

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:
«Система оцінювання тональності інформаційного повідомлення»

Виконав: студент 2-го курсу групи 1БС-20м
спеціальності 125 Кібербезпека

_____ Лиськов Д. В.

Керівник: к. т. н., доц. каф. ЗІ

_____ Дудатьєв А. В.

Опонент: к. т. н., проф. каф. ОТ

_____ Азарова А. О.

«____» _____ 2021 р.

Допущено до захисту

Завідувач кафедри ЗІ

д.т.н., проф.

_____ Лужецький В.А.

«____» _____ 2021 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра Захисту інформації
Рівень вищої освіти II-й (магістерський)
Галузь знань 12 Інформаційні технології
Спеціальність 125 Кібербезпека
Освітньо-професійно програма Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ,
д.т.н., проф.
_____ В.А. Лужецький
«___» _____ 2021 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Лиськову Дмитру Васильовичу

1. Тема роботи: «Система оцінювання тональності інформаційного повідомлення», керівник роботи: Дудатьєв Андрій Веніамінович, к. т. н., доцент каф. ЗІ, затверджені наказом ВНТУ від 24 вересня 2021 року №277.
2. Строк подання студентом роботи 18 грудня 2021 р.
3. Вихідні дані до роботи:
 - врахування знаків пунктуації не менше чотирьох;
 - вага в діапазоні від 1 до 5;
 - тональність в діапазоні від -5 до 5;
 - можливість завантаження тексту з файлу;
 - мови визначення тональності: російська, українська та суржик.
4. Зміст текстової частини: Вступ. 1. Аналіз проблеми визначення тональності текстових повідомлень. 2. Розробка моделі системи. 3. Розробка програмного засобу. 4. Тестування системи виявлення та протидії. 5. Економічна частина. Висновки. Перелік використаних джерел. Додатки.
5. Перелік ілюстративного матеріалу: Структурна схема роботи системи (плакат, А4) Схема алгоритму роботи користувача із програмою (плакат, А4). Структурна схема роботи програми (плакат, А4). Структурна схема роботи програми (плакат, А4). Функціональна схема роботи програми (плакат, А4). Структура бази даних (плакат, А4). Фрагменти проведення тестування (плакат, А4).

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Дудатьєв А. В., к. т. н., доц. каф. ЗІ		
2	Дудатьєв А. В., к. т. н., доц. каф. ЗІ		
3	Дудатьєв А. В., к. т. н., доц. каф. ЗІ		
4	Дудатьєв А. В., к. т. н., доц. каф. ЗІ		
5	Лесько О. Й., проф., зав. каф ЕПВМ		

7. Дата видачі завдання 9 вересня 2021 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	09.09.2021 – 14.09.2021	
2	Розробка технічного завдання	15.09.2021 – 20.09.2021	
3	Аналіз літературних джерел за напрямком магістерської кваліфікаційної роботи	21.09.2021 – 29.09.2021	
4	Розробка рішень, алгоритмів	30.09.2021 – 11.10.2021	
5	Практична реалізація, моделювання, тестування, результати	12.10.2021 – 07.11.2021	
6	Розробка розділу економічного обґрунтування доцільності розробки	08.11.2021 – 12.11.2021	
7	Аналіз виконання ТЗ, висновки	14.11.2021 – 21.11.2021	
8	Оформлення пояснювальної записки	22.11.2021 – 29.11.2021	
9	Попередній захист МКР	30.11.2021 – 03.12.2021	
10	Представлення МКР до захисту, рецензування	06.12.2021 – 16.12.2021	
11	Захист МКР	21.12.2021 – 23.12.2021	

Студент _____ Д. В. Лиськов
(підпис)

Керівник роботи _____ А. В. Дудатьєв
(підпис)

АНОТАЦІЯ

УДК 681.325.5

Лиськов Д. В. Система оцінювання тональності інформаційного повідомлення. Магістерська кваліфікаційна робота зі спеціальності 125 – Кібербезпека, освітня програма – Безпека інформаційних і комунікаційних систем. Вінниця: ВНТУ, 2021. 109 с.

На укр. мові. Бібліогр.: 25 назв; рис.: 27; табл. 6.

Магістерська кваліфікаційна робота присвячена розробці системи оцінювання тональності інформаційного повідомлення. Розглянуто існуючі підходи до аналізу текстів, зокрема в частині визначення їх тональності. Для подальшої реалізації за основу взято лексемний метод, який запропоновано удосконалити шляхом розгляду не тільки окремих слів, а й словосполучень із двох слів. Також запропоновано для кожної лексеми, окрім ступеня позитивності/негативності, розглядати її вагу, що уособлює достовірність застосування даної лексеми. В результаті створено систему, що опрацьовує інформаційні повідомлення, а аналізує ступень деструктивності та пропонує методи протидії. Проведене тестування показало достатню ефективність роботи створеної системи.

Графічна частина складається з 6 плакатів з демонстрацією результатів моделювання і проведених досліджень.

В економічному розділі оцінено витрати на розробку.

Ключові слова: програмний засіб, тональність повідомлення.

ABSTRACT

Lyskov D. V. Information message tone assessment system. Master's thesis in specialty 125 – Cybersecurity. Vinnytsia: VNTU, 2021. – 109 p.

In Ukrainian language. Bibliographer: 25 titles; fig.: 27; tabl.: 6.

Master's thesis is devoted to developing of a system for assessing the tone of an information message. The existing approaches to the analysis of texts are considered, in particular in terms of determining their key. For further implementation, the lexical method is taken as a basis, which is proposed to be improved by considering not only individual words, but also phrases of two words. It is also proposed to consider for each token, in addition to the degree of positivity/negativity, its weight, which represents the reliability of the use of this token. As a result, a system has been created that processes information messages, analyzes the degree of destructiveness and suggests methods of counteraction. The conducted testing showed sufficient efficiency of the created system.

The graphical part consists of 6 posters with simulation results.

The economic part estimates the development costs.

Keywords: software, message tone.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРОБЛЕМИ ОЦІНЮВАННЯ ТОНАЛЬНОСТІ ІНФОРМАЦІЙНИХ ПОВІДОМЛЕНЬ.....	8
1.1 Загальні питання галузі обробки природних текстів (NLP)	9
1.2 Аналіз проблеми дослідження	16
1.3 Уточнена постановка задачі дослідження	20
2 РОЗРОБКА МОДЕЛІ СИТЕМИ.....	22
2.1 Структура системи	22
2.2 Розробка критеріїв для визначення тональності тексту.....	33
2.3 Обґрунтування методу вирішення багатокритеріальної задачі	35
2.4 Розробка алгоритмів	38
3 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ.....	47
3.1 Вибір технології розробки програмного забезпечення.....	47
3.2 Вибір засобів розробки	50
3.3 Проектування інтерфейсу користувача програмного забезпечення.....	58
3.4 Розробка бази даних для системи аналізу тональності текстових повідомлень	62
3.5 Особливості програмної реалізації окремих елементів застосованих алгоритмів.....	64
3.6 Розробка документації щодо експлуатації програмного засобу	66
4 ТЕСТУВАННЯ СИТЕМИ.....	69
4.1 Формування деструктивного повідомлення.....	69
4.2 Обробка виявленого деструктивного повідомлення	71
4.3 Протидія впливу	74
5 ЕКОНОМІЧНА ЧАСТИНА.....	77
5.1 Оцінювання комерційного потенціалу розробки (технологічний аудит розробки).....	77
5.2 Прогнозування витрат на виконання науково-дослідної та конструкторсько-технологічної роботи.....	80
5.3 Прогнозування комерційних ефектів від реалізації результатів розробки.....	84
5.4 Розрахунок ефективності вкладених інвестицій та період їх окупності... ..	86
ВИСНОВКИ	90
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	91
ДОДАТКИ.....	94
Додаток А. Технічне завдання	95
Додаток Б. Схема роботи програми.....	99
Додаток В. Хід встановлення ODBC	100
Додаток Г. Текст програми	103
Додаток Д. Протокол перевірки на наявність плагіату	109

ВСТУП

На сьогодні є актуальною галузь, що займається обробкою інформації на предмет її емоційної нахиленості. Так як, суспільство постійно контактує з новою інформацією з різноманітних джерел, важливим є питання фільтрації даної інформації. Одним з інструментів, які можуть використовуватись для даних цілей, є оцінка емоційного забарвлення. Цим направлення займається така складна та багатогранна дисципліна, як обробка природних текстів (англ. Natural Language Processing – NLP), яка об'єднує декілька галузей: інформаційні технології (IT), лінгвістика та штучний інтелект (англ. Artificial Intelligence – AI) [1].

Очевидно, ефективна діяльність щодо протидії деструктивному впливу в інформаційних протиборствах повинна здійснюватися на різних рівнях: як на рівні центрального телебачення та радіовіщання, так і на рівні повідомлень у групах соціальних мереж. І тут виникає проблема нестачі людських ресурсів, що фізично не можуть забезпечувати моніторинг усіх повідомлень в усіх групах усіх соціальних мереж, каналів Інтернет-месенджерів, коментарях на веб-сайтах під «гострими» новинами, і т.п. Однак, безперечно, дана діяльність має виконуватися, аби не пропустити організацію небезпечної події (повстання, терористичного акту, заколоту, витоку важливої/секретної інформації тощо), що може, іноді, і спонтанно виникнути, як реакція на певну новину, повідомлення, чийсь необдуманий вислів, і т.п. Відповідно, **актуальною** є задача автоматизованого аналізу текстових повідомлень, зокрема на предмет їх тональності.

Навіть окремо виділені задачі, що вирішуються у даній галузі, мають високий рівень складності, причому як алгоритмічної, так і обчислювальної [2]. Саме тому в рамках даної роботи і буде розглядатися задача визначення тональності повідомлення, без поглиблення у його семантичний зміст.

Об'єктом дослідження є процеси визначення тональності інформаційного повідомлення та протидії інформаційному впливу.

Предметом дослідження є методи та засоби, що дозволяють здійснювати визначення тональності інформаційного повідомлення та протидії впливу.

Метою даної роботи є автоматизація процесу аналізу повідомлень, а також пришвидшення процесу протидії на джерела інформації з деструктивним впливом, що досягається шляхом розробки комплексної системи для оцінювання тональності інформаційних повідомлень.

Для досягнення поставленої мети слід виконати наступні задачі:

- проаналізувати наявні у відкритому доступі науково-технічні джерела для встановлення особливостей існуючих способів визначення тональності інформаційних повідомлень та протидії впливам;

- обрати базовий спосіб визначення тональності інформаційних повідомлень;

- розробити детальні наочні представлення алгоритмічних складових обраного способу визначення тональності та методів протидії, проаналізувати їх на предмет можливості удосконалення та внести відповідні покращення у обраний спосіб;

- обрати оптимальні для даного випадку технології та засоби розробки та здійснити програмну реалізацію обраного способу;

- здійснити тестування розробленої системи;

- оцінити перспективи розвитку дослідження та зробити висновки по роботі.

Практичне цінність роботи полягає у створенні системи, яка може використовуватися, як допоміжний засіб у реальній діяльності відповідних служб по забезпеченню інформаційної безпеки України, а також, як додатковий елемент систем захисту в державних установах та на приватних підприємствах.

Наукова новизна магістерської роботи полягає в тому, що удосконалено лексемний метод оцінювання тональності, який на відміну від базового підходу відрізняється тим, що окрім розгляду кожного слова окремо, розглядається також наявність словосполучень з двох слів, що вказують на той чи інший рівень тональності.

Проміжні результати роботи доповідались на XLIX Науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії (2020).

В перспективі програма може розвиватися шляхом розширення переліку слів, що присутні у базі даних, яку вона використовує.

1 АНАЛІЗ ПРОБЛЕМИ ОЦІНЮВАННЯ ТОНАЛЬНОСТІ ІНФОРМАЦІЙНИХ ПОВІДОМЛЕНЬ

Як було зазначено у вступі, проблема визначення тональності інформаційних повідомлень відноситься до галузі NLP, оскільки мова йде виключно про текстові повідомлення (тобто такі, що складаються із друкованих літер, набраних на клавіатурі персонального комп'ютеру (ПК)). Звичайно, у деяких соцмережах та на окремих сайтах дозволяється розміщення і аудіо повідомлень, однак задача визначення їх тональності містить ще один додатковий етап технічного характеру, який слід виконувати перед власне визначенням тональності цього повідомлення, а саме, розпізнавання мови: із аудіо сигналу слід отримати текст, що у цьому сигналі присутній. Нажаль, і ця задача є дуже-дуже складною, містить надзвичайно багато аспектів, і являє собою цілком окрему проблему галузі ІТ, тому у даній роботі аналіз аудіо повідомлень виконуватися не буде. В подальшому під словами «інформаційне повідомлення» будемо мати на увазі виключно текстове повідомлення, тобто таке, що має наступні властивості:

- представлене послідовністю символів, основну частину яких становлять літери української абетки;

- крім букв, повідомлення може містити знаки пунктуації, числа та деякі спецсимволи, що несуть додатковий семантичний зміст, чи здійснюють емоційне забарвлення, чи іншу додаткову, в т.ч. невербальну функцію.

Інформаційні повідомлення можуть мати різне походження, що є важливим для визначення осіб, які несуть відповідальність (можливо часткову) за виникнення, оприлюднення та поширення цього повідомлення. Так повідомлення може бути:

- набране на клавіатурі ПК людиною-користувачем безпосередньо у формі (веб-формі), що надається тим ресурсом, де воно опубліковане;

- скопійоване з іншого першоджерела (з його зазначенням, або без);

– згенероване спеціальною комп'ютерною програмою, що працює відповідно до алгоритму та налаштувань, які закладені у неї особою-розробником (наприклад, програма-бот).

Вказані особливості, звичайно, несуть важливий зміст, однак не мають відношення до технічної складової процесу оцінки тональності тексту. Для цього використовуються методи із галузі NLP (нажаль, ця аббревіатура співпадає з позначенням нейролінгвістичного програмування, однак, оскільки галузі психології та ІТ мало перетинаються, зазвичай це не викликає труднощів і з контексту завжди зрозуміло, про яке саме NLP йде мова). Розглянемо особливості цієї дисципліни більш детально.

1.1 Загальні питання галузі обробки природних текстів (NLP)

Природна мова, якою користуються люди у повсякденному житті являє собою досить складну систему, у якій часто допускається неоднозначне трактування багатьох таких елементів, як речення, виразу, словосполучення чи навіть окремі слова. В першу чергу це стосується слів омонімів, що пишуться абсолютно однаково, але зміст несуть цілком різний. Відповідно, для того, щоби «зрозуміти», який саме варіант із доступних для даного омоніму, мається на увазі, слід орієнтуватися у змісті всього речення чи навіть тексту.

Також характерною рисою природної мови є її надзвичайно широка варіативність, яка реалізується у можливості виразити абсолютно різними, не схожими одне на одне словами одне і те ж саме поняття, тезу, ситуацію, сукупність фактів, і т.п. І чим більший обсяг інформації слід передати словами, тим більше існує варіантів реалізації такої передачі. В цілому для кожного слова східноєвропейської мовної сім'ї можна підібрати не менше 2-3 синонімів (а часто і набагато більше), таким чином, речення, що складається з

10 слів допускає $3^{10} \approx 60$ тис. різних варіантів запису, а, якщо говорити про невеликий текст обсягом 10 речень, то кількість варіантів сягатиме числа

«космічних» розмірів $(6 \cdot 10^4)^{10} = 6 \cdot 10^{47}$. Навчити комп'ютер розпізнавати усі ці комбінації слів, причому так, щоб вони були рівними одному і тому ж підсумковому результату, є складним, а точніше надзвичайно складним завданням галузі NLP.

Вказані питання прямим чином відносяться до семантичних особливостей людської мови (тобто розпізнавання її змісту, суті, сенсу), а вирішувати їх якраз допомагають особливості морфології тексту, його синтаксису, що уособлюються у наявності допоміжних службових слів, знаків пунктуації, уточнюючих слів, і т.п.

Відзначимо, що семантичний зміст тексту є головним елементом в обох існуючих напрямках застосування NLP, а саме, таких як:

– аналіз комп'ютерною системою отриманого із зовнішнього середовища природного тексту, початково призначеного для сприйняття людиною (причому форма тексту має бути символною, за допомогою літер відповідного алфавіту), в результаті чого система повинна бути здатною використовувати інформацію, закладену у тексті, тобто фактично має «зрозуміти» його;

– генерація комп'ютерною системою природного тексту, який відповідає усім вимогам до синтаксису обраної цільової мови, достатньої повноти лексики (тобто є таким, який би згенерувала доросла освічена людина-носій), і який інкапсулює певну змістовну інформацію.

Обидві задачі є надзвичайно складними, і в комплексі являють собою чи не найвідоміший тест для оцінки якості штучного інтелекту (тест Тюрінга). Запропонований у 1950 р., він полягає у спробі автоматизованої системи (комп'ютерної програми) вести осмислену розмову, причому так, щоб проінформована та підготовлена людина-співрозмовник не могла зробити висновок, хто ж саме з нею спілкується: комп'ютерна програма чи інша людина [3].

Наявність безперервного обговорення та, тим більше, стабільна популярність тесту Тюрінга серед професіоналів з IT, AI та математичної

лінгвістики доводить актуальність теми створення систем аналізу (з одного боку) та синтезу (з другого) природних текстів.

Таким чином, підсумовуючи наведену загальну інформацію про особливості обробки природної мови, можна сказати, що найголовнішу роль у ній, мету всього процесу складає усвідомлення інформації такою сутністю, як комп'ютерна програма. Вона в ідеальному випадку має «зрозуміти» семантику тексту, зокрема, вміти користуватися у майбутньому тією інформацією, що закладена у тексті, робити на її основі логічні висновки, будувати припущення та ін. Здійснити це «усвідомлення» допоможуть різні підвиди лінгвістичного аналізу тексту, що детально будуть розглянуті у наступному підрозділі. В нашому випадку достатньо оцінити загальну тональність повідомлення (як саме це виконувати, розглянемо в подальшому викладенні).

Окрім питань семантики, дещо менш важливу (але не останню) роль грають проблеми синтаксичної побудови текстів, тобто правила їх запису, яких слід дотримуватися щоб текст вважався правильним і коректним. Сюди відносять правильну розстановку розділових знаків, узгодження роду відповідних частин речення, дотримання правильних відмінків у реченнях, і т.п. Синтаксичний аналіз являє собою одну з двох частин граматики природних мов, в яку також входить морфологія відповідної мови (рис. 1.1).

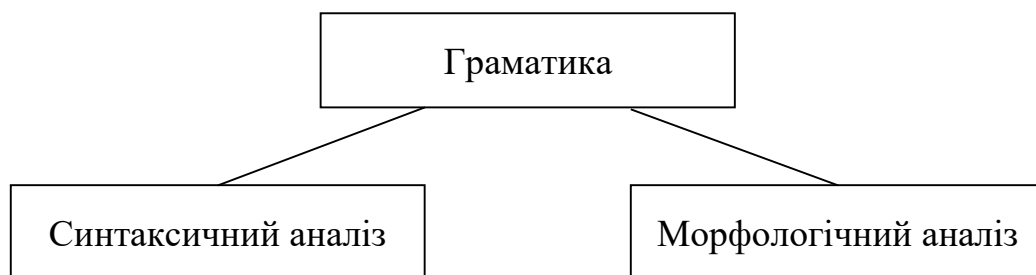


Рисунок 1.1 – Склад граматики природної мови.

Ще одним аспектом процесу аналізу текстів є дослідження їх морфологічних особливостей. Морфологія, це розділ лінгвістики, що вивчає форму елементів тексту: окремих слів, а також і речень. Морфологічний розбір слова в цілому більш поширений (ніж аналіз форми та складових речень) у

лінгвістичних дослідженнях і представляє інтерес для визначення його семантичних особливостей, етимології, історії розвитку та/або поступових перетворень. Для створення цілісної картини про великий текст морфологічний розбір слова має невелике значення. Набагато більш доцільно проводити морфологічний аналіз речень, як більш укрупнених одиниць тексту. У речень, як і слів, як і інших елементів текстової інформації, можна виділяти деякі формалізовані характеристики їхньої форми, що можуть бути корисними при класифікації текстів. Елементарним прикладом тут може бути такий простий показник, як кількість слів у реченні: якщо їх число мале, таке речення з більшою імовірністю відноситься до негативних, оскільки позитивні речення частіше несуть додаткові члени речення, що являють собою різні художні засоби на зразок метафор. В той же час, переважання речень великої довжини може свідчити про нейтральний характер тексту (текст науково-технічного, методичного характеру).

Таким чином, морфологічний аналіз, наряду з синтаксичним, є суттєвими елементами NLP, що становлять суттєву допомогу у виконанні семантичного аналізу, який, в свою чергу є вершиною галузі обробки природного тексту.

Якщо, розвиваючи цю тему, вказати інші розділи лінгвістики, що використовуються при аналізі текстів, то можна додати наступні:

- фонетику;
- лексику;
- фразеологію;
- орфографію.

Дамо коротку характеристику цим розділам науки мовознавства в контексті можливості їх застосування для визначення тональності текстових повідомлень.

Правила фонетики визначають звучання того, чи іншого слова, і, взагалі кажучи, не є важливими для цілей даної роботи, оскільки весь процес аналізу тексту спирається на його друкований варіант (а не усний!). Фонетичні

особливості мови повинні братися до уваги розробниками синтезаторів людської мови.

Крім фонетичного, також може розглядатися лексичний аналіз, сутність якого є набагато більш близькою до цілей даної роботи. Лексичний аналіз ставить на меті збір та аналіз лексики тексту, тобто набору слів, які у ньому присутні. Деякі слова можуть повторюватися багато разів, що спричиняє певне забарвлення тексту, або взагалі визначає його предметну галузь. Принципова наявність (навіть і без повторюваності) деяких слів (специфічних, термінів) у тексті уже дозволяє однозначно визначити його напрямок, наприклад, наявність слова «лапочка» однозначно класифікує текст як позитивний, а «скотиняка» – навпаки.

