.Show();
    }
    private void button7_Click(object sender, EventArgs e)
    private void button8_Click(object sender, EventArgs e)
    {
        Form15 form15 = new Form15();
        this.Hide();
        form15.ShowDialog();
        this.Show();
    }
}
}
```

#### Лістинг Form2.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace kursach
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void BindingNavigatorSaveItem_Click(object sender, EventArgs e)
        {
            this.Validate();
            this.BindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.dataSet1);
            this.прихожаниBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.dataSet1);
            this.BindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.dataSet1);
            this.BindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.dataSet1);
        }
        private void Form2_Load(object sender, EventArgs e)
        {
            this.TableAdapter.Fill(this.dataSet1.Прихожани);
        }
        private void button1_Click(object sender, EventArgs e)
```

#### Лістинг Form3.cs:

```
using System;
using System.Collections.Generic; using System.ComponentModel; using System.Data;
```

```

using System.Drawing; using System.Linq; using System.Text;
using System.Threading.Tasks; using System.Windows.Forms;

namespace klasyfikator
{
public partial class Form1 : Form
{
int k,z; public Form1()
{
InitializeComponent();
}

private void вихідToolStripMenuItem_Click(object sender, EventArgs e)
{
this.Close();
}

private void button1_Click(object sender, EventArgs e)
{
z = Convert.ToInt32(textBox2.Text); k = Convert.ToInt32(textBox1.Text); if (k < 2 || k > 5)
{
MessageBox.Show("Введіть значення в межах від 2 до 5");
}
else
{
Form2 f2 = new Form2(k,z); f2.Show();
}
}
}

string[] mas = { "\", "-" };

List<string> text1 = new
List<string>(); List<string> text2 =
new List<string>(); List<string> text3
= new List<string>(); List<string>
text4 = new List<string>();
List<string> text5 = new
List<string>();

List<string> text_1 = new
List<string>(); List<string> text_2 =
new List<string>(); List<string> text_3
= new List<string>(); List<string>
text_4 = new List<string>();
List<string> text_5 = new
List<string>();

List<string> test_text = new List<string>();
string nazva;
Text t = new Text();
Methods Met = new
Methods(); Sum Count = new
Sum();
Classification Kategor = new Classification();

string[]
mas_text; string
text_new = "";
int zagal_kilk =
0; int number_vk1
= 1; int len_gr;
int k =
0; string
text;
int[]
index;
string[,]
mas_ob_; double[]
rezerv;
double temp; int temp_ind;

string[]
nowy_mas_test;
string[] nowy_mas;
string[] nowy_mas_2;
string[] nowy_mas_3;
string[] nowy_mas_4;
string[] nowy_mas_5;

int[]
mas_kilk_;
int[] mas_kilk;
int[]
mas_kilk_2;
int[]
mas_kilk_3;

```

```

int[]
mas_kilk_4;
int[]
mas_kilk_5;

string[,]
mas_ob_1;
string[,]
mas_ob_2;
string[,]
mas_ob_3;
string[,]
mas_ob_4;
string[,]
mas_ob_5;
int len = 0, i_len = 0;
string[] mas1;
double[] chast_;
string[] mas2;
double[] chast_2;
int[] mas_kilk2;
string[] mas3;
double[] chast_3;
int[] mas_kilk3;
string[] mas4;
double[] chast_4;
int[] mas_kilk4;
string[] mas5;
double[] chast_5;
int[] mas_kilk5;

int[] kilk_text;
string txt1, txt2, txt3, txt4, txt5,
txt6; string[] mas_text2;
string[]
mas_text3;
string[]
mas_text4;
string[]
mas_text5;

double[] chast;
double[] chast2;
double[] chast3;
double[] chast4;
double[] chast5;

double[] idf1;
double[] tf_idf1;

double[] idf2;
double[] tf_idf2;

double[] idf3;
double[] tf_idf3;

double[] idf4;
double[] tf_idf4;

double[] idf5;
double[] tf_idf5;
int len_for_kat = 30;

#endregion
public void text_new_txt(string nazva)
{
    WordprocessingDocument wordDoc = WordprocessingDocument.Open(@Environment.CurrentDirectory + "/" + nazva +
".docx", true);
    Body body = wordDoc.MainDocumentPart.Document.Body;
    var qwe = body.ChildElements;
    foreach (var child_element in qwe)
    {
        var asdf = child_element.InnerText;
        File.AppendAllText(@nazva + ".txt", '\n' + asdf);
    }
}
public List<string> txt_stati(string nazva, List<string> lines)
{
    string[] b = File.ReadAllText(@nazva + ".txt", Encoding.UTF8).
        Split(new Char[] { '\n', ' ' }, StringSplitOptions.RemoveEmptyEntries);

    for (int i = 0; i < b.Length; i++)
    {
        lines.Add(b[i]);
    }
}

```

```

    }
    return lines;
}

public Form2(int k, int z)
{
    InitializeComponent();

    ClassifierEntities context = new ClassifierEntities();
    var text = context.THEM_TEXT;
    foreach (var i in text)
    {
        if (i.ClassificationId == 1)
            text_1.Add(i.Text);

        else if (i.ClassificationId ==
            2) text_2.Add(i.Text);

        else if (i.ClassificationId ==
            3) text_3.Add(i.Text);

        else if (i.ClassificationId ==
            4) text_4.Add(i.Text);

        else if (i.ClassificationId ==
            5) text_5.Add(i.Text);
    }

    #region
    button if
    (k == 5)
    {

        button4.Visible = true;
        button5.Visible = true;
    }
    else if (k == 4)
        button3.Visible = true;
        button4.Visible = true;
    }
    else if (k == 3)
    {
        button3.Visible = true;
    }
    #endregion

    button7.Visible = false;
    button8.Visible = false;

    for (int i = 0; i < text_1.Count; i++)
        richTextBox1.AppendText(text_1[i].ToString() + " ");
}

private void button1_Click(object sender, EventArgs e)
{
    number_vk1 = 1;
    richTextBox1.Clea
r();
    for (int i = 0; i < text_1.Count; i++)
        richTextBox1.AppendText(text_1[i].ToString() + " ");
    richTextBox4.Clea
r();
    richTextBox5.Clea
r();
    richTextBox6.Clea
r();
    richTextBox7.Clea
r();

    for (int i = 0; i < text_2.Count; i++)
        richTextBox4.AppendText(text_2[i].ToString() + " ");

    for (int i = 0; i < text_3.Count; i++)
        richTextBox5.AppendText(text_3[i].ToString() + " ");

    for (int i = 0; i < text_4.Count; i++)
        richTextBox6.AppendText(text_4[i].ToString() + " ");

    for (int i = 0; i < text_5.Count; i++)
        richTextBox7.AppendText(text_5[i].ToString() + " ");
}

private void button2_Click(object sender, EventArgs e)

```

```

{
    number_vk1 = 2;
    richTextBox1.Clear();
    richTextBox4.Clear();
    richTextBox5.Clear();
    richTextBox6.Clear();
    richTextBox7.Clear();

    for (int i = 0; i < text_2.Count; i++)
        richTextBox1.AppendText(text_2[i].ToString() + " ");

    for (int i = 0; i < text_1.Count; i++)
        richTextBox4.AppendText(text_1[i].ToString() + " ");

    for (int i = 0; i < text_3.Count; i++)
        richTextBox5.AppendText(text_3[i].ToString() + " ");

    for (int i = 0; i < text_4.Count; i++)
        richTextBox6.AppendText(text_4[i].ToString() + " ");

    for (int i = 0; i < text_5.Count; i++)
        richTextBox7.AppendText(text_5[i].ToString() + " ");
}

private void tabControl1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (tabControl1.SelectedTab.Name == "tabPage2") { button6.Visible = false; button7.Visible = true; button8.Visible = false;

        button9.Text = "Автоматизований підбір альтернативних текстів";
        button10.Text = "Обрахунок значень TF IDF для унікальних слів обраного класу";
        button11.Text = "Сортування та обмеження множини ключових слів класу";
    }
    if (tabControl1.SelectedTab.Name == "tabPage1") { button6.Visible = true; button7.Visible = false;
    button8.Visible = false;
        button9.Text = "Обробити тексти";
        button10.Text = "Сформувати послідовні множини слів статей";
        button11.Text = "Сформувати унікальні множини слів статей";
    }
    if (tabControl1.SelectedTab.Name == "tabPage3") { button6.Visible = false; button7.Visible = false;
    button8.Visible = true;
        button9.Text = "Обробка тексту статті";
        button10.Text = "Пошук збігу тестової статті за ключовими словами класів";
        button11.Text = "Обрахунок оцінки приналежності обраної статті до класів";
    }
}

public void grid_vyvod(string[,] mas_ob_mas, int len_gr)
{
    dataGridView3.RowCount = len_gr;
    for (int i = 0; i < len_gr; i++)
    {
        dataGridView3.Rows[i].Cells[0].Value = mas_ob_mas[i, 0];
        dataGridView3.Rows[i].Cells[1].Value = mas_ob_mas[i, 1];
        dataGridView3.Rows[i].Cells[2].Value = mas_ob_mas[i, 2];
        dataGridView3.Rows[i].Cells[3].Value = mas_ob_mas[i, 3];
    }
}

private void button7_Click(object sender, EventArgs e)
{
    #region TF-IDF kat1

    kilk_text = new int[nowy_mas.Length];
    txt1 = richTextBox4.Text.ToLower();
    txt2 = richTextBox5.Text.ToLower();
    txt3 = richTextBox6.Text.ToLower();
    txt4 = richTextBox7.Text.ToLower();

    text_new = "";
    mas_text2 = t.txt1(txt1, mas, text_new);
    text_new = "";
    mas_text3 = t.txt1(txt2, mas, text_new);
    text_new = "";
    mas_text4 = t.txt1(txt3, mas, text_new);
    text_new = "";
    mas_text5 = t.txt1(txt4, mas, text_new);
    text_new = "";

    idf1 = new double[nowy_mas.Length];
    t.kilkist_textiv(nowy_mas, mas_text2, mas_text3, mas_text4, mas_text5, kilk_text);
}