Так само, як за наявністю, так і за відсутністю певних слів можна судити про стилістичну приналежність тексту. Наприклад, якщо у великому тексті відсутні слова, що забезпечують емоційне забарвлення (такі, як «люблю», «прекрасний», «чудовий», «гарний», і т.п.), то ймовірніше всього він має нейтральне забарвлення, не може бути позитивним.

Таким чином, лексичний аналіз тексту повинен урахуватися у даній роботі наряду з синтаксичним та морфологічним (для речень), що і буде впроваджено у подальшій програмній реалізації.

Під фразеологією мається на увазі розділ мовознавства, який вивчає стійкі фрази, фразеологізми, складними прикладами яких є прислів'я, промовки і т.п. У сучасній мові це можуть бути різноманітні Інтернет-меми, які повторюються мільйонами користувачів і т.д. Виконувати фразеологічний аналіз для встановлення тональності, а тим більше – семантики тексту на думку автора немає жодного сенсу, тому в подальшому викладі він не розглядається.

Нарешті, орфографічний аналіз має на меті порівняння написання слів тексту з їх еталонними варіантами з урахуванням різноманітних правил, які супроводжують відмінювання слова за відмінками, родами і т.д. Слід відмітити, що комп'ютерні програми, що виставлялися у різні роки для проходження згаданого вище тесту Тюрінга, досить часто імітували певні орфографічні

помилки, щоб їх «мова» виглядала більш людяно. Наявність орфографічних помилок у тексті, що аналізується програмою, значно затрудняє розуміння тональності всієї інформації, однак ця ж проблема виникає при спробі людини зрозуміти деякі надписи-коментарі в соціальних мережах, які залишають користувачі із низьким рівнем знання відповідної мови. Все це спричинює висновок, що аналіз орфографії, так само, як і її контроль при генерації текстів, не може бути «наріжним каменем» всього процесу лінгвістичного аналізу текстів. Насправді, правила орфографії при генерації текстів мають виконуватися, а вхідні тексти повинні бути записані на «правильній» літературній мові, принаймні програма має передбачати саме такий варіант. Зважаючи на це, орфографічний аналіз у подальшому виконувати не будемо.

Підсумовуючи, можна сказати, що для цілей даної роботи можна використовувати наступні види аналізу текстів:

- синтаксичний;
- лексичний;
- морфологічний (на рівні дослідження форми і будови окремих речень).

Синтаксичний розбір, в першу чергу, відноситься до такої одиниці тексту, як речення, хоча може застосовуватися і до його частин (наприклад, словосполучень). Синтаксичний розбір речення може виконуватися за процедурою, виробленою у лінгвістиці, яку розглянемо по крокам:

- 1) Охарактеризувати речення за метою висловлювання: розповідне, питальне або спонукальне;
- 2) Встановити емоційне забарвлення речення: окличне або неокличне;
- 3) За наявністю граматичних основ визначити, чи є речення простим або складним.

Якщо речення просте, слід виконати наступні кроки:

- 4а) За наявністю головних членів речення слід встановити, чи воно односкладне (наявний тільки підмет, або тільки присудок), чи двоскладне (обидва головних члени речення присутні);

5а) Встановити наявність другорядних членів речення та, відповідно, тип речення: поширене, або непоширене;

6а) Вказати, чи ускладнене речення (однорідними членами, зверненням, ввідні слова, і т.п.);

7а) Вказати частини мови речення та, можливо, скласти схему речення.

Якщо речення є складеним, кроки мають бути такими:

4б) Вказати тип зв'язку частин речення: за допомогою сполучників чи без них.

5б) Вказати засіб зв'язку частин речення: інтонація, сурядні чи підрядні союзи.

6б) На основі 4б) та 5б) визначити тип речення: безсполучникове, складносурядне, складнопідрядне;

7б) Розібрати кожну частину складного речення за алгоритмом аналізу простого речення, тобто за пунктами 4а)-7а).

Проведення синтаксичного розбору у такій формі дозволяє визначити деякі другорядні особливості тексту, однак у не дуже великій мірі сприяє його розумінню на семантичному рівні, крім, звичайно, пункту 7а). Дійсно, встановити суть інформації, яка виражається певним реченням, можна лише розібравши лексичне значення кожного слова, наявного у реченні (лексичний аналіз), а потім встановивши зв'язки між цими словами, тобто виконавши синтаксичний аналіз за пунктом 7а).

Відповідно, лінгвістичною основою подальшої програмної реалізації доцільно зробити лексичний та синтаксичний аналіз, доповнений елементами морфологічного. Відзначимо, що застосування даних типів аналізу тексту може виконуватися математично різними методами, серед яких можна назвати:

- лексемний метод;
- наївний класифікатор Баеса;
- метод опорних векторів;
- логістична регресія;
- штучні нейронні мережі.

У подальшому розглянемо ці методи більш детально.

1.2 Аналіз проблеми дослідження

Лінгвістичний аналіз текстів та його частинні методи можуть бути корисними для різноманітних предметних галузей застосування. Це і наукові дослідження на зразок пошуку спільних мов-предків, або прогнозування динаміки розвитку природних мов. До нього можна віднести і галузь людино-машинної взаємодії (НСІ – Human-Computer Interface), що дуже динамічно розвивається при взаємодії ІТ, лінгвістики, математики та електроніки. І, звичайно, сюди відноситься автоматизований аналіз текстів для отримання їх чисто утилітарних характеристик. Зокрема, в ідеальному випадку комп'ютерна програма (або апаратна система, яка в кінці-кінців все одно працює під управлінням певного програмного забезпечення) має визначати сенс, або зміст інформаційних повідомлень, що циркулюють в електронному вигляді, зокрема і у мережі Інтернет [4]. Під повідомленнями тут маються на увазі наступні тексти:

- коментарі, що залишають користувачі у соціальних мережах, висловлюючи власну думку про ту, чи іншу початкову інформацію, а також і про коментарі інших користувачів;
- повідомлення програм електронних месенджерів (на зразок WhatsApp, Viber, і т.д.);
- повідомлення SMS, що проходять через GSM-мережу країни;
- публіцистичні статті різних тематик;
- аналітично-оглядові матеріали;
- відомості пізнавального та світоутворюючого характеру, особливо такі, до творення яких мають доступ будь-які бажаючі користувачі глобальної мережі Інтернет (як, наприклад, ресурси Вікіпедії може корегувати будь-хто, через що серйозні науковці скептично ставляться до посилань на цей та аналогічні ресурси, що насправді створюються усім світом, спільнотою);

– та подібні, в яких допустимим є висловлення користувачами власних думок, вражень, прагнень та закликів.

Очевидно, що в описаних повідомленнях можуть розміщуватися заклики до скоєння протиправних та навіть протизаконних (кримінальних) дій (не говорячи вже про такі «дрібниці» як образи інших осіб, в т.ч. за національними або расовими ознаками та ін.). Якщо говорити про публікації в електронних ЗМІ, на офіційних сторінках підприємств та організацій, а також і на офіційних (підтверджених) сторінках відомих в Україні фізичних осіб, то вони підлягають безперервному моніторингу з боку відповідних підрозділів силових структур України, який традиційно (ще з часів паперової преси) здійснюється у ручному режимі. тобто шляхом читання цих текстів співробітниками-аналітиками (в першу чергу, Служби безпеки України, а також Національної поліції України). Очевидно, що штат таких підрозділів не може збільшуватися до нескінченості, тому зрозуміти надзвичайну важливість виконання даного дослідження, яке присвячене автоматизації цих робіт, можна з наступних міркувань.

За дослідженнями швидкість сприйняття інформації для дорослої тренованої людини складає 200 слів на хвилину при задовільному ступені розуміння матеріалу [5]:

$$v = 200 \text{ (слів/хв.)}.$$

У кількох популярних соціальних мережах наявні сторінки різних структур та організацій, а також високих посадовців України, що називають «офіційними». Усі вони активно коментуються, так само, як і не менше десятка популярних новинних сайтів, що мають щодня по кілька мільйонів переглядів, як,

наприклад, unian.net. На кожен публікацію тут простими користувачами мережі виконується коментування, причому число таких підписів сягає сотень (а під деякими, особливо злободенними питаннями – до десятка тисяч і більше, як наприклад, на «офіційній» сторінці Президента України у мережі Facebook, яка розміщена за адресою <https://www.facebook.com/groups/zeprezedent/>). Для

конкретності приймемо кількість таких сторінок в українському сегменті мережі Інтернет, де щоденно активно виконуються публікації коментарів, рівним 100 (і це ще занижена оцінка):

$$N_{стор} = 100 \text{ (шт. або точніше стор.)}$$

На кожній такій сторінці протягом дня викладаються десятки статей (новин, публікацій), з яких лише певна частина може спричинити викладання коментаря, важливого для цілей оперативно-пошукової діяльності. В основному такі публікації присвячені злободенним питанням, як, наприклад, військовий конфлікт на Сході України, особливості зовнішньої політики України, корупція у вищих органах влади держави, розслідування резонансних злочинів, соціальні погіршення, і т.п. Будемо вважати, що число таких публікацій на кожному вищеописаному ресурсі складає 5 публікацій на добу:

$$N_{публ} = 5 \text{ (публ./стор. за добу)}$$

Щоб оцінити кількість коментарів під кожною суттєвою публікацією, що викликає інтерес у широких мас, можна керуватися наступними оцінками: злободенна новина не може викликати менше 100 коментарів, а з іншого боку, коли користувач заходить на сторінку, де уже зроблено кілька тисяч коментарів, то мотивація сказати щось самому сильно зменшується через очевидне розуміння, що при такій кількості уже наявних коментарів, навряд чи він зможе сказати щось нове, а головне, навряд чи хтось взагалі дійде у читанні коментарів до двохтисячного чи навіть більше. Отже, процес коментування публікацій, що мають властивість швидкого застарівання (на відміну, наприклад, від відеокліпів чи фільмів на YouTube, що можуть мати стабільні інтенсивності коментування протягом місяців і навіть років), загасає сам собою, коли нові читачі, бачачи велику кількість уже зроблених коментарів, відмовляються від висловлення власної думки. Таким чином, число коментарів розумно розглядати у діапазоні від 200 до 2000, і для конкретності приймаємо його на рівні 200:

$$N_{ком} = 200 \text{ (ком./публ.)}.$$

Число слів у кожному коментарі суттєво різниться, починаючи від 1-2 (якими може виражатися проста хуліганська позиція користувача, або навпаки – виражатися підтримка певній позиції) і завершуючи кількома сотнями. Однак, дуже короткі коментарі аналізувати взагалі-то і не потрібно, оскільки у них просто не вистачає місця (з математичних позицій теорії інформації), аби містити якісь важливі відомості фактичного характеру, або такі, що можуть допомогти у пошуку злочинців, і т.п. Отже, середнє число слів у коментарі приймаємо рівним 10 (пам'ятаємо про виконання мінімальної оцінки):

$$N_{слів} = 10 \text{ (слів/ком.)}.$$

Таким чином, можна оцінити кількість людино-годин, необхідних для читання та аналізу усіх коментарів, що з'являються щодня в українському сегменті мережі Інтернет:

$$T = \frac{N_{стор} \cdot N_{публ} \cdot N_{ком} \cdot N_{слів}}{\nu} = \frac{100 \cdot 5 \cdot 200 \cdot 10}{200} = 5000 \text{ хв./добу} = 84 \text{ год/добу}$$

Як бачимо із виконаної оцінки, на одну добу слід виділити 84 людино-годин, щоби перевірити усі коментарі, які виконуються користувачами на популярних веб-ресурсах. Вважаючи, що тривалість однієї зміни складає 6 годин (враховуючи перерви, які слід проводити відповідно до норм охорони праці для усіх осіб, що працюють за ПК), а на тиждень у операторів повинно бути 2 вихідних, то відповідна кількість штатних одиниць складе досить велику цифру у 25 працівників:

$$H = T / 6 \cdot (7 / 5) = 84 / 8 \approx 25 \text{ людей.}$$

Нагадаємо, що оцінка виконувалася по нижній границі, а реальна кількість людей для аналізу усіх добових коментарів на найбільш популярних ресурсах країни буде у декілька разів більшою (тобто порядку сотні працівників, що займатимуться виключно читанням та аналізом коментарів). Очевидно, що такі задачі насправді мають вирішуватися автоматизованими системами, що

переглядають великі обсяги матеріалу в автоматичному режимі, а до допомоги людини звертаються тільки якщо повідомлення було визначене автоматичною системою як небезпечне.

Таким чином, бачимо, що дійсно існує проблема автоматизації визначення тональності інформаційних повідомлень, яка може бути вирішена шляхом розробки та впровадження відповідного програмного продукту. Беручи до уваги викладену вище інформацію, можна сформулювати комплекс вимог до такого продукту, записаний у наступному підрозділі. Також можемо констатувати, що на місці проблеми автоматизації визначення тональності інформаційних повідомлень виникає проблема технічної реалізації відповідного програмного продукту, яка буде вирішуватися у наступному розділі.

1.3 Уточнена постановка задачі дослідження

Таким чином, розробці підлягає система, яка повинна відповідати наступним вимогам:

- вміти здійснювати диференціацію текстів усіх розмірів (крім дуже малого), або, приймаючи точно: із обсягом більше 10 слів;
- система має визначати точну числову оцінку, яку можна використати для взаємного дихотомічного порівняння текстів (тобто парного, один з іншим);
- алгоритм роботи системи має бути не тільки ефективним, як зафіксовано у попередніх вимогах, а й швидким, тобто базуватися на обчислювально простих математичних залежностях;
- система повинна базуватися на проведенні лексичного аналізу, тобто використовувати словник (словники) типових сигнальних слів, за наявності (або відсутності) яких у тексті можна формувати його числову оцінку, описану вище;
- система повинна надавати рекомендації для протидії деструктивному впливу, який характеризує оцінка, отримана на попередньому кроці;

– у якості місця зберігання словників слід використати високопродуктивне сховище, а саме одну із доступних на сьогоднішній день систем управління базами даних (СУБД);

– система має бути розроблена для використання як компонент для систем захисту, як у державних установах так на підприємствах приватного типу власності.

Керуючись цими вимогами, можна переходити до конкретних проектних рішень та розгляду особливостей реалізації системи оцінювання тональності інформаційних повідомлень, що буде виконуватися у наступних розділах.

Таким чином, у даному розділі детально розглянуто різні аспекти предметної галузі, пов'язаної з деякими спеціальними аспектами лінгвістичного аналізу текстів. Обґрунтована нагальна необхідність проведення автоматизованої оцінки текстових повідомлень, для чого, в першу чергу, запропоновано проведення лексичного аналізу текстів, ймовірно, із окремими елементами синтаксичного та морфологічного. Кінцево, сформульовано детальні вимоги до системи, за допомогою якого можна оцінювати тональності інформаційних повідомлень та протидіяти впливам, при їх виявленні. Ці відомості поставлені в основу подальшого дослідження в межах даного роботи.

2 РОЗРОБКА МОДЕЛІ СИТЕМИ

2.1 Структура системи

Перш ніж почати розробку системи необхідно встановити основні елементи з яких вона буде складатись. Для будь-якого впливу повинно існувати джерело з якого власне буде здійснюватись вплив на систему. На наступному кроці система повинна виявити та обробити надходження тієї чи іншої інформації з різних вхідних джерел. Так, як виявлення та обробка, це процеси які виконуються одночасно тому, дані елементи можна об'єднати в одну підсистему виявлення. Після надходження впливу з джерела та його обробки підсистемою виявлення, необхідно обробити характер впливу та сформувати відповідну реакцію на даний вплив. Для формування та реалізації відповіді на виявлений вплив, суттєвим кроком є створення підсистеми протидії. На основі вищезазначеного створено структуру системи, яка відображена на рисунку 2.1.

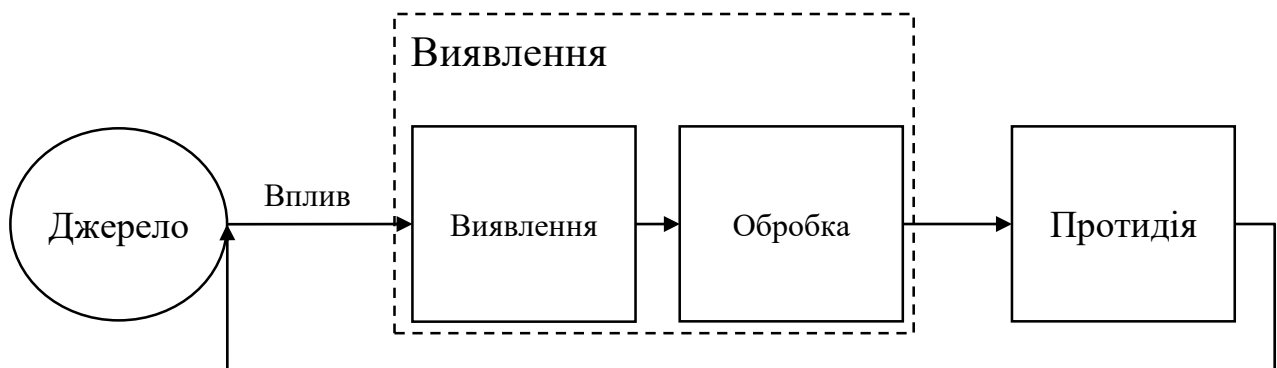


Рисунок 2.1 – Структура елементів системи

Дана структура системи була створена опираючись роботу, в якій запропоновано розглядати наступні основні частини інформаційних війн [6]:

1. Проведення аналізу інформаційного простору. Ця частина передбачає збір, обробку та обмін про потенційних супротивників і союзників з метою ведення подальших активних дій. На цьому етапі зі сторони противника проектування нових інформаційних впливів-мемів з метою їх подальшого поширення.

2. Проведення інформаційного впливу. Даний етап передбачає поширення і впровадження підготовленої спеціальної інформації в інформаційне поле противника, а також припинення спроб противника отримати потрібну йому інформацію.
3. Супровід або коментування інформаційного повідомлення. Даний етап передбачає врахування специфіки джерел поширення інформаційного повідомлення-мема, специфіку соціальної частини соціотехнічної системи (СТС), на яку дана інформація поширюється. Іншими словами, на цьому етапі практично реалізується першість і забезпечується ефективність трактування поширюваної інформації.
4. Інформаційна протидія або захист. Етап передбачає блокування і компенсацію деструктивного інформаційного впливу, який поширює і впроваджує потенційний конкурент або противник.

Для подальшої розробки системи необхідно розглянути підходи щодо реалізації підсистем, що складають загальну структуру системи. Тому в наступних підрозділах розглянемо основні підходи до процесу визначення тональності інформаційних повідомлень та методи протидії впливам, що надходять з різних інформаційних джерел.

2.1.1 Аналіз основних методів

На сьогоднішній день існує чимало методів визначення тональності інформаційних повідомлень. Відзначимо, що саме поняття тональності традиційно означає оцінку по одновимірній шкалі емоційного забарвлення, на якій, як і завжди, виділяється два напрямки: позитивний (добре повідомлення) та негативний (погане). Якщо на даній шкалі проставити певні прийнятні числові мітки та задати діапазони визначення тональності, то будемо мати необхідну для проведення оцінювання текстів систему вимірювань. Тож розглянемо характеристики цих методів для обґрунтування вибору одного з них у якості базового для зведення програмного засобу, що проектується.

Одним із найбільш передових методів визначення величин, що слабо підлягають формалізації (як, наприклад, тональність тексту), є використання апарату нейронних мереж [7]. Хоча принципову наявність такої можливості неможливо заперечувати, але практична реалізація її стикається із непереборними на даному етапі розвитку галузі штучного інтелекту (ШІ, або AI від англ. Artificial Intelligence) труднощами: абсолютно не очевидним є питання про те, яку саме інформацію слід подавати на вхід нейронної мережі і у якій формі вона має бути представлена (проблема формалізації вхідних даних [8]). Також для початку використання нейронної мережі вона попередньо повинна бути навченою на достатньому наборі навчальних вибірок. При цьому кожному повідомленню із цієї вибірки має бути поставлене у відповідність певне значення тональності, що, очевидно, має пропоставляти людина-експерт у даній галузі [9]. Від якості та адекватності цих оцінок залежить якість роботи нейронної мережі при подальшій звичайній експлуатації. Єдиним шляхом дотримання цієї вимоги є залучення не одного, а цілої групи експертів та усереднення оцінок виставлених по кожному конкретному текстові усією групою [10]. В такому випадку адекватність оцінок текстів для навчальної вибірки не викликатиме сумнівів, але виникає проблема забезпечення роботи великої кількості досвідчених експертів, що легко може бути організовано централізовано силами державних структур та організацій, але неможливо виконати в рамках написання даної роботи однією фізичною особою-студентом. Беручи до уваги усі перераховані особливості застосування нейронних мереж для задачі визначення тональності інформаційних повідомлень, можна зробити висновок, що у даному дослідженні застосувати такий підхід немає можливості.

Також для цілей даного дослідження можуть використовуватися наступні методи: опорних векторів, наївний класифікатор Баєса, логістична регресія та лексемний метод.

Наївний класифікатор Баєса (НБК) – це простий імовірнісний класифікатор, заснований на застосуванні Теорема Баєса зі строгими припущеннями про незалежність [11]. Часто результати роботи класифікатора

продовжують корелювати з істинною приналежністю образів до класів навіть за умови істотної залежності. Згідно теореми Баєса імовірнісна модель буде мати вигляд:

$$p(C|F_1, \dots, F_n) = \frac{p(C)p(F_1, \dots, F_n \vee C)}{p(F_1, \dots, F_n)}, \quad (2.1)$$

де C – класова змінна;

$F_1 \dots F_n$ – змінні.

Після ряду ітерацій та використовуючи припущення про незалежність, умовний розподіл по класовій змінній C може бути виражено так:

$$p(C|F_1, \dots, F_n) = \frac{1}{Z} p(C) \prod_{i=1}^n p(F_i \vee C) \quad (2.2)$$

де Z – це масштабний множник, що залежить тільки від F_1, \dots, F_n , тобто константа, якщо значення змінних відомі.