```

```

tf_idf1 = new
double[nowy_mas.Length];
Met.tf_idf(nowy_mas, kilk_text,
idf1);

#endregion
#region TF-IDF kat2

kilk_text = new int[nowy_mas_2.Length];
idf2 = new double[nowy_mas_2.Length];
t.kilkist_textiv(nowy_mas_2, mas_text2, mas_text3, mas_text4, mas_text5, kilk_text);

tf_idf2 = new double[nowy_mas_2.Length];
Met.tf_idf(nowy_mas_2, kilk_text, idf2);

#endregion
#region TF-IDF kat3

kilk_text = new int[nowy_mas_3.Length];
idf3 = new double[nowy_mas_3.Length];
t.kilkist_textiv(nowy_mas_3, mas_text2, mas_text3, mas_text4, mas_text5, kilk_text);

tf_idf3 = new double[nowy_mas_3.Length];
Met.tf_idf(nowy_mas_3, kilk_text, idf3);

#endregion
#region TF-IDF kat4

kilk_text = new int[nowy_mas_4.Length];
idf4 = new double[nowy_mas_4.Length];
t.kilkist_textiv(nowy_mas_4, mas_text2, mas_text3, mas_text4, mas_text5, kilk_text);

tf_idf4 = new double[nowy_mas_4.Length];
Met.tf_idf(nowy_mas_4, kilk_text, idf4);

#endregion
#region TF-IDF kat5

kilk_text = new int[nowy_mas_5.Length];
idf5 = new double[nowy_mas_5.Length];
t.kilkist_textiv(nowy_mas_5, mas_text2, mas_text3, mas_text4, mas_text5, kilk_text);

tf_idf5 = new double[nowy_mas_5.Length];
Met.tf_idf(nowy_mas_5, kilk_text, idf5);

#endregion
mas_ob_1 = grid(nowy_mas, chast, idf1, tf_idf1);
mas_ob_2 = grid(nowy_mas_2, chast2, idf2, tf_idf2);
mas_ob_3 = grid(nowy_mas_3, chast3, idf3, tf_idf3);
mas_ob_4 = grid(nowy_mas_4, chast4, idf4, tf_idf4);
mas_ob_5 = grid(nowy_mas_5, chast5, idf5, tf_idf5);

if (number_vkl == 1)
{
    len_gr = nowy_mas.Length;
    grid_vyvod(mas_ob_1, len_gr);
}

else if(number_vkl == 2)
{
    len_gr = nowy_mas_2.Length;
    grid_vyvod(mas_ob_2, len_gr);
}
}

//запис tf_idf 2 вкладка
public string[,] grid(string[] nowy_mas, double[] chast_i, double[] idf, double[] tf_idf)
{
    mas_ob_ = new string[nowy_mas.Length, 4];
    index = new int[nowy_mas.Length];
    rezerv = new double[tf_idf.Length];
    for (int i = 0; i < tf_idf.Length; i++)
    {
        tf_idf[i] = chast_i[i] * idf[i];
        rezerv[i] = tf_idf[i];
        index[i] = i;
    }
    //сортування
    for (int j = 0; j < rezerv.Length - 1; j++)
    {
        for (int i = j + 1; i < rezerv.Length; i++)
        {
            if (rezerv[j] * (-1) > rezerv[i] * (-1))
            {

```

```

        temp = rezerv[j];
        temp_ind =
        index[j];

        rezerv[j] = rezerv[i];
        index[j] = index[i];

        rezerv[i] = temp;
        index[i] =
        temp_ind;
    }
}

for (int i = 0; i < rezerv.Length; i++)
{
    mas_ob_[i, 0] = nowy_mas[index[i]];
    mas_ob_[i, 1] =
    chast_i[index[i]].ToString(); mas_ob_[i, 2] =
    idf[index[i]].ToString(); mas_ob_[i, 3] =
    rezerv[i].ToString();
}
return mas_ob_;
}
public void classif(string text, string[] mas_text, string text_new)
{
    text_new =
    "";
    test_text.Clea
    r(); k = 0;
    for (int i = 0; i < mas_text.Length; i++)
    {
        test_text.Add(mas_text[i]);
        mas_text[i] = "";
    }
    for (int i = 0; i < test_text.Count; i++)
    {
        if (test_text[i] == "\n" || test_text[i] == " " || test_text[i] ==
        "" || string.IsNullOrEmpty(test_text[i]))
        {
        }
        else
        {
        }
    }
}
test_text.Remove(test_text[i]);

private void button8_Click(object sender, EventArgs e)
{
    text = richTextBox15.Text.ToLower();

    mas_text = t.txt1(text, mas, text_new);
    classif(text, mas_text, text_new);

    int[] mas_kilk_test = new int[mas_text.Length];
    string[] mas_slova = new string[mas_text.Length];

    mas_slova[0] = mas_text[0];
    zagal_kilk = 0;

    nowy_mas_test = t.mas(mas_text, mas_slova, mas_kilk_test);

    dataGridView8.RowCount = nowy_mas_test.Length + 1;
    dataGridView9.RowCount = 5;
    double[] znach = new double[nowy_mas_test.Length];
    double[] znach_2 = new double[nowy_mas_test.Length];
    int z = 1;
    Random rand = new Random();

    for (int i = 0; i < nowy_mas_test.Length; i++)
    {nowy_mas_test); nowy_mas_test); nowy_mas_test); nowy_mas_test); nowy_mas_test);
    dataGridView8.Rows[i].Cells[0].Value = nowy_mas_test[i]; z = 1;
    Kategor.klas_statti(dataGridView8, len_for_kat, mas_ob_1, mas_kilk_test, i, tf_idf1, z,
    z = 2;
    Kategor.klas_statti(dataGridView8, len_for_kat, mas_ob_2, mas_kilk_test, i, tf_idf2, z,
    z = 3;
    Kategor.klas_statti(dataGridView8, len_for_kat, mas_ob_3, mas_kilk_test, i, tf_idf3, z,
    z = 4;
    Kategor.klas_statti(dataGridView8, len_for_kat, mas_ob_4, mas_kilk_test, i, tf_idf4, z,