До переваг цього класифікатора можна віднести наступні [12]:

- простота реалізації;
- досить швидкий процес навчання;
- незважаючи на те, що припущення про незалежність класифікаційних ознак не є вірним в природній мові (значення слова залежать від контексту), НБК часто показує хороші результати при текстовій класифікації.

До недоліків цього методу можна віднести наступні:

- значення, що повертаються при класифікації, не можна трактувати, як імовірнісні. Таким чином не можна відповісти на питання, з якою часткою впевненості вийшов результуючий клас;
- так як в природній мові слова не є незалежними, НБК не є найоптимальнішим методом.

Таким чином, через необхідність забезпечення максимальної простоти представлення результатів масових досліджень тональності повідомлень класифікатор Баєса використовувати не доцільно.

Метод опорних векторів (англ. Support vector machines) шукає гіперплощину, яка найкраще розділить дану вибірку на два класи. Допускається багатокласова класифікація стратегіями one-vs-all і one-vs-one. Дана вибірка елементів $x_i \in \mathbb{R}_n$ і зіставлені їм класи $y_i \in \{-1,1\}$. Об'єкти вибірки представляються точками. Опорні вектори – це точки даних, розташовані ближче всього до гіперплощини, при їх видаленні зміниться становище гіперплощини [13]. Чим далі дані лежать від гіперплощини, тим вірніше вони класифіковані. Кращою гіперплощиною вважається та, відстань $\frac{1}{|w|v}$ від якої до кожного класу максимальна, тут w - нормальний вектор до роділяючої гіперплощини, яка може бути записана як множина точок x , задовольняючих рівняння:

$$wx - b = 0, \quad (2.3)$$

де b - допоміжний параметр.

Мінімізуючи вказану відстань і виключаючи попадання у проміжок, дане завдання вважають еквівалентним пошуку сідлової точки функції Лангранжа. Та завдання зводять до спеціального типу – завдання квадратичного програмування де присутні лише подвійні змінні λ_i . Після вирішення його остаточний класифікатор записується як:

$$a(x) = \text{sign}(\sum_{i=1}^n \lambda_i y_i x - b) \quad (2.4)$$

Якщо отримана вибірка лінійно нероздільна, відбувається відображення векторів в простір більш високої розмірності, при цьому, в наведеній вище формулі відбувається заміна скалярного добутку на одну з функцій нелінійного ядра $K(x_i, x)$. Після чого будується найкраща розділяюча гіперплощина.

Логістична регресія здійснює оцінку, з урахуванням деяких попередніх даних, ймовірність настання будь якої події для конкретного об'єкта [14]. Логістична регресія працює з бінарними даними: подія не відбувається (0) чи відбувається (1).

Функція логістичної регресії має s-образну криву, обмежену між 0 і 1 по осі ординат, яка показує імовірність того, що подія відбулася. Для обчислення ймовірності такий класифікатор використовує поняття коефіцієнта або відношення шансів. Воно визначається як відношення ймовірності $P(y = 1 | x)$ того, що подія для об'єкта x відбудеться, до того моменту, що цього не станеться:

$$Odds = \frac{P(y=1|x)}{1-P(y=1|x)} \quad (2.5)$$

Взяття натурального логарифма відношення можливостей називають логістичним:

$$Logit(P(x)) = \ln\left(\frac{P(y=1|x)}{1-P(y=1|x)}\right) \quad (2.6)$$

Передбачення настання події може бути сформульовано через лінійну регресію як:

$$P(x) = \beta_0 + \beta_1 x \quad (2.7)$$

де β_0, β_1 - коефіцієнти регресії, де $P(x) \in P(y = 1 | x)$.

Але ймовірність при цьому приймає значення $P(x) \in (-\infty; \infty)$, для її обмеження в необхідних межах формули прирівнюються, перетворюються і отримується остаточне рівняння логістичної кривої, наступного вигляду:

$$P(y = 1|x) = f(z) = \frac{1}{1+e^{-z}} \quad (2.8)$$

де $z = \beta_0 + \beta_1 x$.

У випадку з n незалежними змінними-ознаками x_i і фіктивним $x_0 = 0$, маємо вектор ознак $xx = (1, x_1, \dots, x_n)$, параметри $\beta = (\beta_0, \dots, \beta_n)$ і $z = \beta^T x$. Навчання класифікатора полягає в пошуку коефіцієнтів логістичної регресії, для цього використовується метод максимальної правдоподібності, де відбувається їх підбір і максимізація функції правдоподібності L . Для простоти зазвичай використовують максимізацію натурального логарифма цієї функції, на навчальній вибірці.

$$\beta = \arg \max_{\beta} \ln L(\beta) = \arg \max_{\beta} \left[\sum_{i=1}^m \log P\{y = y^{(i)} \vee x = x^{(i)}\} \right], \quad (2.9)$$

де $P\{y | x\} = f(\beta^T x)^y (1 - f(\beta^T x))^{1-y}$, $y \in \{0,1\}$, $x \in \mathbb{R}^n$ – функція розподілу у при даному у,

m - кількість елементів в навчальній вибірці.

У методі опорних векторів та логістичної регресії результат представляється у дихотомічному (бінарному) вигляді, який дозволяє визначити наявність або відсутність певного емоційно-психологічного забарвлення тексту, але не надає його чисельної оцінки, що не зовсім відповідає уточненій постановці задачі.

Після розгляду вищеописаних методів, за основу у даному дослідженні доцільно прийняти дуже простий, але в той же час досить ефективний лексемний метод. При його використанні складається словник, у якому кожній лексемі (слову) ставиться у відповідність певний бал полярності або краще сказати тональності x_i [15]. На основі тональностей окремих слів вираховується результат для усього тексту в цілому, що виконується за простою для сприйняття, але досить ефективною наступною формуло [16]:

$$S = \frac{1}{m} \sum_{i=1}^m x_i \quad (2.10)$$

де m – загальна кількість лексем із тексту, тональність якого оцінюється, що присутні у словнику, який використовується системою.

Підсумовуючи даний аналіз, складено таблицю, в якій розглянуто основні засади згідно яких приймалось рішення про вибір методу для системи (табл. 2.1).

Таблиця 2.1 – Порівняння методів визначення тональності.

Назва методу	Розглядає, окрім слів, словосполучення	Швидкодія	Об'єктивність
НБК	–	–	–
Опорні вектори	–	–	–

Назва методу	Розглядає, окрім слів, словосполучення	Швидкодія	Об'єктивність
Логістична регресія	–	–	–
Розширений лексемний метод	+	+	+
Нейронні мережі	–	+	+

В подальшому викладі формула (2.10) буде удосконалена для підвищення ефективності виконуваних оцінок (без суттєвого збільшення обчислювальної складності моделі, або значного ускладнення її сприйняття людиною).

2.1.2 Аналіз систем та методів протидії інформаційному впливу

Окрім питання оцінювання тональності інформаційного повідомлення, на наступному кроці постає завдання, як трактувати отримані результати оцінювання та які дії (протидії) необхідно виконувати при розпізнаванні тієї чи іншої тональності. На цьому кроці необхідно вже створювати або використовувати системи та методи протидії певному впливу, який буде виявлено шляхом присвоєння деякої оцінки тональності проаналізованої інформації.

Для того, щоб розуміти які дії необхідно виконувати в тій чи іншій ситуації, потрібно визначити, що власне мають передбачати механізми протидії інформаційним загрозам. Один з запропонованих підходів, описує наступні пункти які повинні передбачати механізми протидії [17]:

- мету забезпечення безпеки, що полягає у збереженні цілісності й захищеності інформаційної сфери у процесі її функціонування й розвитку;
- рівень безпеки, що диференціює структурні складові системи, які можуть стикатися з потенційними і реальними небезпеками;
- сфери безпеки, що визначають можливості функціонування й розвитку інформаційної сфери;

– параметри безпеки, що встановлюють припустимі межі відхилень у потенціалі системи інформаційної безпеки, кількості її елементів, їх якості, властивостях, зв'язках;

– перелік загроз, наслідки їх реалізації й механізм запобігання, обумовлені: внутрішніми закономірностями функціонування системи інформаційної безпеки й впливом на неї зовнішнього середовища, що веде до небажаних і незворотних порушень в її відтворенні й розвитку; змінами, що наступають у випадку реалізації загроз (такі, що компенсуються або не компенсуються; оборотні та необоротні; такі, що зачіпають або не зачіпають життєздатність системи); організаційно оформленою сукупністю стратегічних і тактичних дій, що забезпечують підтримку функціонування системи інформаційної безпеки та її здатність до самовідтворення.

Серед інформаційним загроз зовнішніх джерел найбільшу небезпеку становлять [17]:

– спроби несанкціонованого доступу до інформації й впливу на інформаційні ресурси та інформаційну інфраструктуру організації;

– недостатня поінформованість працівників;

– поширення дезінформації за межами організації, установи, країни тощо;

– порушення прав громадян в інформаційній сфері.

Отже, своєчасний моніторинг характеру, особливостей, масштабів загроз та їх наступне прогнозування мають особливе значення. Прогнозування є важливим і самостійним елементом профілактики інформаційних загроз зовнішніх джерел, та, відповідно, забезпечення безпеки усього об'єкту.

Основним методом прогнозування є моделювання, головними принципами якого є [17]: встановлення мети моделі; виділення обмеженої кількості ключових факторів, які привносять істотні зміни досліджувану систему; встановлення характеру взаємозв'язків між виділеними факторами; встановлення принципу множинності зв'язків між факторами й виділення сутнісних зв'язків, які й визначають характер розвитку й зміни системи.

Стратегічні знання, отримані за результатами прогнозування, дозволяють ілюструвати модель розвитку досліджуваного середовища.

Методи протидії інформаційним загрозам зовнішніх джерел можна умовно розділити на дві групи [17]:

– профілактичні, або превентивні, методи, які використовуються для недопущення розгортання відповідних загроз або для запобігання появі подальших ризиків на початковому етапі розгортання таких загроз;

– оперативні методи, які використовуються безпосередньо у відповідь на агресивні кроки, що виходять від зовнішніх джерел інформаційних загроз та пов'язані з їх розгортанням та реалізацією.

При розгляді питання інформаційного впливу необхідно також визначити хто є об'єктами цього впливу. Ними можуть виступати [18]: окрема особа – *Ind*, колектив (група осіб) – *Kol*, регіон – *Reg*, держава – *Der*. Кожен з перерахованих об'єктів можна описати певним набором ознак, який може варіюватися залежно від мети інформаційного впливу. Слід виділити основну групи, з якою в подальшому будемо працювати та розібрати її прогнозовані набори ознак. Колектив (група осіб) [18]:

$$Kol = \{Gal, Ker, Reg, StP, Pr, IK\} \quad (2.11)$$

де *Gal* – галузь народного господарства, до якої належить колектив, *Ker* – ставлення до керівництва, *Reg* – регіон де працює колектив (де проживає група осіб), *StP* – стан підприємства (проблеми регіону), *Pr* – проблеми колективу, *IK* – інші ознаки колективу (групи людей).

Для переваги над організованим супротивником, під час інформаційного протистояння, необхідно мати власну ефективну систему інформаційного впливу. При цьому таку систему треба розглядати через призму цілей інформаційного впливу, які можуть бути стратегічними і тактичними, та задач, на які спрямовані впливи – деструктивні або захисні.

В інформаційній війні виділяються три основних мети: контроль інформаційного простору і забезпечення захисту своєї інформації від ворожих

дій; використання контролю над інформаційним простором для проведення інформаційних атак на ворога; підвищення загальної ефективності збройних сил шляхом повсюдного впровадження військових інформаційних функцій. Можна виділити такі методи протидії інформаційним війнам та методи захисту інформаційного простору [19]:

- пряме спростування;
- встановлення та знешкодження потенційних каналів просочування інформації;
- непряме спростування (наприклад, вказати на сумнівність джерела інформації; абсурдизація звинувачень; прив'язка джерела інформації до будь-якої негативної події; введення ще одного негативного факту, який легко піддається спростуванню);
- відвертання уваги. Варіантами можуть бути: відвертання ресурсів противника на інший об'єкт шляхом перенаправлення його на іншу діяльність (наприклад на відбиття інформаційної атаки на нього або на його протеже); введення в інформаційний простір нового сенсаційного повідомлення та відвертання уваги аудиторії на нову сенсацію; відвертання уваги аудиторії на малозначний факт у рамках поточної проблеми (концентрація уваги аудиторії на не принципових для вас моментах у рамках озвученої супротивником проблеми);
- мовчання у відповідь;
- мінімізація впливу (наприклад, акцентування на тому, що в повідомленні вказано на деякі правдиві події);
- дискредитація (умисні дії, спрямовані на підривання авторитету, іміджу і довіри до джерела негативної інформації). Варіантами можуть бути: розголошення компромату; розголошення віртуального компромату; негативна похвала (така дія має на увазі публічну похвалу об'єкту дискредитації, специфіка полягає в тому, як похвалили або хто похвалив, якщо похвала виходить від негативного у сприйнятті аудиторії об'єкту, то вона матиме також негативний характер); невмотивоване освістування (масове висловлювання негативного судження з приводу інформації або її джерела); громадське обурення (метод,

близький до невмотивованого освистування, але зі зверненням до інших соціальних почуттів аудиторії);

– розмиття негативу (генерація нейтральної або позитивної інформації про об'єкт в об'ємах, що перевищують об'єми негативної інформації);

– доведення до абсурду (спосіб, що ґрунтується на виробленні імунітету у аудиторії до негативу про об'єкт).

2.2 Розробка критеріїв для визначення тональності тексту

Отже, у попередньому підрозділі розглянуто існуючі методи визначення тональності повідомлень, серед яких за рядом критеріїв обрано один – лексемний – на основі якого буде розвиватися даний проект та виконуватися програмна реалізація. Однак, доцільно розглянути і інше питання, яке з одного боку пов'язане з вибором методів вирішення задачі, але з другого являє собою відокремлену задачу, що, при правильному вирішенні, може доповнити прийнятий лексемний метод. Мова іде про вибір інформації, на основі якої можна здійснювати оцінку тональності (будь-яким методом), тобто про критерії оцінювання.

Іншими словами процедура вибору критеріїв відповідає на питання «На основі якої інформації буде здійснюватися оцінка?», а вибір методу – на питання «Як саме буде використовуватися вхідна інформація для вироблення кінцевого рішення?». Очевидно, що як зазначено вище, ці питання взаємопов'язані, але в той же час і мають певну автономність.

Так, при виборі лексемного методу очевидно, що у якості вхідної інформації буде використано два набори (або множини у термінах математичної теорії множин) слів:

W_1 – множина слів, з яких складається саме повідомлення;

W_2 – множина слів, що містяться у використовуваному словнику.

Переріз цих множин якраз і являтиме собою набір критеріїв W для оцінки тональності даного конкретного інформаційного повідомлення лексемним методом:

$$W = W_1 \cap W_2 \quad (2.11)$$

Тоді число m із попереднього підрозділу являє собою потужність (тобто кількість елементів) множини W :

$$m = |W|$$

Відзначимо, що у такій постановці задача визначення тональності уже може розглядатися як багатокритеріальна. Дійсно, можна легко представити спрощений метод визначення тональності, що працює за наявності у тексті всього лише одного якогось характерного слова (наприклад, «вбити» та його варіацій – словоформ) – цей метод, який є підвидом лексемного, цілком законно можна називати однокритеріальним, але якщо таких слів розглядати хоча би два (наприклад, «вбити» - «любити»), то задача стає двохкритеріальною, а при більшій кількості слів (очевидно, що будь-який більш-менш серйозний словник, не зважаючи на ступінь специфічності предметної галузі, повинен містити принаймні сотню-другу лексем) – багатокритеріальною.

Отже, підсумовуючи, можна сказати, що набір критеріїв для визначення тональності не є фіксованим, а залежить від самого повідомлення і може бути ототожненим із виразом (2.2). Крім того, якщо говорити про тексти в мережі Інтернет (а не, наприклад, в газеті, чи на заборі), то до лексем можна додати ще деякі показники, які можуть покращувати ефективність оцінки повідомлень. Серед них можна назвати т.зв. смайли, або що часто використовують у сучасному Інтернет-суспільстві – скорочені смайли, яких розглядають лише два види:

– деяка кількість закриваючих круглих дужок «))))» означає позитивне (веселе) забарвлення повідомлення;

– відкриваючі фігурні дужки, що ідуть підряд, означають негативне (сумне) повідомлення.

Окрім смайлів, у деякій мірі можна використовувати знаки оклику та питання, адже наявність у повідомленні значної кількості питань робить його спокійним, закликає думати та робити власні висновки, що, при задовільному рівні освіченості населення та грамотній державній політиці в цілому є позитивним процесом. В той же час численні знаки оклику явно свідчать про незадоволений стан автора повідомлення, наявність закликів до активних дій (які, нажаль, в останні роки мають виключно негативний характер «боротися», «скидати», «долати» і т.п., а заклики до творіння «зробити», «створити», «розробити» та ін. зустрічаються у мережі Інтернет усе рідше). Таким чином, суттєвий інтерес являє відносна кількість знаків оклику та питання у текстовому повідомленні.

Ще одним критерієм, що може використовуватися скоріше не для оцінки тональності, а для визначення ступеня важливості повідомлення, для вироблення рішення, чи потрібно взагалі звертати увагу на повідомлення (навіть при дуже негативному забарвленні), може братися довжина цього повідомлення: чим воно довше, тим доцільніше розглядати його докладно. Дійсно, коротке образливе слово, може при використанні лексемного методу надати дуже високу оцінку негативності, але очевидно, що у нормальному суспільстві (де є необхідність лише уникати тяжких кримінальних злочинів, як тероризм, сепаратизм, перевороти, заворушення, і т.д.) таке повідомлення не повинно викликати якогось реагування спеціальних державних служб, а скоріше, відноситься до площини моралі і підлягає лише суспільному осуду.

Таким чином, у повній мірі задачу, що розглядається, можна вважати багатокритеріальною, а набір критеріїв визначається виразом (2.2), деякими спецсимволами та знаками пунктуації (якими можна виражати емоції), а також деякими синтаксичними особливостями тексту (як, наприклад, довжина тексту).

2.3 Обґрунтування методу вирішення багатокритеріальної задачі

У попередньому підрозділі за основу при побудові програмного засобу для визначення тональності інформаційного повідомлення обрано лексемний метод

та наведена формула (2.1), яка дозволяє отримати єдиний кінцевий числовий результат за багатьма критеріями x_i , що описують задане повідомлення. У цій формулі використовуються тональності x_i для кожної лексеми, присутньої у словнику та висхідному повідомленні. Однак, у даній роботі пропонується на додачу до тональності лексеми x_i ще розглядати її вагу w_i , яка виражає достовірність яку несе використання даної лексеми [20]. Пояснити це зручно на наступних прикладах. Якщо розглянути дві лексеми «тварюко» та «пройдисвіт», то як мінімум вони мають однакову степінь негативності, а, ймовірноше, більшість експертів віднесли би перше слово до більш грубих, тобто негативних. У будь-якому разі перше слово однозначно є більш емоційним, виражає високу ступінь ненависті до особи та, можливо, гнів, в той час як друге частіше використовується у спокійних та виважених обставинах. Однак, першому слову слід приписати значно меншу вагу, ніж другому, оскільки назвати тварюкою можна будь-кого (чоловіка чи жінку, стару людину чи молоду, таку, що зробила щось конкретне або існує взагалі), це слово є дуже загальним, неконкретним. В той же час, людину, котра когось вбила, не називають пройдисвітом, так само, як і гвалтівника, і грабіжника, і т.п. Насправді це слово є набагато конкретнішим, ніж перше, і одразу значно звужує суть проблеми, через яку це слово взагалі застосовується (тобто несе більше достовірної інформації, ніж перше). Ще конкретнішим (тобто таким, що повинно мати ще більшу вагу) є, наприклад, слово «хабарник», оскільки тут мова іде уже про конкретний злочин і це слово несе набагато більше інформації, ніж просто «пройдисвіт» (як, в принципі, можна назвати і завзятого хабарника).

Аналогічна ситуація спостерігається і зі словами позитивного плану. Наприклад, навіть при використанні у якості лексем цілих словосполучень, можна наводити безліч ситуації коли і вони матимуть досить різноманітний зміст. Оскільки українська мова є надзвичайно багатою, то передбачити усі варіанти використання того, чи іншого слова або словосполучення просто неможливо, і єдиним виходом із цієї ситуації є якраз впровадження вагових коефіцієнтів для усіх слів словника, що

показували би достовірність тієї тональності, яку несе використання даного слова у текстовому повідомленні.

Таким чином, вага кожного негативного чи позитивного слова показує ступінь його конкретності, обґрунтованості використання, достовірності правильного розуміння цього слова і враховується шляхом модифікації формули (2.1) до зваженої суми виду:

$$S = \frac{1}{m} \sum_{i=1}^m (w_i x_i), \quad (2.12)$$

де w_i , – вага i -тої лексеми.

Для конкретності пропонується усі ваги задавати за 5-бальною шкалою від 1 до 5 включно, де більшому значенню числа відповідає більша достовірність, конкретність та обґрунтованість використання даної лексеми. Для зручності по такій же звичній для середньостатистичної людини схемі пропонується оцінювати і тональність самих слів, причому, на відміну від ваги, тональність «поганих» слів будемо брати зі знаком «мінус».

При прийнятих допущеннях результат розрахунку може бути тим більше, чим більше слів у тексті, який оцінюється. Для утримання результату оцінки будь-якого тексту в одних і тих же межах слід трохи модифікувати (2.3), вводячи нормуючий множник, у якості якого можна вибрати найбільше значення ваги $w_{\max} = 5$:

$$S = \frac{1}{m} \sum_{i=1}^m \left(\frac{w_i}{w_{\max}} x_i \right) = \frac{1}{m \cdot w_{\max}} \sum_{i=1}^m (w_i x_i) = \frac{1}{5m} \sum_{i=1}^m (w_i x_i). \quad (2.13)$$

Введення такого нормуючого множника гарантує, що при будь-яких характеристиках тексту (довжині та наповненості його словами із словника) кінцева оцінка не вийде за прийняті для тональності однієї лексеми межі від -5 до $+5$. Формула 2.4 прийнята за основу для виконання подальшої програмної реалізації.