```

```

z = 5;
Kategor.klas_statti(dataGridView8, len_for_kat, mas_ob_5, mas_kilk_test, i, tf_idf5, z,
    zagal_kilk += mas_kilk_test[i];
}

dataGridView8.Rows[nowy_mas_test.Length].Cells[0].Value = "Cyma"; double
s = 0;
int q = 1;

for (int i = 0; i < 5; i++)
{
    Count.Summ(dataGridView8, s, nowy_mas_test, q);
    s = 0; q++;
}

dataGridView9.Rows[0].Cells[0].Value = Column6.HeaderText;
dataGridView9.Rows[1].Cells[0].Value = Column7.HeaderText;
dataGridView9.Rows[2].Cells[0].Value = Column8.HeaderText;
dataGridView9.Rows[3].Cells[0].Value = Column9.HeaderText;
dataGridView9.Rows[4].Cells[0].Value = Column10.HeaderText;

double sum=0;
for (int i = 0; i < 5; i++)
{
    1].Value;
sum += Convert.ToDouble(dataGridView8.Rows[nowy_mas_test.Length].Cells[i + 1].Value); dataGridView9.Rows[i].Cells[1].Value
    = dataGridView8.Rows[nowy_mas_test.Length].Cells[i +
}

for (int i = 0; i < 5; i++)
{
    dataGridView9.Rows[i].Cells[2].Value
Math.Round(Convert.ToDouble(dataGridView9.Rows[i].Cells[1].Value) / sum * 100,2) + "%";
}
}

private void button3_Click(object sender, EventArgs e)
{
    richTextBox1.Clear();
    for (int i = 0; i < text_3.Count; i++)
        richTextBox1.AppendText(text_3[i].ToString() + " ");
}

private void button4_Click(object sender, EventArgs e)
{
    richTextBox1.Clear();
    for (int i = 0; i < text_4.Count; i++)
        richTextBox1.AppendText(text_4[i].ToString() + " ");
}

private void button5_Click(object sender, EventArgs e)
{
    number_vk1 = 5;
    richTextBox1.Clear();
    for (int i = 0; i < text_5.Count; i++)
        richTextBox1.AppendText(text_5[i].ToString() + " ");
}

private void button6_Click(object sender, EventArgs e)
{
    string[]
    mas_slova;
    richTextBox2.Clear
    ();
    richTextBox3.Clear
    ();
    if
    (number_vk1 == 1)
    {
        #region
        tabl k
        = 0;
        string text = richTextBox1.Text.ToLower();

        mas_text = t.txt1(text, mas, text_new);
        text_new = "";

        text1.Clear();

        for (int i = 0; i < mas_text.Length; i++)
        {
            text1.Add(mas_text[i]);
            mas_text[i] = "";

```

```

}
for (int i = 0; i < text1.Count; i++)
{
    if (text1[i] == "\n" || text1[i] == " " || text1[i] == "" || string.IsNullOrEmpty(text1[i]))
    {
        text1.Remove(text1[i]);
    }
    else
    {
        richTextBox2.AppendText(text1[i] + "\n");
        richTextBox3.AppendText(text1[i].ToString() + " ");
        mas_text[k] = text1[i];
        k++;
    }
}
mas_kilk = new int[mas_text.Length];
mas_slova = new string[mas_text.Length];

mas_slova[0] = mas_text[0];
zagal_kilk = 0;
nowy_mas = t.mas(mas_text, mas_slova, mas_kilk);
for (int i = 0; i < nowy_mas.Length; i++)
{
    zagal_kilk += mas_kilk[i];
}
#endregion

//Chastota
chast = new double[nowy_mas.Length];
Met.tf(nowy_mas, mas_kilk, zagal_kilk, chast);
textBox3.Text = (zagal_kilk - mas_kilk[nowy_mas.Length - 1]).ToString();
for (int i = 0; i < mas_text.Length; i++)
{
    mas_text[i] = "";
}
i_len = 0;
for (int i = 0; i < nowy_mas.Length; i++)
{
    if (mas_kilk[i] > 1)
    { i_len++; }
}

mas1 = new string[i_len];
chast_ = new double[i_len];
mas_kilk_ = new int[i_len];
grid2_0(nowy_mas, mas_kilk, chast, mas1, chast_, mas_kilk_); grid2_0_v(dataGridView1,
mas1, mas_kilk_, chast_);
}

if (number_vk1 == 2)
{
    #region
    tab2 k
    = 0;
    string text = richTextBox1.Text.ToLower();

    mas_text = t.txt1(text, mas, text_new);
    text_new = "";

    text2.Clear();

    for (int i = 0; i < mas_text.Length; i++)
    {
        text2.Add(mas_text[i]);
        mas_text[i] = "";
    }

    for (int i = 0; i < text2.Count; i++)
    {
        if (text2[i] == "\n" || text2[i] == " " || text2[i] == "" || string.IsNullOrEmpty(text2[i]))
        {
            text2.Remove(text2[i]);
        }
        else
        {
            richTextBox2.AppendText(text2[i] + "\n");
            richTextBox3.AppendText(text2[i].ToString() + " ");
            mas_text[k] = text2[i];
            k++;
        }
    }
}