2.4 Розробка алгоритмів

2.4.1 Розробка алгоритмів визначення тональності текстового повідомлення

Як відомо, розробка будь-якого програмного забезпечення повинна спиратися на відповідні алгоритми, що можуть записуватися у текстовій формі (якщо є порівняно простими), представлятися діаграмами мови UML чи блок-схемами алгоритмів, іншими наочними засобами. Для даної роботи можна виділити два різних за призначенням та своєю суттю алгоритмів:

- алгоритм роботи користувача із розробленим програмним засобом;
- алгоритм процесу визначення тональності за самим текстовим повідомленням.

Інші особливості роботи програмного продукту не представляють будь-якого інтересу через їх очевидну простоту. Отже, розглянемо алгоритм роботи користувача із програмою; він є досить простим та має лінійну (пакетну) послідовність виконання, схема якої наведена на рисунку 2.2. схема самого алгоритму наведена на рисунку 2.3.

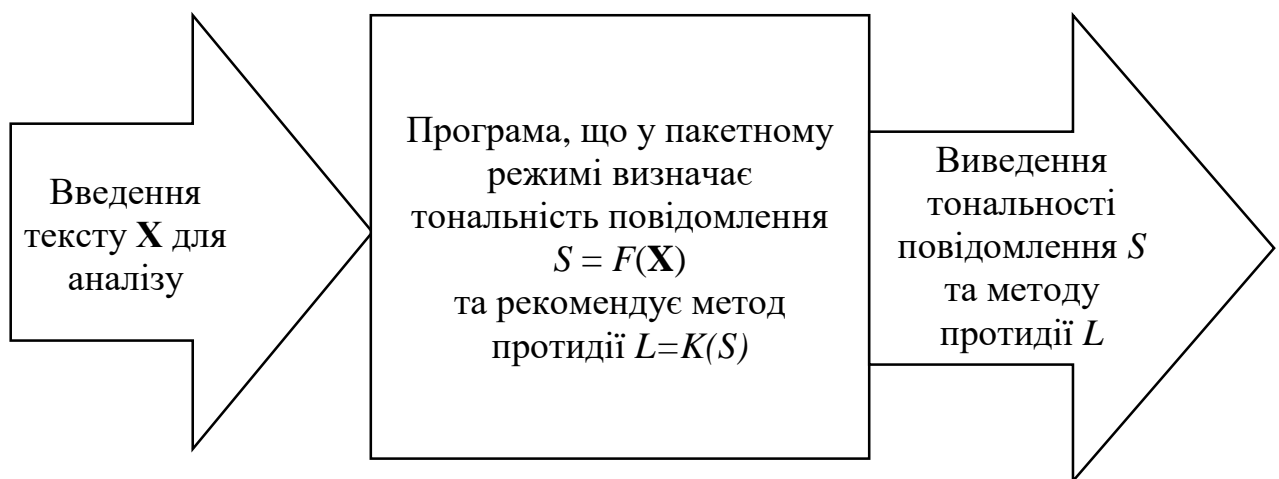


Рисунок 2.2 – Пакетна схема роботи програми (лінійний алгоритм)

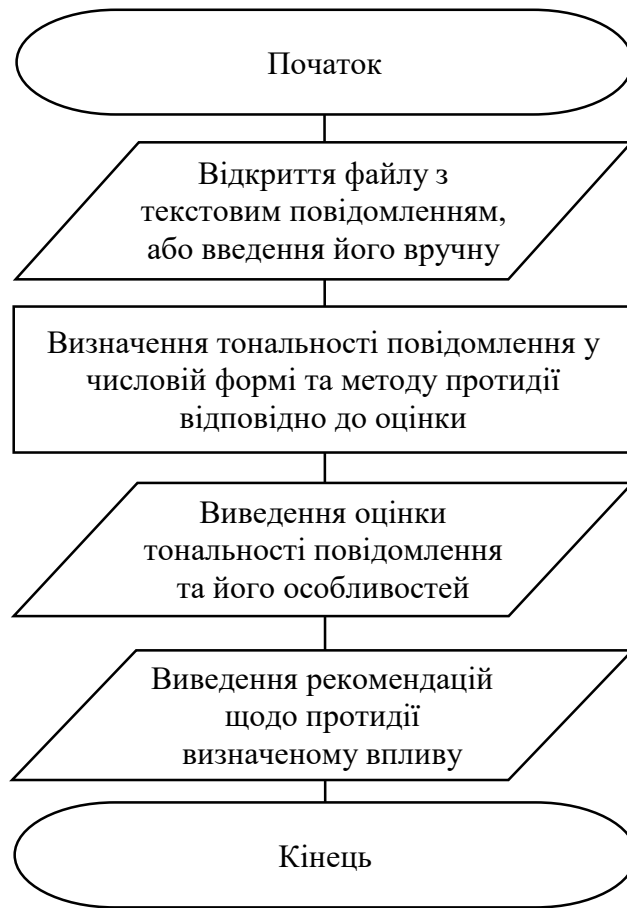


Рисунок 2.3 – Схема алгоритму роботи користувача із системою

Також можна розглянути структуру програми у вигляді схеми взаємодії її елементів (рис. 2.4).

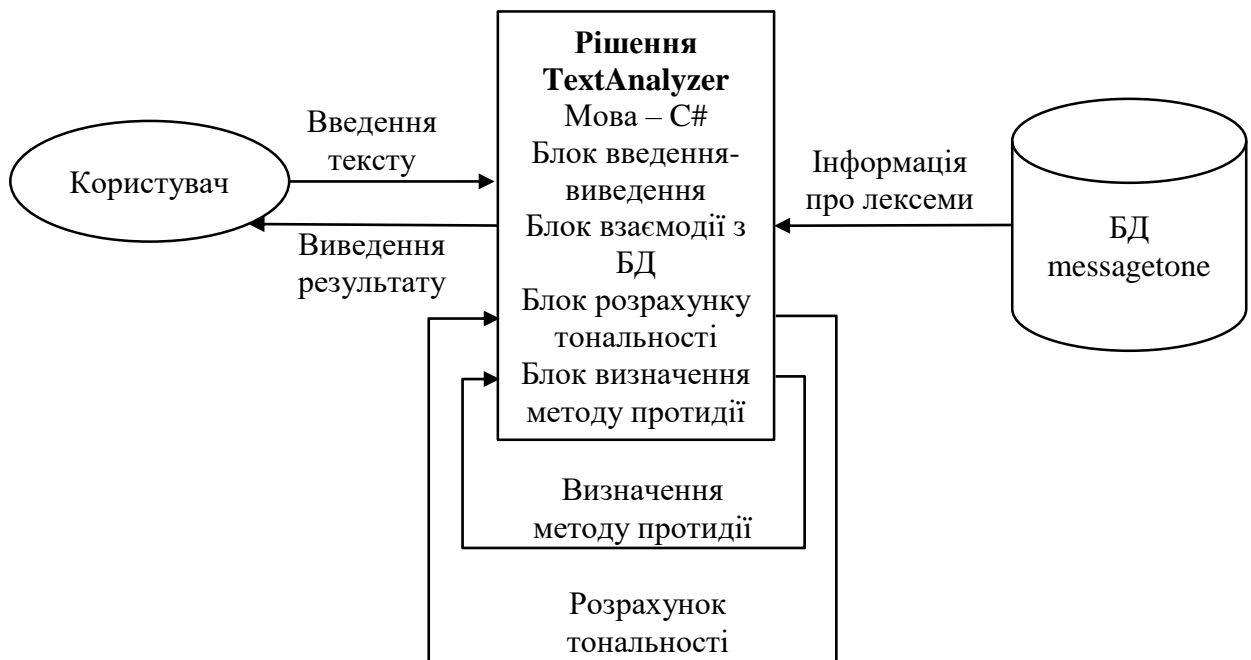


Рисунок 2.4 – Структурна схема роботи програми

Наведені схеми відображують сутність роботи користувача із програмою для аналізу тональності текстових повідомлень. Дещо більший інтерес представляє алгоритм самого програмно-технічного процесу підрахунку числової оцінки тональності відповідно до залежності (2.4), який коротко позначено на рисунку 2.3 третім блоком з початку.

Для виконання деталізації цього алгоритму слід зауважити, що при консультаціях з провідними фахівцями у даній галузі було встановлено, що доцільно у якості лексем розглядати не тільки окремі слова, а й їх словосполучення (комбінації із двох слів), оскільки участь деяких слів у словосполученнях кардинально міняє їх суть. В той же час існує і значний масив слів, які зберігають свій зміст у будь-яких комбінаціях. Таким чином, контроль за (2.5) доцільно вести за двома типами лексем (суть методу від цього не змінюється, лише виникають особливості програмної реалізації):

- окремі слова, що завжди, незалежно від участі у словосполученнях, несуть певний позитивний або негативний зміст;
- словосполучення, які часто використовуються у текстових комп'ютерних повідомленнях, та у комбінації можуть нести у деякій мірі інший зміст, аніж при їх використанні окремо.

Оскільки словосполучення є більшою одиницею, аніж окреме слово, то, по-перше, розбір тексту доцільно вести по одному слову, а, по-друге, для чергового слова спочатку слід перевірити його участь (разом із наступним словом) у контрольованих словосполученнях (тобто таких, що наявні у відповідному словнику бази даних, що використовується для збереження інформації). Якщо це так, тобто чергове слово є першим у одному із контрольованих словосполучень, то слід пропустити і наступне слово (яке утворює пару в даному словосполученні), урахувати їх спільний внесок у загальну тональність і перейти до другого за чергою слова. Якщо ж чергове слово не утворює жодного із контрольованих словосполучень із наступним за ним

словом, то тоді, по-третє, слід перевірити його наявність у словнику одиноких слів.

Для виконання контролю спецсимволів та знаків пунктуації слід ще раз посимвольно перевірити увесь масив символів, якими можна представити текстове повідомлення, що аналізується, та порахувати кількість знаків «!», «?» (причому кириличних та латинських), «(» та «)» (ці символи враховуються тільки у випадку, якщо розміщені по 2 підряд і більше). Це можна зробити до перевірки слів або після неї.

Детальна схема алгоритму роботи розробленої системи представлена у додатку Б.

2.4.2 Розробка алгоритмів протидії деструктивному впливу

Оскільки у одному з попередніх підрозділів було зазначено, що оцінка лексеми знаходиться в межах від -5 до 5. З цього слідує, що кінцева оцінка тональності також буде знаходитися в межах від -5 до 5. Тому для підбору методу протидії розділимо дану шкалу на 5 проміжків, які будуть визначати який з методів потрібно застосовувати в тому чи іншому випадку. З огляду на кінцеві значення меж, доцільно розділити їх на проміжки з кроком 2. В кінцевому результаті отримуємо 5 проміжків, для яких будуть складені рекомендації для протидії впливу, що надійде з деякого джерела. Тобто 5 проміжків кожен з яких буде знаходитись в наступних межах:

- 1) [-5; -3);
- 2) [-3; -1);
- 3) [-1; 1);
- 4) [1; 3);
- 5) [3; 5].

Спираючись на це розділення перейдемо до розробки методів протидії.

На кожен з даних проміжків необхідно реагувати по різному так, як можливий вплив, вони за собою несуть, як правило різний.

Відповідно для кожного з даних проміжків було створено ряд рекомендацій, що відображені у вигляді таблиці (табл. 2.1).

Таблиця 2.2 – Рекомендації для протидії деструктивному впливу.

Проміжок	Рекомендація
[-5; -3)	<p>Тимчасове блокування взаємодії з джерелом. У випадку, якщо це повідомлення надійшло з веб-ресурсу або у групі соц. мережі, або у месенджері повідомити адміністрацію про його негативний характер. Якщо це лист на поштовій скриньці тимчасово заблокувати відправника. Відповідальній особі провести аналіз повідомлення і прийняти рішення про те, що робити у конкретно цьому випадку надалі. Можливі набори заходів повинні зазначатись у політиці безпеки установи.</p> <p>Можна рекомендувати виконання наступних дій:</p> <ul style="list-style-type: none"> – максимально виключити дане повідомлення із функціонування системи (для того щоб користувачі, які ще не ознайомились з данною інформацією, з нею не контактували взагалі); – повідомити користувачів, які взаємодіяли з даною інформацією, що це був сплавнований вплив на те, щоб нанести шкоду функціонуванню системи, в особливості нанести максимальну шкоду самим співробітникам (як правило психічну); – після усунення даного впливу, сформувавши повідомлення для робітників, що були піддані даному впливу, що його нейтралізує та відновить функціонування людського ресурсу у системі; – відповідно до цієї «атаки» сформувавши звіт, та використати його для подальшого формування інструктажів для усього колективу організації. <p>Якщо надалі з даного джерела будуть надходити повідомлення з оцінкою цього діапазону, виконати ті ж кроки для кожного випадку та при можливості необхідно заблокувати на постійній основі зв'язок з цим ресурсом.</p>

Проміжок	Рекомендація
[-3; -1)	<p>Віднести джерело з якого надійшло повідомлення до підозрілих. Тобто необхідно провести аналіз повідомлень, що надходили з даного ресурсу та будуть надходити найближчим часом (наприклад на протязі тижня). Якщо оцінка буде знаходитись в цьому ж діапазоні, відповідальній особі необхідно прийняти рішення з приводу чи є суттєво необхідним надходження інформації від даного джерела. Відповідно, провести більш глибокий аналіз вхідних повідомлень за певний проміжок часу:</p> <ul style="list-style-type: none"> – 2-3 тижні, якщо частота надходження інформації не дуже велика, знаходиться в діапазоні 1-3 повідомлення в день; – 1-2 тижні, якщо частота надходження інформації не дуже велика, знаходиться в діапазоні 4-5 повідомлення в день; – тиждень, якщо частота надходження інформації більша ніж 6 повідомлень в день. <p>При отриманні результатів оцінки з такими самими значеннями та прийнятним змістом даних повідомлень, зробити позначку, що дана оцінка є прийнятною для даного ресурсу. Та зрозуміти, що немає потреби у постійному аналізі фахівцем повідомлень з даного ресурсу з оцінкою, що знаходиться в даному діапазоні. А у випадку зміни оцінки потрібно насторожитись та проаналізувати чому оцінка була змінена.</p> <p>Якщо оцінка постійно змінюється, як у бік більш негативної оцінки (і це трапляється доволі часто) так і у бік більш позитивної оцінки, з'ясувати з чим пов'язані дані зміни. При необхідності, або при отриманні небажаних результатів припинити обмін з даним джерелом.</p> <p>У випадку, якщо було виявлено, що дане повідомлення або не тільки одне повідомлення, а й усі (більшість) повідомлення з даного джерела містять деструктивний вплив зробити інформативну розсилку</p>

Проміжок	Рекомендація
	<p>серед співробітників, що контактували з данною інформацією. Дана розсилка повинна роз'яснити співробітникам, що це були повідомлення, які мали на меті дестабілізувати роботу системи шляхом впливу на них (людський ресурс). А також розповсюдити у системі нейтралізуюче повідомлення, для приведення функціонування системи до норми.</p>
[-1; 1)	<p>Зазвичай таку оцінку можна віднести до категорії нейтральних, але в залежності від змісту того чи іншого повідомлення. Бажано, щоб фахівець (відповідальна особа) провів аналіз вхідних повідомлень за певний проміжок часу, з даного джерела:</p> <ul style="list-style-type: none"> – 2-3 тижні, якщо частота надходження інформації не дуже велика, знаходиться в діапазоні 1-2 повідомлення в день; – 1-2 тижні, якщо частота надходження інформації не дуже велика, знаходиться в діапазоні 3-4 повідомлення в день; – тиждень, якщо частота надходження інформації більша ніж 5 повідомлень в день. <p>Це є необхідним для того, щоб зрозуміти чи дана оцінка є характерною для інформації з даного джерела чи ні (наприклад це може бути своєрідний стиль написання автора). Якщо оцінка повідомлень перебуває в даному діапазоні або деколи і у вищому, то дане джерело можна віднести до прийнятних. Та реагувати тільки у випадку зміни оцінки у бік нижчого діапазону.</p> <p>Якщо оцінка постійно змінюється, у тому числі і в негативну сторону, з'ясувати з чим пов'язані дані зміни. При необхідності, повідомити адміністрацію ресурсу про ті чи інші зміни, можливо це буде своєчасною реакцією на проникнення до їхньої системи, якого вони не змогли виявити.</p>

Проміжок	Рекомендація
[1; 3)	<p>Така оцінка є прийнятною. Головне, щоб оцінка повідомлень джерела постійно знаходилась в своєму діапазоні, у такому випадку ніяких дій застосовувати до даного каналу віщання. Якщо оцінка даного джерела постійно змінюється необхідно провести змістовний аналіз даних повідомлень та з'ясувати з чим це пов'язано. У випадку впливу на цю зміну декількох несуттєвих фраз, то можна віднести дане джерело до декількох прийнятних діапазонів. Якщо зміни діапазону спостерігаються дуже рідко, то необхідно проводити аналіз саме тих повідомлень, які виходять за межі встановленого для даного джерела діапазону значень оцінки тональності.</p>
[3; 5]	<p>Оцінки з даного діапазону є позитивними. Тому ніяких мій саме для цих повідомлень приймати не потрібно. Але потрібно перевірити чи така оцінка є постійною чи рідкісно спливає з певного джерела. У випадку раптової зміни оцінки повідомлень з нижчих діапазонів даного джерела, потрібно проаналізувати дане повідомлення на предмет прихованих повідомлень, які через оцінку тональності виявити не є можливим. У випадку виявлення такого повідомлення тимчасово заблокувати дане джерело та зв'язатись з адміністрацією джерела, для вирішення питання чому така подія відбулась. Окрім цього виключити дане повідомлення із системи і розповсюдити нейтралізуючу розсилку, серед співробітників, що встигли ознайомитись з даною інформацією.</p>

На основі створених схем алгоритмів, в наступному розділі буде реалізовуватись програмний засіб, що розробляється.

Таким чином, у даному розділі створено теоретичне підґрунтя для наступної роботи, що полягає у впровадженні у програмних кодах системи оцінювання тональності інформаційного повідомлення:

- проаналізовано існуючі методи проведення такого визначення та обрано основний, що найбільш точно відповідає цілям та задачам даної роботи;

- запропоновано покращення обраного алгоритму, яке полягає у введенні вагових коефіцієнтів для урахування конкретності, обґрунтованості використання, достовірності правильного розуміння відповідних слів та словосполучень;

- запропоновано використовувати деякі символи для додаткової оцінки повідомлення у порівнянні з чистим лексемним методом;

- розроблено схеми алгоритмів, які будуть покладені в основу програмної реалізації оцінки тональності текстів.

Керуючись даними відомостями, можна переходити до наступного етапу роботи, а саме, проведення програмної реалізації продукту, що розробляється.

3 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ

3.1 Вибір технології розробки програмного забезпечення

Першочерговим питанням, яке постає перед розробниками будь-якого програмного забезпечення, є вибір моделі або технології, парадигми його розробки. Такими, що широко використовуються на сьогоднішні день у виробничій практиці, є технології структурного (процедурного) та об'єктно-орієнтованого програмування. Кожна з них має свої особливості, переваги і недоліки, які розглянемо детальніше.

Структурне, або як його ще називають практикуючі програмісти, процедурне програмування засноване на використанні окремих структурних блоків – в першу чергу, підпрограм (процедур і функцій) [21].

Слід зазначити психологічні особливості сприйняття людиною складних «великих» завдань. Неструктуроване «велике» завдання (наприклад, написання дипломної роботи) зазвичай викликає певний психологічний ступор і, як результат, повну неможливість розібратися з ним. Людині зручно розбити проблему на певну кількість завдань, не замислюючись про реалізацію кожного з них. Коли є ясність і розуміння проблеми на найвищому рівні абстракції, слід приступати до деталізації підзадач, кожна з яких слід розбити на окремі підзадачі нижчого рівня, більш дрібні. Уже після такого розбиття слід аналізувати всі перераховані підзадачі. На певному етапі зупиняються і виконують не розбиття чергової підзадачі на більш дрібні, а безпосередню її реалізацію в текстах програм.

Говорячи більш конкретно, структурне програмування має на увазі побудову програми відповідно до трьох основних принципів: слідування, розгалуження, повторення.

Слідування означає, що оператори і блоки тексту програми слідують і виконуються один за іншим. Розгалуження реалізується різними умовними операторами типу `if`, і дозволяє вибирати один з декількох подальших варіантів

виконання програми. Повторення зазвичай відносять на рахунок циклів, хоча цей же принцип можна віднести і до підпрограм.

Взагалі ж, структурне програмування є більш старшою методикою програмування, на зміну якій поступово прийшло об'єктно-орієнтоване програмування. Проте, при створенні невеликих програм (наприклад, до 10000 рядків коду і без передбачуваного розширення) застосування цієї методики програмування все ще виправдано і такий код краще сприймається, ніж його об'єктно-орієнтований варіант.

Зміст методології об'єктно-орієнтованого програмування полягає в тому, що система розглядається, як сукупність окремих активних сутностей – об'єктів, які мають набір певних своїх внутрішніх параметрів – властивостей, а також можуть взаємодіяти між собою за допомогою деяких дій – викликів методів.

Якщо говорити про програмний код, то для того, щоб оперувати об'єктом, його спочатку потрібно створити. Об'єкти створюються як змінні, у яких типом виступає клас об'єкта. Клас являє собою опис, які властивості можуть мати об'єкти такого типу (тобто яку інформацію вони можуть зберігати), і які у них є методи (тобто які дії вони можуть виконувати). Об'єктом же є набір значень, якими є властивості даного об'єкта (а свої методи кожен об'єкт отримує від свого класу, тобто методи однакові у всіх об'єктів, що належать даному класу).

Реалізуючи всі сутності, необхідні, згідно з алгоритмом, для роботи програми, у вигляді класів і об'єктів, ми спрощуємо її розуміння для самих себе. Саме тому ОО-підхід рекомендується до застосування для великих проектів (більше десятків тисяч рядків коду), коли утримувати «в голові» всю систему цілком стає важко. Можна сказати, що розбиття програми на об'єкти і проектування їх класів наближає розуміння предметної області до звичного людського образу мислення (в разі «великих» проектів). Людина мислить класами, об'єктами і зв'язками між ними.

Слід відзначити, що часто, крім розглянутих міркувань, також на вибір методики програмування впливають інші чинники, наприклад, можливість майбутнього розширення функціональності, створення якомога більш

зрозумілого коду (для роботи над проектом цілої команди, а не одного програміста), або просто побажання замовника застосувати найбільш сучасний підхід до програмування та ін.

Крім розбиття (декомпозиції) всієї предметної області на об'єкти (класи) і співвідношення між ними, також ОО-підхід має на увазі дотримання трьох основних його принципів: інкапсуляція, наслідування, поліморфізм.

Під інкапсуляцією мається на увазі об'єднання даних (значення властивостей класу у деякого конкретного об'єкта) та засобів їх обробки (методи класу). Це знову ж таки зручно психологічно, так як дозволяє реалізовувати окремі завершені сутності – класи, які самі обробляють свої дані. Звернення до об'єктів цих класів відбувається за допомогою методів, що утворюють інтерфейс класу.

Наслідування є дуже корисною можливістю, тому що дозволяє значно скорочувати обсяги повторюваного коду. Згідно з цим принципом виділяється клас, який має спільний набір властивостей і методів для декількох більш розширених класів. Цей клас оголошується батьківським, базовим класом для декількох похідних від нього (нащадків, спадкоємців). Всі класи-нащадки успадковують від базового всі його властивості та методи, але до цього ще мають свої власні оригінальні властивості і / або методи.

При наслідуванні іноді методи батьківського і похідного класу мають однакове призначення, але реалізуються по-різному. Такі методи називаються перевантаженими. При цьому ще раз підкреслимо, що призначення методу в обох випадках одне і те саме.

Поліморфізм є можливістю деякої функції приймати об'єкти як батьківського, так і похідних класів, і вміти викликати перевантажені методи саме того класу, об'єкт якого був переданий у функцію.

Важливими поняттями в ООП також є: статичні члени класу, абстрактні методи і класи, дружба функцій і класів, і т.д.

Ґрунтуючись на перерахованих особливостях двох найпоширеніших методик програмування, обрано об'єктно-орієнтований підхід як більш

сучасний, гнучкий та прогресивний, а також такий, що відповідає середнім масштабам проєктованого ПЗ.

3.2 Вибір засобів розробки

Після визначення технології програмування слід обрати засоби розробки, а саме мову програмування, компілятор та текстовий редактор, або, наприклад, тільки середовище розробки, що включає в себе усі засоби, необхідні для створення сучасних програм з графічним інтерфейсом.

3.2.1 Обґрунтування вибору мови програмування

При розробці програмного забезпечення після обрання технології програмування слід встановити, які з засобів розробки надають достатні можливості для реалізації цього алгоритму у програмних кодах. Слід відзначити, що різні засоби розробки, чи навіть цілі мови програмування, з самого початку свого створення мали різне призначення і деякі з них є спеціалізованими, тобто пристосованими до виконання лише певних задач окремих прикладних галузей, а деякі – засобами загального призначення. Серед останніх можна виділити такі, що містять окремі уже готові компоненти для вирішення заданої прикладної задачі, а також такі, в яких таких компонентів немає.

Також важливим фактором при виборі засобів розробки виступає його популярність (зокрема, мови програмування), тобто поширеність фактів застосування цього засобу у професійній діяльності. Не слід недооцінювати цей момент, оскільки в процесі розробки нового програмного забезпечення функціональності певного рівня складності, у будь-якого програміста виникатимуть питання щодо її реалізації, причому якраз за допомогою обраного засобу розробки. У даний час найпершим і найефективнішим місцем, де можна отримати інформацію про питання розробки програмного забезпечення є глобальна мережа Інтернет. Сучасні програмісти порівняно легко і швидко знаходять відповіді на практично будь-які питання, пов'язані із професійною діяльністю, в Інтернет. Така, практично ідеальна, ситуація реалізується тільки за

умови поточної популярності мови програмування (або середовища розробки, бібліотеки, фреймворку, і т. п.), що використовується.

Відзначимо, що на сьогоднішній день достатньо популярними є такі мови програмування загального призначення як: C++, Java, C#, Delphi.

Мова C++, є надзвичайно потужною, яка має стабільну популярність вище середнього рівня, але і досить складною для сприйняття; фактично, це інструмент для професіоналів найвищого рівня. В той же час на сьогоднішній день більш популярними є мови Java та C#. Мова Delphi (або точніше – середовище розробки, оскільки у ньому використовується мова програмування Object Pascal), не зважаючи на свої зручність та простоту (а, можливо, навпаки, через них), стабільно не має широкої популярності серед професійних розробників програмного забезпечення, а використовується в основному для малих проєктів та навчальних цілей.

Таким чином, при реальному процесі вибору мови програмування для професійної розробки програмістом середнього рівня на сьогоднішній день слід розглядати два основних варіанти – Java та C#. Дуже багато полеміки було присвячено порівнянню цих двох мов, але беззаперечним фактом є значна перевага продукту від Microsoft над конкурентом у частині зведення користувацького інтерфейсу. У цьому питанні середовище Visual Studio надає практично той же рівень зручності, що й Delphi, але використовує популярні мови, а не Паскаль, до якого традиційно укорінилося ставлення як до навчальної мови програмування для новачків. Отже, зважаючи на необхідність створення програмного засобу для операційної системи Windows, кінцево обираємо мову програмування C#, що і відповідає завданню на розробку в рамках даної роботи. Коротко розглянемо особливості цієї мови програмування.

C# – це проста, сучасна об'єктно-орієнтована мова програмування з безпечною системою типізації [22]. C# відноситься до широко відомого сімейства мов C, і буде здаватися добре знайомою кожному, хто працював з C, C++, Java або JavaScript.

C# є об'єктно-орієнтованою мовою, але підтримує також і компонентно-орієнтоване програмування. Розробка сучасних додатків все більше схиляється до створення програмних компонентів у формі автономних пакетів, що реалізують окремі функціональні можливості. Важлива особливість таких компонентів – це модель програмування на основі властивостей, методів і подій. Кожен компонент має атрибути, які надають декларативні відомості про нього, а також вбудовані елементи документації. C# надає мовні конструкції, які безпосередньо підтримують таку концепцію роботи, і завдяки цьому мова відмінно підходить для створення і застосування програмних компонентів.

Нижче наведено декілька функцій мови C#, що забезпечують надійність і стійкість програм, розроблених за допомогою неї:

- прибирання сміття автоматично звільняє пам'ять, зайняту знищеними і невикористовуваними об'єктами;
- обробка виключень надає структурований і розширюваний спосіб виявляти і обробляти помилки;
- суворі типізація мови не дозволяє звертатися до неініціалізованих змінних, виходити за межі індексованих масивів або виконувати неконтрольоване приведення типів.

У C# існує єдина система типів: усі типи, включаючи типи-примітиви, такі як `int` і `double`, успадковуються від одного кореневого типу `object`. Таким чином, всі типи використовують загальний набір операцій, і значення будь-якого типу можна зберігати, передавати і обробляти схожим чином. Крім того, C# підтримує визначені користувачем посилальні типи і типи значень, дозволяючи як динамічно виділяти пам'ять для об'єктів, так і зберігати спрощені структури у стеку.

Щоб забезпечити сумісність програм і бібліотек C# при подальшому розвитку, при розробці C# багато уваги було приділено управлінню версіями. Багато мов програмування обходять увагою це питання, і в результаті програми на цих мовах ламаються частіше, ніж хотілося б, особливо, при виході нових

версій залежних бібліотек. Питання управління версіями істотно вплинули на такі аспекти розробки C#, як:

- роздільні модифікатори `virtual` і `override`;
- правила вирішення перевантаження методів;
- підтримка явного оголошення членів інтерфейсу.

Файли висхідного коду C# зазвичай мають розширення `.cs`. Якщо код деякої програми, наприклад "Hello, World", зберегти в файлі `hello.cs`, то для її компіляції можна виконати наступну команду з командного рядка:

```
csc hello.cs
```

В результаті буде отримана виконувана збірка з ім'ям `hello.exe`.

Команда `csc` фактично і є компілятором мови C# (`csc.exe`), тобто виконує компіляцію на повній версії платформи, і вона може існувати не для усіх програмно-апаратних систем (початково призначалася для програмування для ОС Windows).

Програмування мовою C# є досить простим та зручним. В цілому, можна констатувати, що мова програмування C# на сьогоднішній день є найкращою для програмування сучасних додатків загального призначення, тому саме її обрано, як основний засіб, для реалізації програмного засобу визначення тональності інформаційних повідомлень.

Визначившись із мовою програмування, слід провести вибір засобів розробки, що залишилися (конкретний компілятор та компонувальник, текстовий редактор, система допомоги, налагодчик і т.д.). Найкраще всього використовувати для цього якийсь інтегроване середовище розробки (Integrated Development Environment, або IDE).

3.2.2 Вибір середовища розробки

Зважаючи на те, що обрана мова програмування C# з'явилася уже майже 20 років назад (у 2002 році), для неї за цей час з'явилися різноманітні альтернативні компілятори та середовища розробки, потреба у яких сильно

зменшилася, коли Microsoft почала на постійній основі видавати безкоштовні версії власного продукту Microsoft Visual Studio.

Цей інструмент беззаперечно є найпотужнішою платформою для програмування додатків, який пропонується корпорацією Microsoft, що уже більше 20 років розвиває свій продукт Microsoft Visual Studio (MVS) [23]. За цей час він від простого середовища розробки із трьома мовами (C++, Java та Visual Basic) перетворився на супергіганта, повна інсталяція якого у 2019 році займає більше 200 Гб і підтримує десятки різних мов програмування. Слід відзначити, що серцем середовища MVS є .NET Framework – програмна платформа, випущена компанією Microsoft в 2002 році. Основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), яке підходить для різних мов програмування. Функціональні можливості CLR доступні в будь-яких мовах програмування, що використовують це середовище, однак основною мовою із самого початку тут позиціонувалася саме C#. Вважається, що платформа .NET Framework стала відповіддю компанії Microsoft на платформу Java компанії Sun Microsystems (нині належить Oracle), що набрала на той час дуже велику популярність. Дійсно, детальний аналіз показує, що Microsoft використовує дуже схожі по своїй суті технології, але називаючи їх по-іншому.

Величезною перевагою продукту від Microsoft стала на порядок більша зручність використання середовища MVS, у порівнянні з продуктами для Java-розробки. Наприклад, середовище Eclipse потребує численних налаштувань для кожного проекту, який в певній мірі є специфічним. Очевидно, що при наявності досвіду використання у декілька років усі ці дії виконуються за кілька секунд, однак сьогодні часто зустрічається ситуація, коли один програміст у різні періоди використовує різні інструменти, причому значну їх кількість. Відповідно, не лише для новачків, а й для так званих «багатоверстатників» актуальним є питання якомога меншої кількості налаштувань, інсталяцій додаткового програмного забезпечення (бібліотек, модулів, і т.д.), низького порогу входження – по всім цим показникам Microsoft Visual Studio значно

випереджає своїх конкурентів, зокрема продукти для розробки на платформі Java.

І ще одним немаловажним фактором, який також діє у перевагу продукту від Microsoft, є вимоги до швидкості роботи у даному середовищі розробки при заданому обсязі оперативної пам'яті та характеристиках процесора. MVS, наприклад, виконує такі базові речі, як компіляція проекту в рази швидше (і за цілком прийнятний час порядку секунд чи десятків секунд для проектів середньої складності), у порівнянні із дійсно зручними середовищами розробки для платформи Java на зразок IntelliJ IDEA (компіляція проекту у якій може займати час порядку хвилин і навіть їх десятків).

Суттєвою перевагою Java спочатку була відкритість засобів розробки, але на сьогоднішній день існують повністю вільні для використання версії MVS, тому відповідно у цьому показникові дві даних платформи є рівними. Також хоча .NET є патентованою технологією корпорації Microsoft і офіційно розрахована на роботу під операційними системами сімейства Microsoft Windows, існують незалежні проекти, такі, як Mono і Portable.NET, що дозволяють запускати програми .NET на деяких інших операційних системах. В даний час .NET Framework отримує розвиток у вигляді .NET Core, яка з самого початку передбачає кросплатформенну розробку і експлуатацію. Таким чином, і за показником кросплатформенності дві цих платформи стали приблизно рівними у останні роки.

Беручи до уваги усі особливості зручності середовища MVS, обираємо його за основу для подальшої розробки програмного модуля, що проектується.

3.2.3 Вибір системи управління базами даних

Важливою складовою програмного забезпечення, що розробляється, є словник як характерних слів, так і їхніх словосполучень, що будуть застосовуватися у цьому дослідженні. При цьому слід не прив'язуватися до конкретного набору слів та словосполучень, а реалізувати процес зчитування гнучким: читати їх кожного разу при завантаженні програми. Для цього на

комп'ютері, де запускається розроблюване ПЗ, має бути наявним доступ до серверу баз даних із розгорнутою відповідною базою даних.

Зважаючи на відносну простоту задачі, використання потужних корпоративних продуктів типу Oracle Database або PostgreSQL є невиправданим. В той же час найпростіші засоби збереження інформації у БД типу Paradox dBase, Microsoft Access, miniSQL або є застарілими, або мають ліцензійні обмеження, або сильно обмежені функціонально. В той же час існує надзвичайно популярна на сьогоднішній день система управління базами даних MySQL, що вже стала стандартом при зведенні веб-проектів малого та середнього розміру (також може використовуватися і для великих проектів) [24]. Не зважаючи на те, що у даному проекті весь процес має виконуватися локально, це не заважає взяти до використання той же сервер баз даних, але підключатися до нього не за допомогою веб-серверу, а безпосередньо, через його програмний інтерфейс (API), чи іншими методами.

Вибір СУБД MySQL обумовлений також наступними її перевагами у порівнянні з конкурентними продуктами:

- це порівняно легкий продукт, що займає малу кількість оперативної пам'яті сервера (а також, що менш критично, і малі обсяги постійної пам'яті жорсткого диску сервера);
- це порівняно функціональний продукт, що включає практично усі основні функції по обробці даних та створенню ефективних БД;
- це програмне забезпечення є вільним для використання;
- існує значна кількість уже розробленого програмного забезпечення (фрагментів, модулів, і т. п.) для роботи з цією СУБД;
- існує чимало Інтернет-ресурсів, на яких здійснюється технічна підтримка програмістів, що здійснюють розробку для MySQL (тобто висвітлюються особливості роботи із нею).

Зважаючи на усі ці особливості, СУБД MySQL береться за основу для збереження даних, які використовуватимуться програмою аналізу текстів, що розробляється.

Наступним кроком після обґрунтування вибору СУБД є визначення алгоритмів її взаємодії із прикладним програмним забезпеченням, тобто програмою, що розробляється. Для цього можна використовувати набір функцій, що надаються самою СУБД, тобто її API, однак такий підхід не є далекоглядним кроком, оскільки, якщо в подальшому виникне потреба у переході на іншу СУБД, то для цього треба буде переписувати значну кількість рядків коду усієї програми. Іншим шляхом, що робить програмний продукт більш гнучким, менш залежним від конкретних застосованих засобів управління базами даних, є використання універсального механізму доступу до баз даних – ODBC (Open DataBase Connectivity). Це фактично драйвер для уніфікованого доступу до БД будь-якого виробника, причому особливості самої БД прописуються лише один раз – у рядку підключення до БД. Слід відзначити, що для кожної СУБД існує власний драйвер ODBC, тому перед використанням його слід завантажити та встановити. Процес встановлення даного драйвера покроково наведено у додатку В.

Після вказаних дій це джерело даних можна починати використовувати у своїх програмах, використовуючи механізм ODBC для доступу до БД MySQL із назвою messagetone. Для цього у програмному коді слід вставити наступні рядки:

```
string MyConnectionString = "Driver={MySQL ODBC 8.0 ANSI Driver};Server=localhost;Database=lingvoanalysis;uid=root;pwd=";
OdbcConnection MyConnection = new OdbcConnection(MyConnectionString);
MyConnection.Open();
OdbcCommand MyCommand = new OdbcCommand("ЗАПИТ", MyConnection);
MyCommand.ExecuteNonQuery();
MyConnection.Close();
```

Перший рядок задає драйвер ODBC та особливості підключення до бази даних. Другим рядком створюється змінна MyConnection, що уособлює саме з'єднання та включатиме його параметри. Третім рядком підключення відкривається за допомогою метода Open(). У четвертому рядку створюється нова змінна, що відповідає команді до бази даних, яка містить як сам SQL-запит (його текст слід підставити замість слова ЗАПИТ), так і змінну з'єднання, по якому слід виконати цю команду. Передостанній рядок – це безпосереднє виконання SQL-команди на сервері баз даних, а останній – закриття з'єднання.

Відзначимо, що вказаний метод `ExecuteNonQuery()` використовується для виконання SQL-команд, які не повертають жодних даних (директивні команди типу створити/видалити таблицю чи базу даних, введення відомостей у БД, тощо). Якщо необхідно виконати запит типу `SELECT`, який передбачає повернення даних (як у розроблюваному програмному забезпеченні), то слід використовувати замість методу `ExecuteNonQuery()`, метод `ExecuteReader()`.

Відзначимо, що ще одним додатковим програмним засобом, що використовувався в роботі, є `phpMyAdmin` – утиліта, написана мовою PHP спеціально для зручної роботи у СУБД MySQL (рис. 3.1).

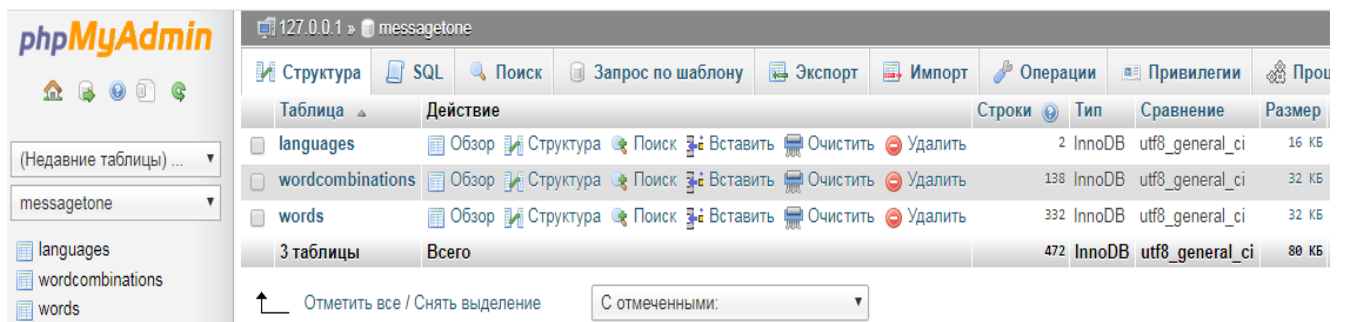


Рисунок 3.1 – Загальний вигляд вікна програмного засобу `phpMyAdmin` (із відкритою базою даних `messagetone`).

Таким чином, у підрозділі детально викладені питання вибору та налаштування засобів для збереження та обробки у БД інформації, необхідної для роботи програмного продукту визначення тональності інформаційних повідомлень.

3.3 Проектування інтерфейсу користувача програмного забезпечення

Графічний інтерфейс програмного продукту є порівняно простим. Обов'язковим елементом інтерфейсу є велике текстове поле для розміщення самого інформаційного повідомлення, що підлягає аналізу. розміщуємо дане поле у верхній частині головного вікна програми. Нижче доцільно розмістити елемент управління, що буде відображати результуючу інформацію про повідомлення. Оскільки кількість такої інформації буде не малою, її також

доцільно розмістити у великому текстовому полі, яке поміщаємо у нижню частину головного вікна програми.

Також у головному вікні розміщуємо елементи управління для запуску аналізу повідомлення, очищення текстових вікон, деяких налаштувань всього процесу. В першу чергу, як уже було зазначено вище, програма використовує два типи лексем: власне слова та цілі словосполучення (із двох слів). Очевидно, робота програми є можливою і з використанням тільки одного їх виду, тому у головному вікні розміщуємо два прапорці, установка яких означає необхідність урахування відповідного типу лексем при виконанні аналізу. Налаштуванням за замовчуванням є установлення обох прапорців (повний аналіз).

Оскільки програма має графічний інтерфейс, впроваджуємо у ній такі стандартні елементи, як головне меню, що містить:

- в меню «Файл» елемент «Відкрити...», що дозволяє завантажити повідомлення із текстового файлу (з розширенням *.txt);
- в меню «Файл» елемент «Очистити»;
- в меню «Файл» елемент «Вихід»;
- в меню «Допомога» елемент «Про програму», що викликає мале нове вікно з інформацією про даний програмний засіб.

Загальний вид спроектованого інтерфейсу наведено на рисунку 3.2.