```

```

}
mas_kilk_2 = new int[mas_text.Length];
mas_slova = new string[mas_text.Length];

mas_slova[0] = mas_text[0];
zagal_kilk = 0;
nowy_mas_2 = t.mas(mas_text, mas_slova, mas_kilk_2);

for (int i = 0; i < nowy_mas_2.Length; i++)
{
    zagal_kilk += mas_kilk_2[i];
}
//Chastota
chast2 = new double[nowy_mas_2.Length];
Met.tf(nowy_mas_2, mas_kilk_2, zagal_kilk, chast2);

textBox3.Text = (zagal_kilk - mas_kilk_2[nowy_mas_2.Length - 1]).ToString(); i_len =
0;
for (int i = 0; i < nowy_mas_2.Length; i++)
{
    if (mas_kilk_2[i] > 1)
    { i_len++; }
}

mas2 = new string[i_len];
chast_2 = new double[i_len];
mas_kilk2 = new int[i_len];
grid2_0(nowy_mas_2, mas_kilk_2, chast2, mas2, chast_2, mas_kilk2);
grid2_0_v(dataGridView1, mas2, mas_kilk2, chast_2);
#endregion

#region
tab1_3 k
= 0;
text = richTextBox5.Text.ToLower();

mas_text = t.txt1(text, mas, text_new);
text_new = "";

text3.Clear();

for (int i = 0; i < mas_text.Length; i++)
{
    text3.Add(mas_text[i]);
    mas_text[i] = "";
}

for (int i = 0; i < text3.Count; i++)
{
    if (text3[i] == "\n" || text3[i] == " " || text3[i] ==
        "" || string.IsNullOrEmpty(text3[i]))
    {
        text3.Remove(text3[i]);
    }
    else
    {
        mas_text[k] = text3[i];
        k++;
    }
}
mas_kilk_3 = new int[mas_text.Length];
mas_slova = new string[mas_text.Length];

mas_slova[0] = mas_text[0];
zagal_kilk = 0;
nowy_mas_3 = t.mas(mas_text, mas_slova, mas_kilk_3);

for (int i = 0; i < nowy_mas_3.Length; i++)
{
    zagal_kilk += mas_kilk_3[i];
}

//Chastota
chast3 = new double[nowy_mas_3.Length];
Met.tf(nowy_mas_3, mas_kilk_3, zagal_kilk, chast3);

i_len = 0;
for (int i = 0; i < nowy_mas_3.Length; i++)
{
    if (mas_kilk_3[i] > 1)

```

```

    { i_len++; }
}

mas3 = new string[i_len];
chast_3 = new double[i_len];
mas_kilk3 = new int[i_len];

#endregio
n #region
tab1_4 k
= 0;
text = richTextBox6.Text.ToLower();

mas_text = t.txt1(text, mas, text_new);
text_new = "";

text4.Clear();

for (int i = 0; i < mas_text.Length; i++)
{
    text4.Add(mas_text[i]);
    mas_text[i] = "";
}

for (int i = 0; i < text4.Count; i++)
{
    if (text4[i] == "\n" || text4[i] == " " || text4[i] ==
        "" || string.IsNullOrEmpty(text4[i]))
    {
        text4.Remove(text4[i]);
    }
    else
    {
        mas_text[k] = text4[i];
        k++;
    }
}
mas_kilk_4 = new int[mas_text.Length];
mas_slova = new string[mas_text.Length];

mas_slova[0] = mas_text[0];
zagal_kilk = 0;
nowy_mas_4 = t.mas(mas_text, mas_slova, mas_kilk_4);

for (int i = 0; i < nowy_mas_4.Length; i++)
{
    zagal_kilk += mas_kilk_4[i];
}

//Chastota
chast4 = new double[nowy_mas_4.Length];
Met.tf(nowy_mas_4, mas_kilk_4, zagal_kilk, chast4);

i_len = 0;
for (int i = 0; i < nowy_mas_4.Length; i++)
{
    if (mas_kilk_4[i] > 1)
    { i_len++; }
}

mas4 = new string[i_len];
chast_4 = new double[i_len];
mas_kilk4 = new int[i_len];

#endregio
n #region
tab1_5 k
= 0;
text = richTextBox7.Text.ToLower();

mas_text = t.txt1(text, mas, text_new);
text_new = "";

text5.Clear();

for (int i = 0; i < mas_text.Length; i++)
{
    text5.Add(mas_text[i]);
    mas_text[i] = "";
}

```

```

for (int i = 0; i < text5.Count; i++)
{
    if (text5[i] == "\n" || text5[i] == " " || text5[i] == "" || string.IsNullOrEmpty(text5[i]))
    {
        text5.Remove(text5[i]);
    }
    else
    {
        mas_text[k] = text5[i];
        k++;
    }
}
mas_kilk_5 = new int[mas_text.Length];
mas_slova = new string[mas_text.Length];

mas_slova[0] = mas_text[0];
zagal_kilk = 0;
nowy_mas_5 = t.mas(mas_text, mas_slova, mas_kilk_5);

for (int i = 0; i < nowy_mas_5.Length; i++)
{
    zagal_kilk += mas_kilk_5[i];
}

//Chastota
chast5 = new double[nowy_mas_5.Length];
Met.tf(nowy_mas_5, mas_kilk_5, zagal_kilk, chast5);

i_len = 0;
for (int i = 0; i < nowy_mas_5.Length; i++)
{
    if (mas_kilk_5[i] > 1)
    { i_len++; }
}

mas5 = new string[i_len];
chast_5 = new double[i_len];
mas_kilk5 = new int[i_len];

#endregion
}

if (number_vk1 == 5)
{
    #region
    tab1_5 k
    = 0;
    text = richTextBox7.Text.ToLower();

    mas_text = t.txt1(text, mas, text_new);
    text_new = "";

    text5.Clear();

    for (int i = 0; i < mas_text.Length; i++)
    {
        text5.Add(mas_text[i]);
        mas_text[i] = "";
    }

    for (int i = 0; i < text5.Count; i++)
    {
        if (text5[i] == "\n" || text5[i] == " " || text5[i] == "" || string.IsNullOrEmpty(text5[i]))
        {
            text5.Remove(text5[i]);
        }
        else
        {
            richTextBox2.AppendText(text5[i] + "\n");
            richTextBox3.AppendText(text5[i].ToString() + " ");
            mas_text[k] = text5[i];
            k++;
        }
    }
}

```

```

mas_kilk_5 = new int[mas_text.Length];
mas_slova = new string[mas_text.Length];

mas_slova[0] = mas_text[0];
zagal_kilk = 0;
nowy_mas_5 = t.mas(mas_text, mas_slova, mas_kilk_5);

for (int i = 0; i < nowy_mas_5.Length; i++)
{
    zagal_kilk += mas_kilk_5[i];
}

//Chastota
chast5 = new double[nowy_mas_5.Length];
Met.tf(nowy_mas_5, mas_kilk_5, zagal_kilk, chast5);

textBox3.Text = (zagal_kilk - mas_kilk_5[nowy_mas_5.Length - 1]).ToString(); i_len =
0;
for (int i = 0; i < nowy_mas_5.Length; i++)
{
    if (mas_kilk_5[i] > 1)

        if (Convert.ToString(text[i]) == mas_chys[j])
        {
            k = 1;
            break;
        }
        }

    for (j = 0; j < mas.Length; j++)
    {

        if (text[i] == Convert.ToChar(mas[j])) { k = 0; }
    }

    if (k == 0)
    {
        text_new += Convert.ToString(text[i]);
    }
    k = 0;
}

return text_new.Split(' ');
}
public string[] mas(string[] mas_text, string[] mas_slova, int[] mas_kilk)
{
    int i, j;
    int kilk, zmin = 1, zagal_kilk = 0;
    for (i = 1; i < mas_text.Length;
i++)
    {
        kilk = 0;

        for (j = 0; j < mas_text.Length; j++)
        {
            if (mas_text[i] == mas_slova[j])
            {
                kilk++;
            }
        }

        if (kilk == 0)
        {
            mas_slova[zmin] =
            mas_text[i]; zmin++;
        }
    }

    kilk = 0;
    for (i = 0; i < mas_slova.Length;
i++) if (mas_slova[i] != null)
        kilk++;

    string[] nowy_mas = new string[kilk];

    //перезаписую в новый массив всі не нул елементи
    for (i = 0; i < nowy_mas.Length; i++)
        nowy_mas[i] = mas_slova[i];
}

```



```

{
    class Methods
    {
        //TF
        public double[] tf(string[] nowy_mas, int[] mas_kilk, int zagal_kilk, double[] chast)
        {
            for (int i = 0; i < nowy_mas.Length; i++)
            {
                chast[i] = Convert.ToDouble(mas_kilk[i]) / zagal_kilk;
            }
            return chast;
        }
        //TF-IDF
        public double[] tf_idf(string[] nowy_mas, int[] kilk_text, double[] idf)
        {
            for (int i = 0; i < nowy_mas.Length; i++)
            {
                idf[i] = Math.Log((7 / (kilk_text[i] + 1)));
            }
            return idf;
        }
    }
}
Лістинг Sum.cs:
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;

namespace klasyfikator
{
    class Sum
    {
        public void Summ(DataGridView dataGridView8, double s, string[] nowy_mas_test, int k)
        {
            for (int i = 0; i < nowy_mas_test.Length; i++)
            {
                s += Convert.ToDouble(dataGridView8.Rows[i].Cells[k].Value);
            }
            dataGridView8.Rows[nowy_mas_test.Length].Cells[k].Value = s;
        }
    }
}

```

## Додаток В

### Презентаційний матеріал

# КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Метод автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками

Виконала: студентка 4 курсу групи КН-18-1 Залуцька Ольга Олександрівна  
Керівник: викладач кафедри КН Молчанова Марина Олексіївна

## Актуальність

Для більшості людей комунікація в мережі Інтернет стала основною, через соціальні сервіси проходить колосальна частина інформації щосекунди.

Автоматизоване визначення тональності тексту допомагатиме у класифікації повідомлень, відгуків, коментарів, тощо.

Засоби визначення емоційного забарвлення тексту використовують торгові компанії, бізнес-сегменті та тих сферах, що досліджують громадську думку.

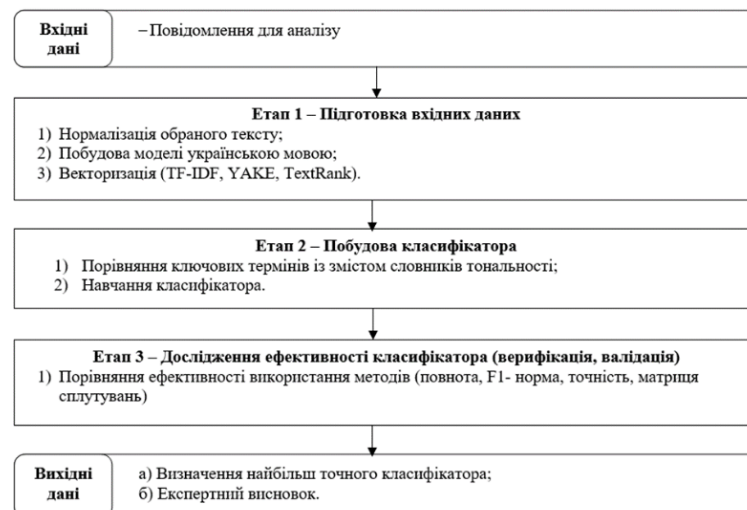


## Завдання

Мета роботи - розробка методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками

- перегляд доступних повідомлень та текстових одиниць для аналізу;
- обрахунок оцінки семантичної важливості обраного унікального слова, що присутнє в обраному повідомленні;
- визначення множин слів текстового повідомлення, які містять семантичне забарвлення різних психоемоційних категорій;
- автоматизоване візуальне відображення психоемоційної тональності слів повідомлення;
- обрахунок числових оцінок психоемоційної тональності текстового повідомлення;
- автоматизоване візуальне відображення психоемоційної тональності кожного текстового повідомлення.