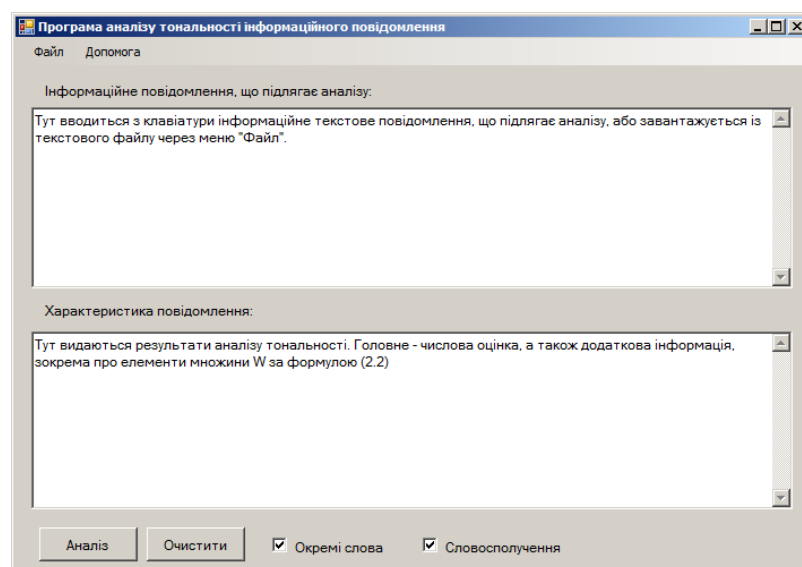


Рисунок 3.2 – Загальний вигляд головного вікна

Окрім головного вікна дане ПЗ також містить інформаційні вікна, для відображення у них відповідних до кожного проміжку оцінки рекомендацій. На рисунку 3.3 відображено вікно рекомендацій, що отримує користувач при оцінці в проміжку [-5;-3).

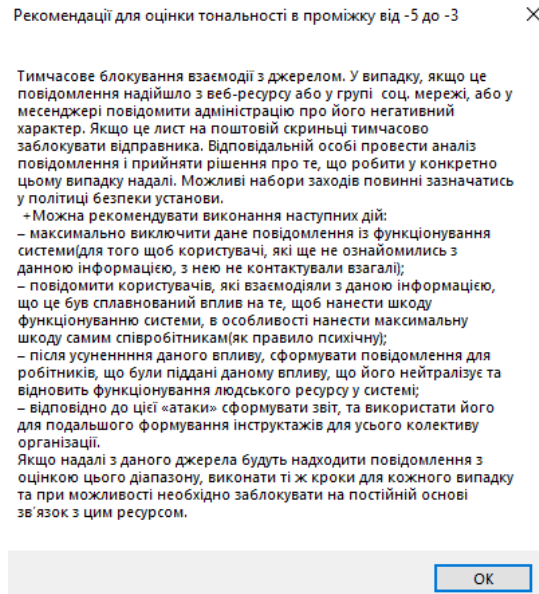


Рисунок 3.3 – Рекомендації для оцінки в проміжку [-5;-3)

Наступний рисунок, відображає вікно з рекомендаціями для оцінок, що знаходяться в проміжку [-3;-1) (див. рис. 3.4).

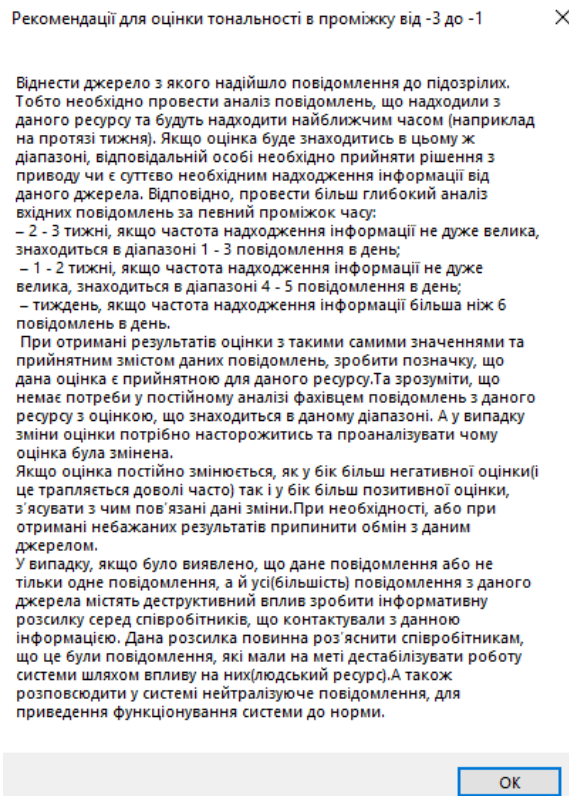


Рисунок 3.4 – Рекомендації для оцінки в проміжку [-3;-1)

Вікно з рекомендаціями отримане для оцінок, що належать проміжку $[-1;1)$ наведено на рисунку 3.5.

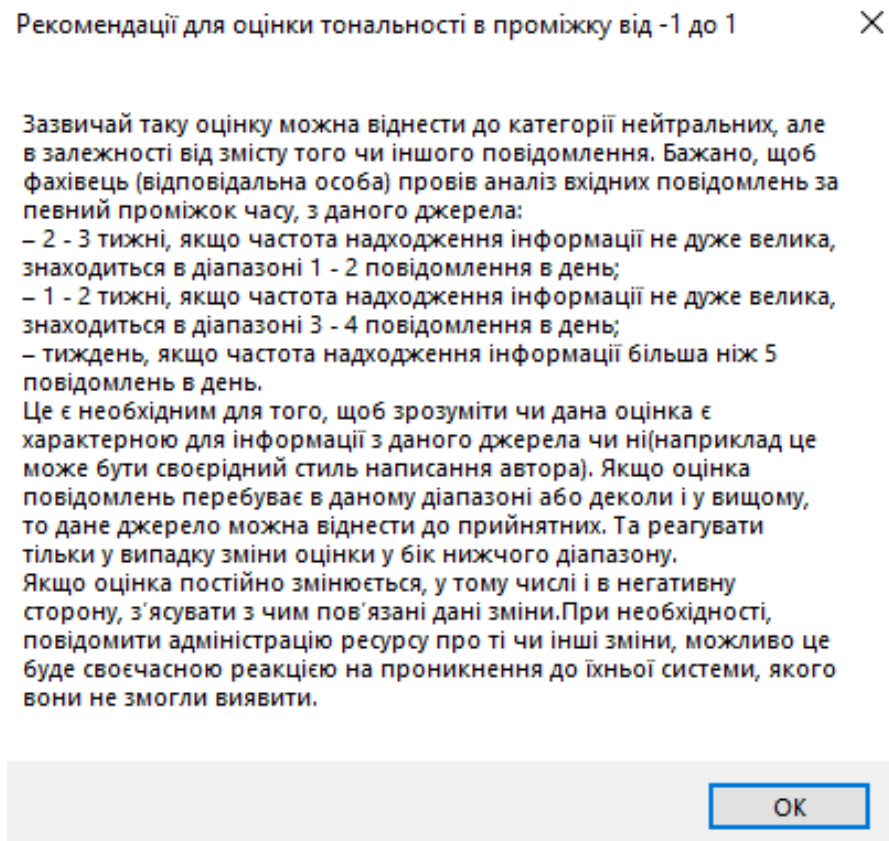


Рисунок 3.5 – Рекомендації для оцінки в проміжку $[-1;1)$

При оцінках, що входять до проміжку $[1;3)$, перед користувачем з'явиться наступне вікно з рекомендаціями (див. рис. 3.6).

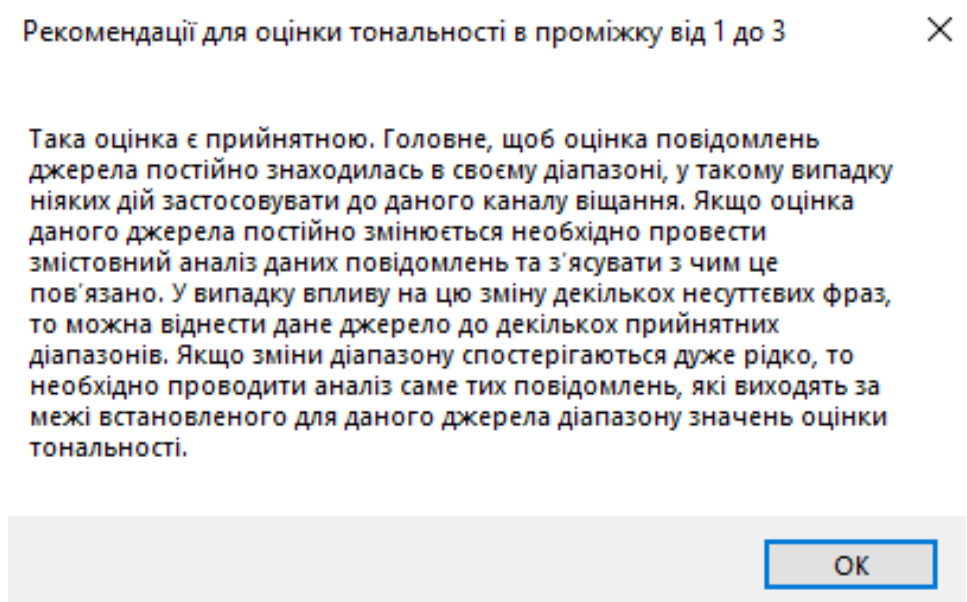


Рисунок 3.6 – Рекомендації для оцінки в проміжку $[1;3)$

А також, якщо при аналізі тексту, оцінка тональності буде знаходитись в проміжку [3;5], то вікно з рекомендаціями для даного випадку будуть мати наступний вигляд (див. рис. 3.7).

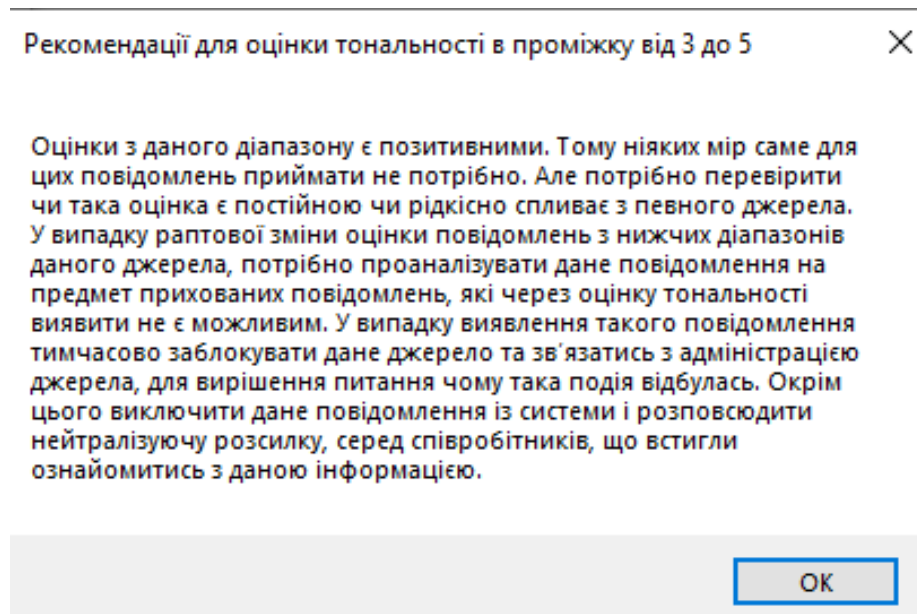


Рисунок 3.7 – Рекомендації для оцінки в проміжку [3;5]

Після проектування інтерфейсу наступним кроком є розробка бази даних, яка є основою розроблюваного програмного засобу. Процес виконання даного кроку відображено у наступному підрозділі.

3.4 Розробка бази даних для системи аналізу тональності текстових повідомлень

Як неодноразово зазначалося вище, програма опирається на використання двох типів словників: окремих слів та словосполучень із двох слів. Відповідно, у базі даних реалізуємо дві таких таблиці, що пристосовані для збереження відповідної інформації, а також таблицю Languages, що містить назви мов, які ураховуються при роботі з програмою, та потрібна для нормалізації усієї бази даних.

Перша таблиця – words, очевидно є більш простою і її структура наведена на рисунку 3.8. Назви полів говорять самі за себе, а тип кожного поля можна бачити з наведеного рисунку. Аналогічно, друга таблиця – wordcombinations, є

трохи складнішою за таблицю слів, і має структуру, що відображена на рисунку 3.9.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	WID	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений
2	WText	varchar(20)	utf8_general_ci		Нет	Нет		Изменить Удалить Обзор уникальных значений
3	WTone	tinyint(4)			Нет	Нет		Изменить Удалить Обзор уникальных значений
4	WWeight	tinyint(4)			Нет	Нет		Изменить Удалить Обзор уникальных значений
5	WLanguage	int(11)			Нет	Нет		Изменить Удалить Обзор уникальных значений

Рисунок 3.8 – Структура таблиці words БД messagetone

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	WCID	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений
2	WCWord1	varchar(20)	utf8_general_ci		Нет	Нет		Изменить Удалить Обзор уникальных значений
3	WCWord2	varchar(20)	utf8_general_ci		Нет	Нет		Изменить Удалить Обзор уникальных значений
4	WCTone	tinyint(4)			Нет	Нет		Изменить Удалить Обзор уникальных значений
5	WCWeight	tinyint(4)			Нет	Нет		Изменить Удалить Обзор уникальных значений
6	WCLanguage	int(11)			Нет	Нет		Изменить Удалить Обзор уникальных значений

Рисунок 3.9 – Структура таблиці wordcombinations БД messagetone

Слід відзначити, що в цілому обидва ці словники можна було б розмістити у текстових файлах, або прописати у тексті програми, але спираючись на можливі майбутні доопрацювання розроблюваного засобу, було використано саме СУБД, яке в подальшому дозволить розширити функціональності програмного засобу у майбутньому. А також уникнувши прописування словників у програмі, програма стала більш адаптивною та менш ресурсоємкою.

Наповнення вказаних таблиць було виконано на базі інформації двох типів:

а) «поганих» слів, що в основному бралися із офіційного словника грубих слів, який використовується у юридичній практиці, за авторством Закірової та були доповнені автором даної роботи;

б) «добрих» слів, взятих із переліку сайту samru.ru, профільтованих та доповнених автором роботи.

Процедура внесення інформації у базу даних була автоматизована:

– вказані словники збережені у форматі CSV (Comma Separated Values);
 – за допомогою описаного вище засобу phpMyAdmin здійснено імпорт цих файлів до бази даних, для чого необхідно просто правильно вказати послідовність розміщення інформації у файлі.

Третя таблиця бази наведена на рисунку 3.10.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	LanguageID	int(11)			Нет	Нет	AUTO_INCREMENT	Изменить Удалить Обзор уникальных значений
2	LanguageName	varchar(32)	utf8_general_ci		Нет	Нет		Изменить Удалить Обзор уникальных значений

Рисунок 3.10 – Структура таблиці languages БД messagetone

Загальна структура розробленої бази даних наведена на рисунку 3.11.

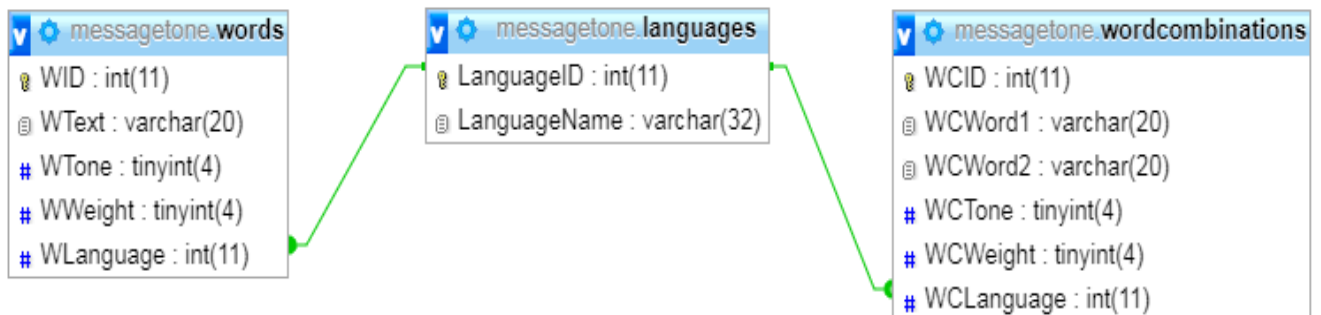


Рисунок 3.11 – Структура розробленої БД messagetone

3.5 Особливості програмної реалізації окремих елементів застосованих алгоритмів

В першу чергу, для кращого розуміння роботи програми, слід розглянути типи даних, що у ній застосовані:

а) структура CWord для збереження інформації із БД про одне слово (призначення полів очевидні із їхніх назв та вказаних типів даних):

```
struct CWord
{
    public string word;
    public int tone;
    public int weight;
};
```

б) структура CCombination для збереження інформації із БД про одне словосполучення (призначення полів також є очевидними):

```
struct CCombination
{
    public string word1;
    public string word2;
    public int tone;
    public int weight;
};
```

в) клас Form1 головного вікна програми:

```
public partial class Form1 : Form
{
    public Form1();
    private void вихідToolStripMenuItem_Click(object sender, EventArgs e);
    private void проПрограмуToolStripMenuItem_Click(object sender, EventArgs e);
    private void відкритиToolStripMenuItem_Click(object sender, EventArgs e);
    private void button1_Click(object sender, EventArgs e);
    private void button2_Click(object sender, EventArgs e);
    private void checkBox1_CheckedChanged(object sender, EventArgs e);
    private void textBox1_TextChanged(object sender, EventArgs e);
}
```

Призначення методів класу також цілком зрозумілі:

- Form1 () - конструктор класу;
- вихідToolStripMenuItem_Click () - обробник вибору пункту меню «Вихід»;
- проПрограмуToolStripMenuItem_Click () - обробник вибору пункту меню «Про програму...» (тут модальним чином викликається вікно «Про програму»);
- відкритиToolStripMenuItem_Click () - обробник вибору пункту меню «Відкрити...» (тут відкривається діалогове вікно для відкриття текстового файлу);
- button1_Click () - обробник натискання на кнопку «Очистити»;
- button2_Click () - обробник натискання на кнопку «Аналіз»;
- checkBox1_CheckedChanged () - обробник натискання на прапорець «Окремі слова», а також на прапорець «Словосполучення»

(тут перевіряється, щоб хоча б один із прапорців був установлений, інакше кнопка «Аналіз» стає неактивною);

– `textBox1_TextChanged()` – обробник події зміни текстового наповнення у верхньому текстовому полі для введення повідомлення (тут перевіряється довжина повідомлення, і якщо вона менше 1 символу, то кнопка «Аналіз» також стає неактивною).

Відзначимо, що повідомлення довжиною 1 символ принципово також може нести певну тональність (хоча і з дуже малою достовірністю), наприклад, якщо містить символ «)», або «(».

Уся основна алгоритмічна робота здійснюється у функції `button2_Click()`, і вона повністю відповідає схемі рисунку 2.3.

3.6 Розробка документації щодо експлуатації програмного засобу

Для належного функціонування будь-якого технічного продукту його експлуатація має відбуватися виключно в межах документації, що на нього поставляється. Для програмних продуктів таким документом, в першу чергу, є інструкція користувача (або керівництво користувача, Manual), яку приведемо у вигляді наступного списку.

1) Завантажити програмне забезпечення, для чого традиційно слід двічі натиснути мишею на відповідному значку.

2) Задати текстове інформаційне повідомлення, що підлягає аналізу його тональності, що можна реалізувати трьома способами:

а) безпосередньо набрати його у верхньому текстовому полі;

б) вставити текст у верхнє текстове поле із буферу обміну;

в) відкрити текстовий файл із меню «Файл»-«Відкрити...».

3) Обрати необхідні налаштування процесу аналізу, а саме впевнитися, що прапорці у нижній частині вікна програми установлені правильно (відповідно до типу аналізу, який бажає використати користувач).

4) Натиснути кнопку «Аналіз» та дочекатися завершення процесу, що може зайняти декілька секунд (або навіть і декілька хвилин, якщо текст дуже великий, а саме має обсяги художнього роману).

5) Прочитати отримані рекомендації та їх застосувати, при необхідності додати додаткові кроки при узгодженні з відповідальними за безпеку особами.

характеристики тексту у великому нижньому текстовому полі та за необхідності скопіювати їх у окремий файл.

6) Переглянути у великому нижньому текстовому полі, які саме слова та словосполучення вплинули на формування данної оцінки, за необхідності скопіювати їх у окремий файл.

7) За необхідності продовження роботи з програмою слід натиснути кнопку «Очистити».

8) За необхідності переглянути відомості про програму, слід вибрати пункт у головному меню «Допомога»-«Про програму...».

9) За необхідності повторювати пункти 2-7 потрібну кількість разів.

10) Закрити програму.

Слід пам'ятати, що для коректної роботи програми на комп'ютері має бути встановлений сервер баз даних MySQL (із базою messagetone, що містить наповнені словами та словосполученнями дві таблиці даних, відповідно до рисунків 3.7 та 3.8), а також драйвер ODBC, встановлення якого детально описувалося у пункті 3.2.3. Відповідно до цих вимог має будуватися і інструкція адміністратора системи, суть якої полягає у перевірці наявності цих програмних продуктів, їх доступності користувачам (причому з його обліковими даними іменем та паролем) та наявності їх необхідного наповнення (відповідного як за структурою, так і за вмістом). Також до інструкцій адміністратора повинно входити, навчання всього персоналу організації, щодо можливого деструктивного інформаційного впливу та методи уникнення данного впливу. Такого типу інструктажі, повинні проводитись не рідше одного разу на місяць та включати в себе інформацію про нещодавно виявлені загрози даного напрямлення.

Таким чином, у даному розділі описані особливості виконання програмної реалізації засобу визначення тональності інформаційних повідомлень, а саме:

- обрано та обґрунтовано вибір об'єктно-орієнтованої технології розробки програмного забезпечення;
- обрано сучасну популярну мову програмування C# та середовище розробки – Microsoft Visual Studio;
- обрано СУБД MySQL як основний засіб для збереження словників, необхідних для роботи програми;
- спроектовано графічний інтерфейс користувача, розробленого програмного засобу;
- розроблено структуру та описано наповнення бази даних, необхідної для роботи продукту;
- описано особливості виконання програмної реалізації даного програмного засобу;
- розроблено базове документаційне забезпечення програмного продукту, а саме інструкцію користувача по роботі з програмним продуктом;

4 ТЕСТУВАННЯ СИСТЕМИ

4.1 Формування деструктивного повідомлення

Для повноцінного тестування необхідно створити п'ять текстових повідомлень. Оцінки тональності даних текстів повинні приймати певні значення, які будуть належати одному з проміжків визначеного діапазону.