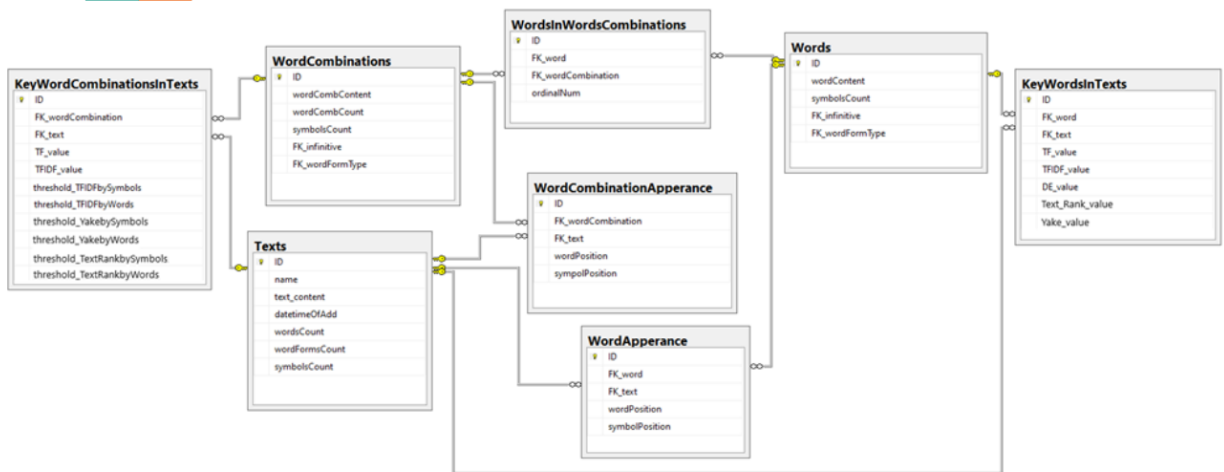
## Схема методу автоматизованого визначення психоемоційної тональності



## Структура компонентів методу автоматизованого визначення психоемоційної тональності



## Схема бази даних



## Робота методу автоматизованого визначення психоемоційної тональності

Аналіз емоційного забарвлення текстів

База повідомлень    Появи семантичних одиниць    Аналіз повідомлень

Повідомлення

Допис

Контент

Друзі, сьогодні о 15 години проведемо зустріч в нашому молодіжному центрі. Тема "Як захистити себе від негативного впливу Інтернету". Приводьте друзів та знайомих, будемо раді. Орієнтовний час півтори години. Дякуємо за лайк та поширення на ваших сторінках. До зустрічі!

Результати:

ID	Позитивна	Негативна	Нейтральна
1	95.2 %	3.16 %	1.64 %
*			

Тональність, визначена алгоритмом

Назва повідомлення: "Допис"  
Платформа, на якій розміщено "Допис" - Facebook

За результати дослідження, цей текст із вірогідністю 95.2 % містить позитивне забарвлення

Із вірогідністю 3.16 % є негативним

## Anti-Plagiarism v-15.257

**Максимальное совпадение с одним документом 2.0%**

**Словари проверки: en\_US, ru\_RU, ua\_UA. Ошибок в документах: 11%**

ID: 104156 Название: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА Метод автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально- орієнтованих сервісах за семантичними ознаками Добавлено в БД: 2022-05-30 Авторы: О.О. Залуцька Руководители: М.О.Молчанова Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	52761	791	3541 (7%)	55 (7%)

### Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы



Ім'я користувача:  
Кафедра КН

ID перевірки:  
1011369411

Дата перевірки:  
30.05.2022 08:15:28 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
30.05.2022 08:15:53 EEST

ID користувача:  
100005671

Назва документа: Залуцька\_Записка\_short

Кількість сторінок: 60 Кількість слів: 9049 Кількість символів: 70303 Розмір файлу: 3.04 MB ID файлу: 1011254530

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

## 11.1% Схожість

Найбільша схожість: 6.71% з джерелом з Бібліотеки (ID файлу: 1009631272)

2.96% Джерела з Інтернету

80

Сторінка 62

9.77% Джерела з Бібліотеки

122

Сторінка 62

## 0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

2

Підозріле форматування

20  
сторінок

## РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками

Автор: студентка групи КН-18-1 Залуцька Ольга Олександрівна

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: викладач кафедри КН Молчанова М.О.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<b>відповідає</b>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

*Підтвердження:*

*Запозичення, виявлені в роботі Залуцької О.О., не є плагіатом, оскільки:*

*1. запозичення розміщені в розділі огляду існуючих підходів, що не описують безпосередньо авторську роботу і не стосуються її результатів;*

*2. усі запозичення фрагментарні;*

*3. до запозичень входять фрагменти програмного коду, що не мають авторства і містять поширені конструкції;*

*4. серед запозичень знаходяться загальновідомі терміни та скорочення;*

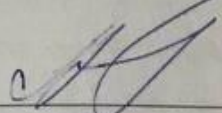
*5. обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості, складає:*

*- за системою Anti-Plagiarism: 2%;*

*- за системою Unicheck: 11.1 %.*

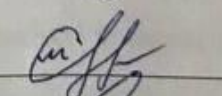
*Отже, запозичення є допустимими та відносяться до описаних вище і адресуються до періоджерел, що, з урахуванням наведених обґрунтувань, свідчить на користь кваліфікаційної роботи.*