Отже, перейдемо до створення першого тексту значення тональності, якого буде знаходитись в першому проміжку [-5; -3). Створимо дане повідомлення, у вигляді «випадкового» листа на поштову скриньку або месенджер. Воно буде нести деструктивний вплив, як на настрій певної особи (осіб) так і на співдружність усього колективу підприємства в цілому.

Кінцевий зміст «Тексту №1» має наступний вигляд: «Серед співробітників фінансового відділу розвелось, так багато всілякого бидла і недоумків. Коли до них звертаєшся, вони постійно хамлять та кажуть розбиратись самим зі своїми питаннями. Також ходять слухи, що начальник даного відділу намагався приставати до директорки компанії за, що вона його прозвала « похотливим кобелем » та відшила цього альфонса. Якщо ця подія дійсно була, то ясно, чого там такий мерзотний колектив, як кажуть «риба гниє з голови». В майбутньому раджу уникати співпраці з ними.»

Даний лист може бути надіслано, як персонально, так і групі (відділу) співробітників.

Переходимо до створення наступного повідомлення, значення тональності якого належатиме діапазону [-3; -1). В даному випадку здійснимо спробу деструктивного впливу на підґрунті різних політичних поглядів. Даний вплив буде реалізовано у вигляді повідомлення, яке матиме назву в даному тестуванні «Текст №2».

«Останнім часом по всім новинам відомі журналюги піднімають провокаційні питання та безсоромно їх обговорюють. Окрім внутрішніх справ у країні, вони часто піднімають питання про політичні кроки президента.

Теперішній президент, нічим не кращий попереднього, алкоголіка. Прийшовши до влади за декілька років створив власну політичну мафію, зібрав збрід навколо себе, зі своїх близьких взяточників та жлобів. Більшість з них некомпетентні в сферах, де займають високі посади, через, що інакше, як мерзавцями та пройходами їх не назвати.»

Повідомлення «Текст №3» напишемо так, щоб його тональність знаходилась в межах $[-1; 1)$, тобто оцінка була нейтральна. Це повідомлення може використовуватись лише індивідуально, тобто спрямоване на певну особу.

«Привіт. Пригадуєш ми з тобою, якось обговорювали нашого нового директора. Казали, що в нього жахлива фізична форма, бо він відростив брюхо і заплив жиром. А ще, про його манери, з приходом вів себе як індюк надутий та й взагалі був, як ідіот. До жіночого колективу відносився різко та визиваюче, як паразит майже до всіх чіплявся. Чоловіків побоювався, тому вів себе обережно. Але зараз перед моїми очима зовсім інша людина, фізична форма стала більш підтягнутою, привабливою та презентабельною. Він налагодив зі всім колективом добрі та дружні відносини, хоча це точно йому легко зробити, після його фальстарту. А в тебе, які думки на цей рахунок?»

Далі продемонструємо мимовільне вкидання, яке надалі може призвести до дестабілізації системи. Даний текст демонструватиме неповноцінність розгляду лише оцінки тональності для виявлення деструктивного впливу. Тобто оцінка буде знаходитись в прийнятних межах $[1; 3)$, але деструктивний вплив у «Тексті №4» буде задіяно.

«Всім співробітникам відомо, що у нашому буфеті готують розкішні та апетитні вкусняшки. Їхня головна « вишенька » серел страв, це божественні десерти, особливо смачні мармеладки та карамельки. Весь їх персонал, це приємні та симпатичні жіночки. Але також всі знають, що відділ розробок постіно забирає найсвіжіші та найсмачніші десерти і керівництво, це підтримує. Однак перші і другі страви завжди в наявності та мають відмінний смак. За їхню працю, нам необхідно бути вдячними шановним кухарям))) »

Необхідно, щоб тональність повідомлення під назвою «Текст №5» знаходилася в діапазоні [3; 5]. Для цього в тексті буде використано лексеми з найбільш позитивною емоційною забарвленістю. Створимо його у вигляді описання одного з позитивних співробітників підприємства.

«Наша красуня з відділу продажів останнім часом стала виглядати сонячною, позитивною та щасливою. Це може бути пов'язано з тим, що вона нещодавно вийшла заміж. Її чоловік виглядає статним та заможним. Завдяки їй кожного дня у співробітників, що з нею контактують піднімається настрій, після чого вони ходять радісні та щасливі. Ця джоймочка в похмурі дні, як бджілка розпилює нектар приємного дня. Коли в тебе поганий настрій спробуй поспілкуватись з нею, він точно зміниться, я тобі це гарантую.»

Усі створені повідомлення було протестовано на розробеній системі. Результати, що були отримані підсистемами відобразимо в наступних підрозділах.

4.2 Обробка виявленого деструктивного повідомлення

«Текст №1» було пропущено через систему, результат отриманий підсистемою виявлення відображено на рисунку 4.1.

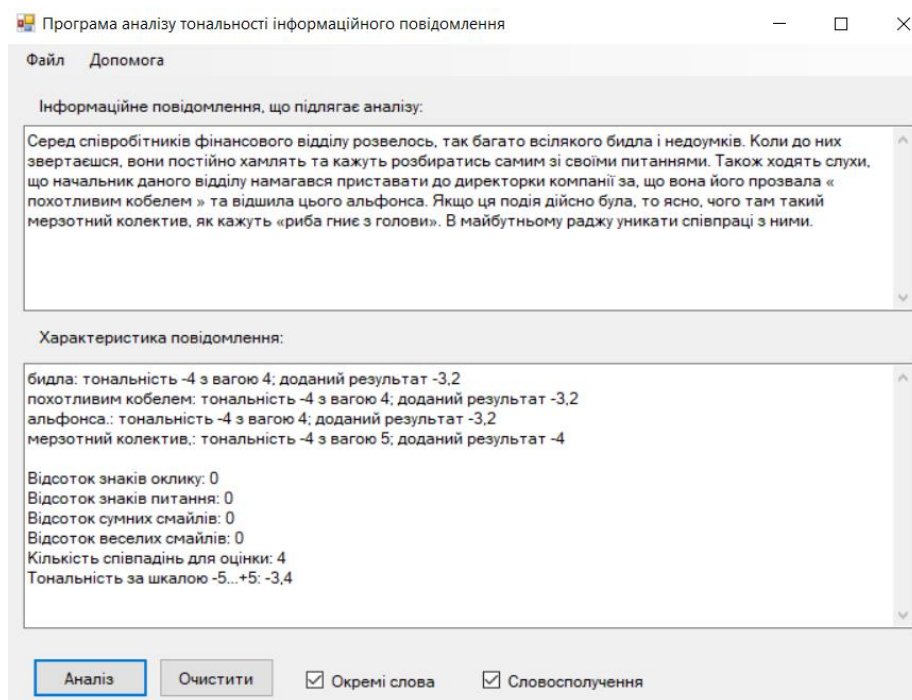


Рисунок 4.1 – Оцінка тональності повідомлення «Текст №1»

Як бачимо повідомлення потрапило у проміжок, який нам був потрібен. Що ж можна переходити до наступного повідомлення «Текст №2», значення тональності, якого повинно знаходитись в межах [-3; -1) (рис. 4.2).

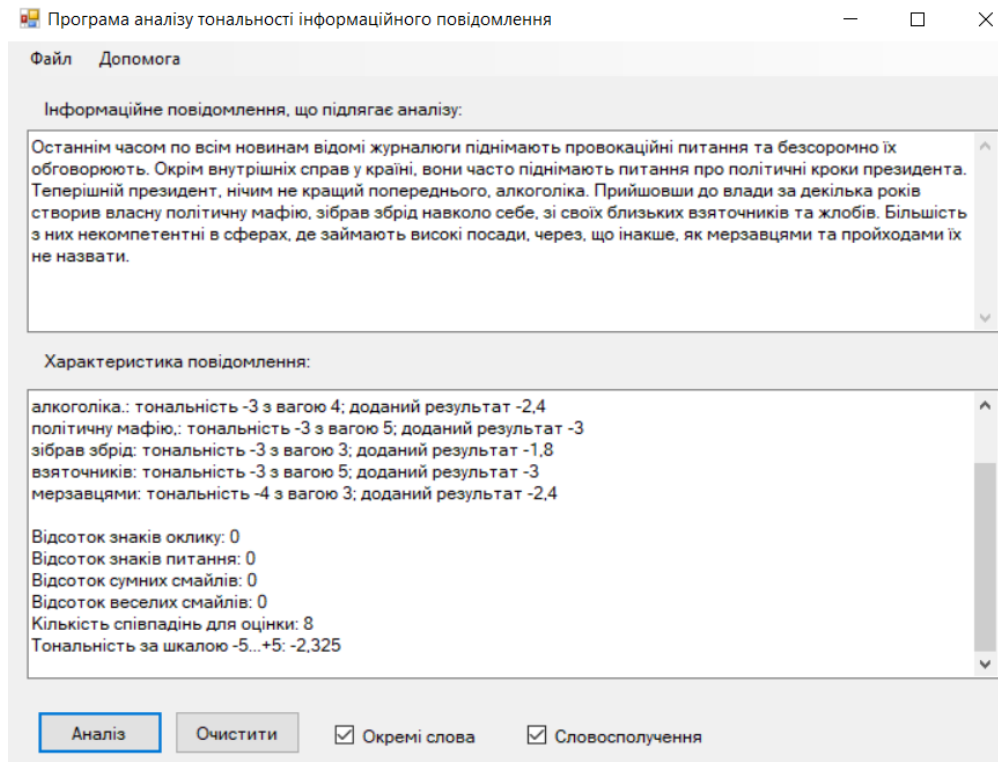


Рисунок 4.2 – Оцінка тональності повідомлення «Текст №2»

Результат нас задовільняє, тому одразу переходимо до наступного повідомлення «Текст №3».

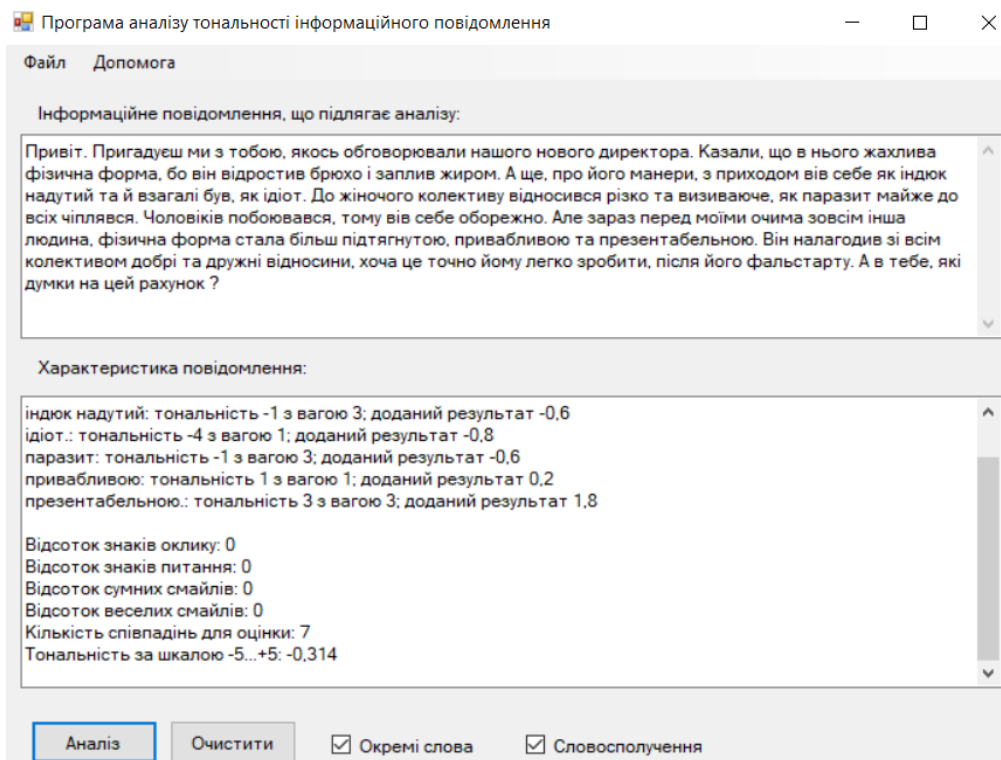


Рисунок 4.3 – Оцінка тональності повідомлення «Текст №3»

Від підсистеми виявлення отримано бажаний результат, тому протестуємо наступне повідомлення.

Результат тестування повідомлення «Текст №4», як і було запланована, потрапив до діапазону [1; 3) (рис. 4.4).

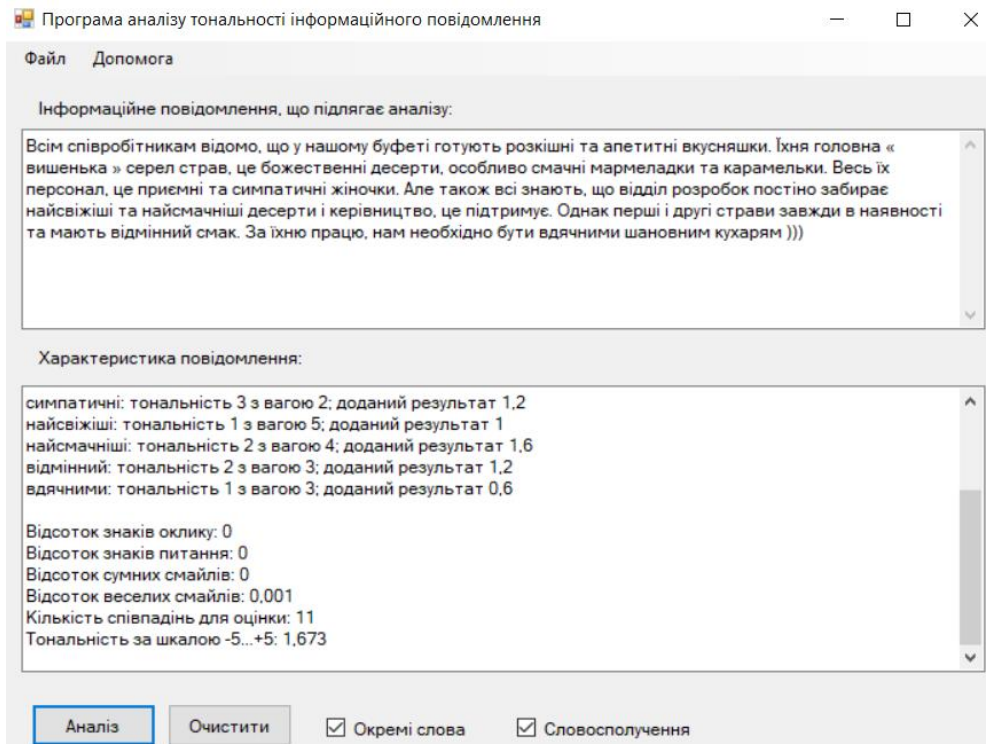


Рисунок 4.4 – Оцінка тональності повідомлення «Текст №4»

Залишилось отримати оцінку для останнього повідомлення «Текст №5». Результат оцінки наведено на рисунку 4.5.

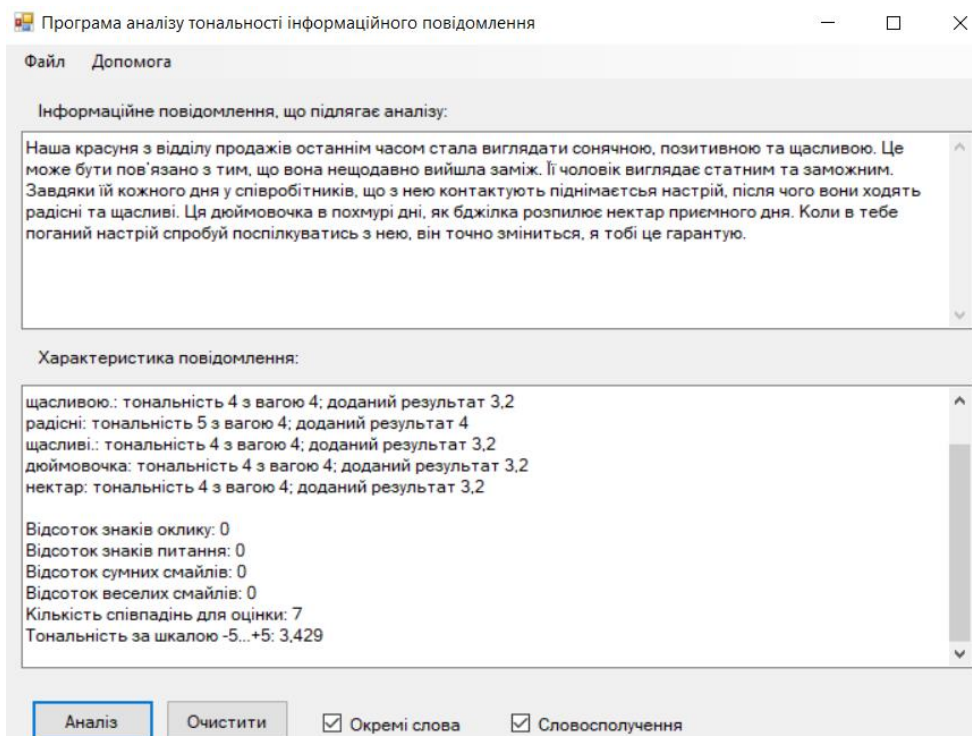


Рисунок 4.5 – Оцінка тональності повідомлення «Текст №5»

Отримані результати говорять, про коректність роботи підсистеми виявлення. Переконаймося у відповідності рекомендацій підсистеми протидії для отриманих оцінок.

4.3 Протидія впливу

Підсистема протидії реалізована у вигляді рекомендацій (приблизна послідовність дій, що необхідно виконати). Для першого повідомлення отримано рекомендації, що наведені на рисунку 4.6.

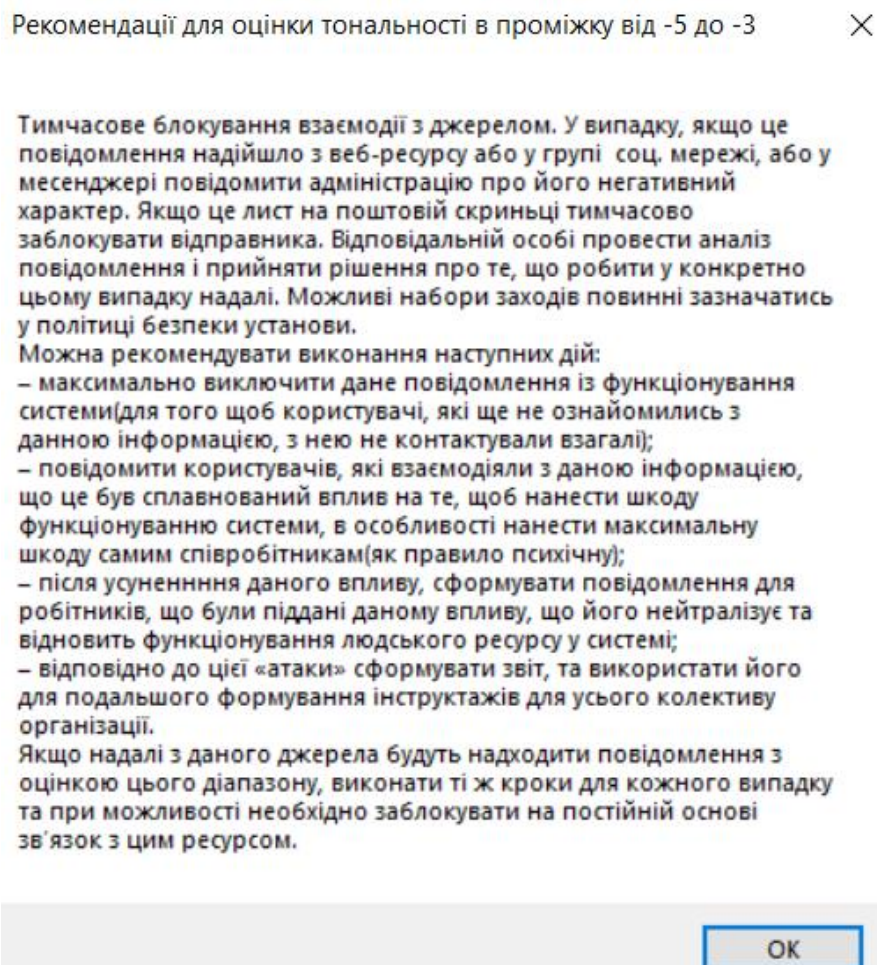


Рисунок 4.6 – Рекомендації для протидії впливу повідомлення «Текст №1»

Для нашого повідомлення, як бачимо рекомендації є доречними. Результат роботи підсистеми протидії, що було отримано при аналізі «Тексту №2» відображено на наступному рисунку (див. рис. 4.7).

Рекомендації достатньо точно описують кроки, для протидії реалізованому у повідомленні впливу. Тож їх виконання нейтралізує даний вплив. Рисунок 4.8 відображає рекомендації для протидії впливу повідомлення «Текст №3».

Рекомендації для оцінки тональності в проміжку від -3 до -1



Рекомендації для оцінки тональності в проміжку від -1 до 1



Віднести джерело з якого надійшло повідомлення до підозрілих. Тобто необхідно провести аналіз повідомлень, що надходили з даного ресурсу та будуть надходити найближчим часом (наприклад на протязі тижня). Якщо оцінка буде знаходитись в цьому ж діапазоні, відповідальній особі необхідно прийняти рішення з приводу чи є суттєво необхідним надходження інформації від даного джерела. Відповідно, провести більш глибокий аналіз вхідних повідомлень за певний проміжок часу:

- 2 - 3 тижні, якщо частота надходження інформації не дуже велика, знаходиться в діапазоні 1 - 3 повідомлення в день;
- 1 - 2 тижні, якщо частота надходження інформації не дуже велика, знаходиться в діапазоні 4 - 5 повідомлення в день;
- тиждень, якщо частота надходження інформації більша ніж 6 повідомлень в день.