Керівник роботи



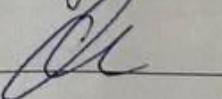
Марина МОЛЧАНОВА

Гарант ОП



Олександр МАЗУРЕЦЬ

Завідувач кафедри КН



Олександр БАРМАК



## РЕЦЕНЗІЯ

### на кваліфікаційну роботу бакалавра

студентки р. КН-18-1 Задуцької Ольги Олександрівни

за темою: Метод автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками

#### 1. Актуальність обраної теми

Дослідження психоемоційного забарвлення цифрових текстів, таких як твіти, дописи у соціально-орієнтованих сервісах, відгуки на товари та послуги, коментарі, тощо представляють великий інтерес в галузях, що безперервно працюють із громадськістю та суспільною думкою. Текстову одиницю можна класифікувати за трьома емоційними категоріями, до якої вона належить: позитивна, негативна та нейтральна. Сегмент IT, а саме галузь NLP досі має ряд невирішених задач, емоційний аналіз текстів – одна з таких.

#### 2. Повнота розкриття мети та завдань роботи

Протягом виконання кваліфікаційної роботи бакалавра мету та завдання роботи було розкрито у повній мірі. Мету та завдання можна вважати розкритими, адже було проведено повний аналіз предметної області, чітко визначено структуру інформаційної системи та реалізовано відповідний програмний продукт.

#### 3. Зміст кожного розділу роботи

У розділах записки кваліфікаційної роботи було описано характеристику предметної області та постановку задачі, проєктування структури інформаційної систем методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками та відображення загальної інформації, а також проєктування методу та бази даних. Розділи також містять опис програмної реалізації інформаційних систем, а саме структуру, особливості реалізації та тестування.

#### 4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблений метод автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками та його програмна реалізація цілком може широко використовуватись у будь-якому колективі для визначення коефіцієнту конфліктності спільноти та соціотипів усіх її членів. Даний метод може бути використана в галузях, що аналізують громадську думку в маркетингових цілях.

#### 5. Якість оформлення кваліфікаційної роботи бакалавра

Записка оформлена якісно, з логічним викладенням матеріалу та наведенням вагомих аргументів. Написання літературно грамотне та доцільне.

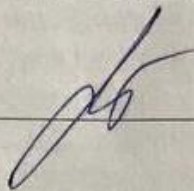
#### 6. Недоліки кваліфікаційної роботи бакалавра

Рекомендовано розширити базу повідомлень для аналізу.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Рецензент \_\_\_\_\_



Т. Говоруненко



## ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу бакалавра

студентки зр. КН-ІС-1 Залуцької Ольги Олександрівни

за темою Метод автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками

### 1. Актуальність теми

Дослідження психоемоційного забарвлення цифрових текстів, таких як твіти, дописи у соціально-орієнтованих сервісах, відгуки на товари та послуги, коментарі, тощо представляють великий інтерес в галузях, що безперервно працюють із громадськістю та суспільною думкою. Текстову одиницю можна класифікувати за трьома емоційними категоріями, до якої вона належить: позитивна, негативна та нейтральна. Сегмент ІТ, а саме галузь NLP досі має ряд невирішених задач, емоційний аналіз текстів – одна з таких.

Для ефективного використання методу було реалізовано програмний застосунок, що за семантичними ознаками класифікує приналежність тексту до певної емоційної категорії. Створення методу та програмного застосунку на його основі є актуальною задачею комп'ютерних наук.

### 2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Відповідно до стандарту спеціальності 122 Комп'ютерні науки, метою роботи студентки є розробки методів і технологій отримання, зберігання, обробки, передачі та використання інформації, інтелектуального аналізу даних і прийняття рішень, а саме розробки методу автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними ознаками. Для досягнення мети роботи використовуються математичні моделі, методи та алгоритми розв'язання теоретичних і прикладних задач, що виникають при розробці інформаційних технологій. Тому результати виконання кваліфікаційної роботи бакалавра відповідають стандарту бакалаври спеціальності 122 – Комп'ютерні науки.

### 3. Професійні та особистісні якості бакалавра

Під час роботи над кваліфікаційною роботою бакалавра Залуцька Ольга Олександрівна проявила себе кваліфікованим та дисциплінованим спеціалістом, вчасно та якісно виконувала поставлені задачі. В процесі створення математично-

Структурально забезпечення поставленої задачі, створенні програмного продукту на основі структурного методу та написання пояснювальної записки проявила достатні для оцінювання успішного результату компетентності та результати навчання. Під час написання опанувала професійні навички за напрямком «Комп'ютерні науки».

#### **4. Ступінь самостійності під час виконання кваліфікаційної роботи**

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував всі поставлені задачі.

#### **5. Ступінь оволодіння методами дослідження**

При реалізації кваліфікаційної роботи показала високий рівень компетентностей та володіння необхідними інструментами та обладнанням, методами, методиками та технологіями предметної області комп'ютерних наук.

#### **6. Повнота та якість розкриття теми роботи**

Тема роботи в повній мірі обґрунтована й розкрита, проведено аналіз актуальності та відомих досліджень в межах обраної теми, поставлені завдання, які у роботі виконані, та розроблено програмне забезпечення для валідації та верифікації запропонованого методу.

#### **7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу**

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

#### **8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин**

Розроблений у роботі метод автоматизованого визначення психоемоційної тональності текстових повідомлень у соціально-орієнтованих сервісах за семантичними оттками та його програмна реалізація може бути використана в галузях, що аналізують громадську думку в маркетингових цілях.

#### **9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота**

Враховуючи високий рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник \_\_\_\_\_

викладач кафедри КН М.О. МОЛЧАНОВА