При отриманні результатів оцінки з такими самими значеннями та прийнятним змістом даних повідомлень, зробити позначку, що дана оцінка є прийнятною для даного ресурсу. Та зрозуміти, що немає потреби у постійному аналізі фахівцем повідомлень з даного ресурсу з оцінкою, що знаходиться в даному діапазоні. А у випадку зміни оцінки потрібно насторожитись та проаналізувати чому оцінка була змінена.

Якщо оцінка постійно змінюється, як у бік більш негативної оцінки (це трапляється доволі часто) так і у бік більш позитивної оцінки, з'ясувати з чим пов'язані дані зміни. При необхідності, або при отриманні небажаних результатів припинити обмін з даним джерелом.

У випадку, якщо було виявлено, що дане повідомлення або не тільки одне повідомлення, а й усі (більшість) повідомлення з даного джерела містять деструктивний вплив зробити інформативну розсилку серед співробітників, що контактували з данною інформацією. Дана розсилка повинна роз'яснити співробітникам, що це були повідомлення, які мали на меті дестабілізувати роботу системи шляхом впливу на них (людський ресурс). А також розповсюдити у системі нейтралізуюче повідомлення, для приведення функціонування системи до норми.

OK

Рисунок 4.7 – Рекомендації для протидії впливу повідомлення «Текст №2»

Отримані рекомендації для протидії «Тексту №3» є доречними та цілком обґрунтовані. Перейдемо до тестування передостаннього повідомлення «Текст №4», результат відображено на рисунку (див. рис. 4.9).

Рекомендації для оцінки тональності в проміжку від 1 до 3



Така оцінка є прийнятною. Головне, щоб оцінка повідомлень джерела постійно знаходилась в своєму діапазоні, у такому випадку ніяких дій застосовувати до даного каналу віщання. Якщо оцінка даного джерела постійно змінюється необхідно провести змістовний аналіз даних повідомлень та з'ясувати з чим це пов'язано. У випадку впливу на цю зміну декількох несуттєвих фраз, то можна віднести дане джерело до декількох прийнятних діапазонів. Якщо зміни діапазону спостерігаються дуже рідко, то необхідно проводити аналіз саме тих повідомлень, які виходять за межі встановленого для даного джерела діапазону значень оцінки тональності.

OK

Рисунок 4.9 – Рекомендації для протидії впливу повідомлення «Текст №4»

Хоча, судячи з оцінки, деструктивного впливу немає, але згідно отриманих рекомендацій, при огляді компетентною особою, даний вплив буде виявлено і попереджено його поширення у системі.

Заключне тестування підсистеми протидії, проведлось шляхом перевірки рекомендацій для «Тексту №5». Рекомендації, що були визначені підсистемою наведені на рисунку нижче (див. рис. 4.10).

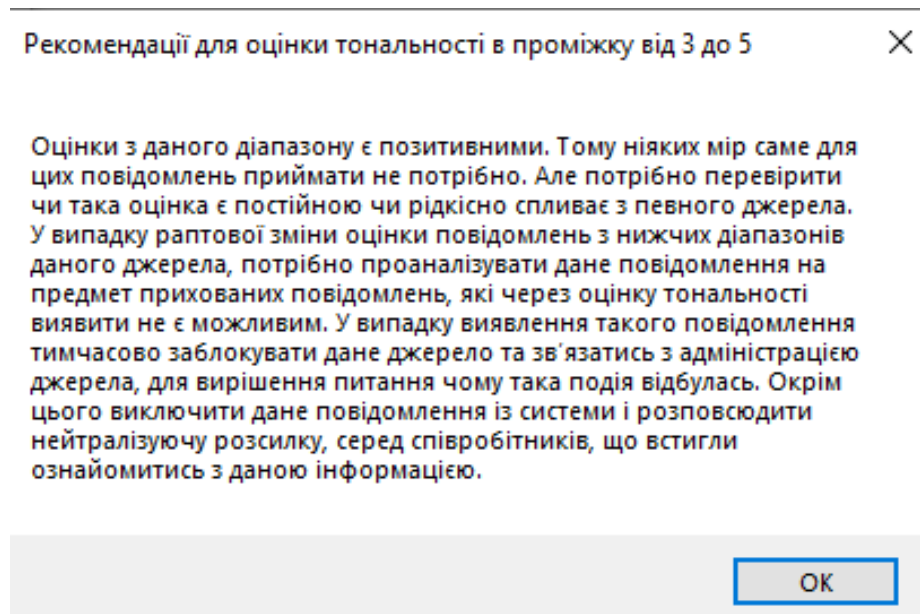


Рисунок 4.10 – Рекомендації для протидії впливу повідомлення «Текст №5»

Отримані результати є коректними для даної вибірки. Тому можна вважати, що тестування системи пройшло успішно та свідчить про її справну роботу.

5 ЕКОНОМІЧНА ЧАСТИНА

Метою економічної частини магістерської кваліфікаційної роботи є обґрунтування економічної доцільності розробки системи оцінювання тональності інформаційного повідомлення. Для цього необхідно виконати такі етапи робіт:

- оцінити комерційний потенціал розробки;
- спрогнозувати витрати на виконання наукової роботи та впровадження її результатів;
- спрогнозувати комерційний ефект від реалізації результатів розробки;
- розрахувати ефективність вкладених інвестицій та період їх окупності.

5.1 Оцінювання комерційного потенціалу розробки (технологічний аудит розробки)

Об'єктом дослідження магістерської кваліфікаційної роботи є метод та засіб визначення фішингових інформаційних ресурсів.

Для проведення технологічного аудиту було залучено трьох незалежних експертів: Лужецький Володимир Андрійович, Дудатьєв Андрій Веніамінович, Войтович Олеся Петрівна. Лужецький Володимир Андрійович – д.т.н., проф., зав. кафедри ЗІ, Дудатьєв Андрій Веніамінович – к.т.н, доц. кафедри ЗІ, Войтович Олеся Петрівна – к.т.н, доц. кафедри ЗІ. Кожен з експертів повинен ознайомитися з запропонованою розробкою та заповнити таблицю, яка визначає рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можливу оцінку в балах. Після виконання цього, підраховується середньоарифметична сума балів та визначається який рівень комерційного потенціалу має нова розробка.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ю критеріями, наведеними в таблиці 5.1.

Таблиця 5.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 5.2.

Таблиця 5.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали експерта		
	Лужецький В. А.	Дудатьєв А. В.	Войтович О. П.
	Бали, виставлені експертами:		
1	2	3	3
2	4	4	4
3	3	3	3
4	3	4	3
5	4	3	4
6	4	4	4
7	3	4	3
8	3	3	3
9	3	3	4
10	4	4	4
11	4	3	4
12	4	4	4
Сума балів	СБ ₁ =41	СБ ₂ =42	СБ ₃ =43
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^n СБ_i}{i} = \frac{41 + 42 + 43}{3} = 42$		

Отже, з отриманих даних таблиці 5.2 видно, що середньоарифметична сума балів дорівнює 42, тобто нова розробка має рівень комерційного потенціалу вище середнього.

На ринку майже не існує систем оцінювання тональності інформаційного повідомлення. Тому гостра необхідність в розробці данної системи. Також існуючі системи мають недоліки пов'язані з не наданням користувачу рекомендацій для протидії деструктивному впливу, якщо він був виявлений. А за результатами проведеного дослідження розроблена система позбавлена цих недоліків.

5.2 Прогнозування витрат на виконання науково-дослідної та конструкторсько-технологічної роботи

Прогнозування витрат на виконання науково-дослідної роботи, дослідно-конструкторської та конструкторсько-технологічної роботи може складатися з таких етапів:

1. Розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;
2. Розрахунок загальних витрат на виконання роботи даної роботи;
3. Прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Розрахунок витрат на розробку методики дослідження здійснюється по статтях калькуляції, а саме:

- основна заробітна плата розробників;
- додаткова заробітна плата розробників;
- нарахування на заробітну плату розробників;
- амортизація персонального комп'ютеру;
- витрати на матеріали, що були використані в процесі розробки програмного продукту;
- витрати на електроенергію;

– інші витрати.

У розробці даного програмного продукту брав участь один спеціаліст та один науковий керівник. Основна заробітна плата для спеціаліста визначається за формулою:

$$Z_o = \frac{M}{T_p} t$$

де M – місячний посадовий оклад конкретного розробника, наукового керівника;

T_p – кількість робочих днів у місяці, $T_p = 22$ дні;

t – число днів роботи.

Оплата за робочий день розробника проекту становить 750 грн, а наукового керівника – 600 грн.

Розрахунки заробітної плати для спеціаліста наведені в таблиці 5.3

Заробітна плата наукового керівника:

$$Z_{окк} = \frac{13200}{22} \times 22 = 600 \times 22 = 13200 \text{ (грн)}$$

Заробітна плата розробника:

$$Z_{ор} = \frac{16500}{22} \times 52 = 750 \times 52 = 39000 \text{ (грн)}$$

Витрати на оплату праці, основна заробітна плата:

$$Z_o = Z_{окк} + Z_{ор} = 13200 + 39000 = 52200 \text{ (грн)}$$

Таблиця 5.3 – Розрахунки основної заробітної плати спеціаліста

Працівник	Оклад М, грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник проекту	13200	600	22	13200
Розробник	16500	750	52	39000
Всього:				52200

Розрахуємо додаткову заробітну плату. Додаткова заробітна плата Z_d розраховується як 10...12% від суми основної заробітної плати, тобто:

$$Z_d = (0,1...0,12) \times Z_o$$

Додаткова заробітна плата:

$$Z_d = 0,1 \times (Z_{онк} + Z_{оп}) = 0,1 \times (13200 + 39000) = 5220 \text{ (грн)}$$

Нарахування на заробітну плату $H_{зп}$ розраховується як 22% від суми основної та додаткової заробітної плати:

$$H_{зп} = (Z_o + Z_d) \times \frac{\beta}{100}$$

де Z_o – основна заробітна плата розробників, грн.;

Z_d – додаткова заробітна плата розробників, грн.;

β – ставка єдиного внеску на загальнообов'язкове державне страхування.

Єдиний соціальний внесок на загальнообов'язкове державне соціальне страхування (ЄСВ) – об'єднаний страховий внесок в Україні, збір якого здійснюється в системі загальнообов'язкового державного страхування в обов'язковому порядку та на регулярній основі. Відповідно до даних офіційного сайту міністерства фінансів в 2021 році ЄСВ становить 22%.

$$H_{зп} = (52200 + 5220) \times \frac{22}{100} = 57420 \times 0,22 = 12632,4 \text{ (грн)}$$

Розрахунок амортизаційних витрат для комп'ютера вартістю 25000 грн. виконується за такою формулою:

$$A = \frac{Ц \times H_A}{100} \times \frac{T}{12}$$

де $Ц$ – балансова вартість обладнання, грн;

H_A – річна норма амортизаційних відрахувань. Для даного випадку можна прийняти, що $H_A = 25\%$;

T – термін використання ($T=3$ міс.);

T_B – корисний час використання (T_B для комп'ютера становить 3 роки).

Отже, розрахуємо амортизаційні відрахування:

$$A = \frac{25000 \times 25}{100} \times \frac{3}{12} = \frac{625000}{100} \times 0,25 = 6250 \times 0,25 = 1562,5 \text{ (грн)}$$

Витрати на матеріали розраховуються за формулою:

$$M = \sum_1^n H_i \times C_i \times K_i - \sum_1^n B_i \times C_B$$

де H_i – витрати матеріалу i -го найменування;

C_i – вартість матеріалу i -го найменування, грн/кг;

K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$;

V_i – маса відходів матеріалу i -го найменування, кг;

C_v – ціна відходів матеріалу i -го найменування, грн/кг;

n – кількість видів матеріалів.

Вартість матеріалів, що були використані на розробку системи зведені до таблиці 5.4.

$$M = 355,1 \times 1,1 = 390,61 \text{ (грн)}$$

Таблиця 5.4 – Вартість матеріалів, що були використані для розробки алгоритму тестування.

Найменування матеріалу	Одиниці виміру	Ціна, грн	Витрачено	Вартість витрачених матеріалів, грн.
Флеш-накопичувач	Шт.	150	1	150
Пачка паперу	Уп	95	2	190
Ручка	Шт	7,55	2	15,1
Всього				355,1

Таким чином, витрати на матеріали, що були використані в процесі розробки алгоритму тестування складають $M = 390,61$ грн.

Розрахуємо витрати на комплектуючі.

Витрати на силову електроенергію розраховуються за формулою:

$$V_E = V \times \Pi \times \Phi \times K_{\Pi}$$

де V – вартість 1кВт-години електроенергії ($V=4,62$ грн/кВт);

Π – установлена потужність комп'ютеру ($\Pi=0,03$ кВт);

Φ – фактична кількість годин роботи комп'ютеру ($\Phi=8$ год \times 74 дня);

K_{Π} – коефіцієнт використання потужності ($K_{\Pi} < 1$, $K_{\Pi} = 0,8$).

Потужність ноутбука складає 30 Вт/год = 0,3кВт/год + потужність на освітлення, тоді $\Pi = 0,04$ кВт/год

$$V_E = 4,62 \times 0,04 \times (8 \times 74) \times 0,8 = 88,97 \text{ (грн)}$$

Розрахуємо інші витрати $V_{ін}$.

Інші витрати $V_{ін}$ можна прийняти як (100-300)% від суми основної заробітної плати розробників, які виконували дану роботу, тобто:

$$V_{\text{ін}} = (1 \dots 3) \times (3_o + 3_p).$$

Отже, розрахуємо інші витрати:

$$V_{\text{ін}} = 1 \times (57420) = 57420 \text{ (грн)}.$$

Усі витрати складають:

$$B = 52200 + 5220 + 12632,4 + 1562,5 + 390,61 + 88,97 + 52200 = 124294,48 \text{ (грн)}$$

Розрахуємо загальну вартість наукової розробки $B_{\text{заг}}$ за формулою:

$$B_{\text{заг}} = \frac{B}{\alpha}$$

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

$$B_{\text{заг}} = \frac{124294,48}{1} = 124294,48 \text{ (грн)}$$

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

$$ЗВ = \frac{B_{\text{заг}}}{\beta}$$

Отже, розрахуємо прогнозовані загальні витрати:

$$ЗВ = \frac{124294,48}{0,7} = 177563,54 \text{ (грн)}$$

5.3 Прогнозування комерційних ефектів від реалізації результатів розробки

В економічному розділі магістерської кваліфікаційної роботи обґрунтовується економічна доцільність розробки системи оцінювання тональності інформаційного повідомлення. На виконання усіх необхідних робіт необхідно 52 робочих днів.

Зростання чистого продукту для даного засобу можна оцінити у теперішній вартості грошей. Зростання чистого прибутку забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \times N + \Pi_{\text{я}} \Delta N)_i$$

де $\Delta\Pi_{\text{я}}$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

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

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

$\Pi_{\text{я}}$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

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

В результаті впровадження результатів наукової розробки витрати на розробку алгоритму зменшаться на 400 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 400 грн), а кількість користувачів збільшиться: протягом першого року – на 100 користувачів, протягом другого року – на 140 користувачів, протягом третього року – на 80 користувачів.

Реалізація продукції до впровадження результатів наукової розробки складала 500 шт., а прибуток, що його отримувало підприємство (організація) на одиницю продукції до впровадження результатів наукової розробки – 450 грн.

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

Отже, збільшення чистого продукту $\Delta\Pi_1$ протягом першого року складатиме:

$$\Delta\Pi_1 = 400 \times 500 + (450 + 400) \times 100 = 200000 + 85000 = 285000 \text{ (грн)}$$

Протягом другого року:

$$\Delta\Pi_2 = 400 \times 500 + (450 + 400) \times (100 + 140) = 200000 + 850 \times 240 = 404000 \text{ (грн)}$$

Протягом третього року:

$$\Delta\Pi_3 = 400 \times 500 + (450 + 400) \times (100 + 140 + 80) = 200000 + 850 \times 320 = 472000 \text{ (грн)}$$

5.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

1-й крок. Розрахуємо теперішню вартість інвестицій PV, що вкладаються в наукову розробку. Такою вартістю ми можемо вважати прогнозовану величину загальних витрат ЗВ на виконання та впровадження результатів НДДКР, розраховану раніше, тобто будемо вважати, що $ZB = PV = 177563,54$ грн.

2-й крок. Розрахуємо очікуване збільшення прибутку $\Delta\Pi_i$, що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку також було розраховане нами раніше та становить:

$$\Delta\Pi_1 = 285000 \text{ грн}, \quad \Delta\Pi_2 = 404000 \text{ грн}, \quad \Delta\Pi_3 = 472000 \text{ грн}.$$

3-й крок. Для спрощення подальших розрахунків необхідно побудувати вісь часу, на яку наносять всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Якщо загальні витрати ЗВ на виконання та впровадження результатів НДДКР (або теперішня вартість інвестицій PV) дорівнюють 177563,54 грн., а результати вкладених у наукову розробку інвестицій почнуть виявлятися вже в кінці другого року впровадження. То ці результати виявляться у тому, що у першому році підприємство отримає збільшення чистого прибутку на 285000 грн. відносно базового року, у другому році – збільшення чистого прибутку на 404000 грн (відносно базового року), у третьому році – збільшення чистого прибутку на 472000 грн (відносно базового року).

Тоді рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рис. 5.1.

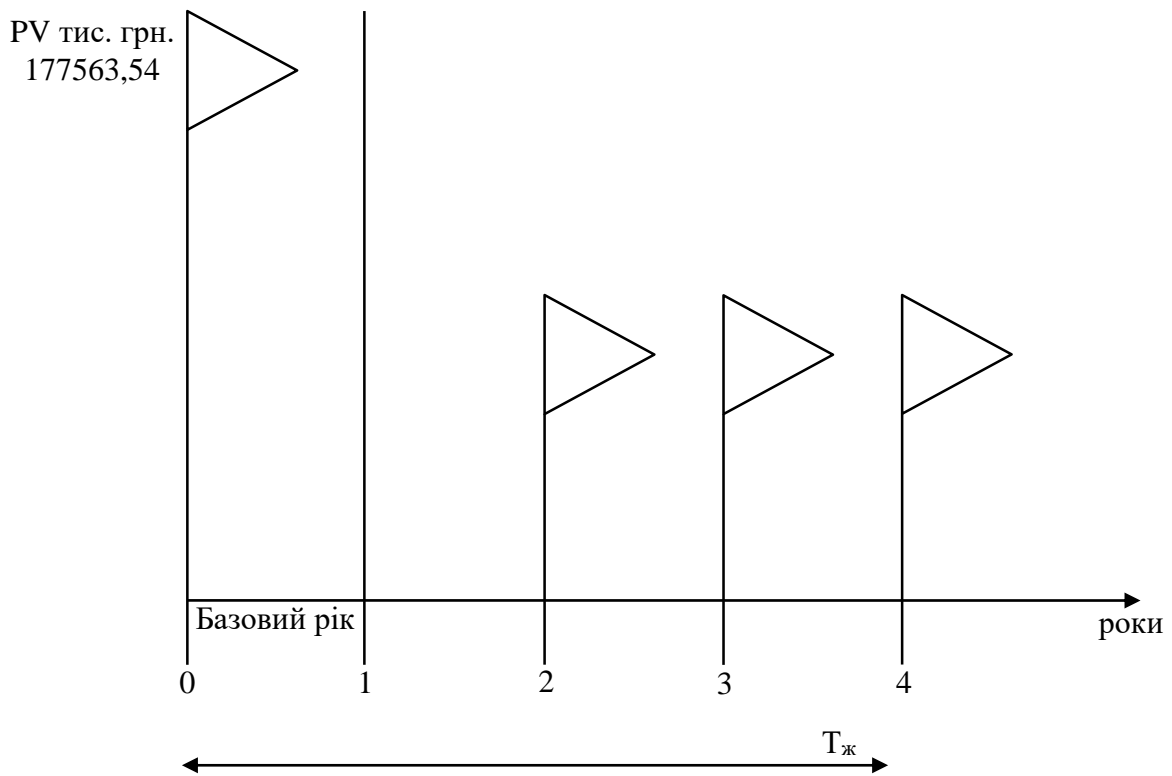


Рисунок 5.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

4-й крок. Розрахуємо абсолютну ефективність вкладених інвестицій $E_{абс}$. Для цього скористаємося формулою:

$$E_{абс} = (ПП - PV)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій $PV = ЗВ = 177563,54$ грн.

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)}$$

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

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

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

t – період часу (в роках) від моменту отримання чистого прибутку до точки «0».

$$ПП = \frac{285000}{(1+0,1)^2} + \frac{404000}{(1+0,1)^3} + \frac{472000}{(1+0,1)^3} = \frac{285000}{1,21} + \frac{404000}{1,331} + \frac{472000}{1,4641} = 235537,19 + 303531,18 + 322382,35 = 861450,72 \text{ (грн)}$$

$$E_{abc} = 861450,72 - 177563,54 = 683887,18 \text{ (грн)}$$

Оскільки $E_{abc} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР може бути доцільним.

5-й крок. Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_B . Для цього використаємо формулу:

$$E_B = T_j \sqrt[T_j]{1 + \frac{E_{abc}}{PV}} - 1,$$

де E_{abc} – абсолютна ефективність вкладених інвестицій, грн; PV – теперішня вартість інвестицій $PV = ZB$, грн; T_j – життєвий цикл наукової розробки, роки.

Далі, розрахована величина E_B порівнюється з мінімальною (бар'єрною) ставкою дисконтування τ_{min} , яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування τ_{min} визначається за формулою:

$$\tau = d + f$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2021 році в Україні $d = (0,14...0,2)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,1)$, але може бути і значно більше.

Якщо величина $E_B > \tau_{min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки. В іншому випадку фінансування наукової розробки здійснюватися не буде.

Спочатку спрогнозуємо величину τ_{min} . Припустимо, що за даних умов

$$\tau_{min} = 0,15 + 0,05 = 0,2.$$

Тоді відносна (щорічна) ефективність вкладених інвестицій в проведення наукових досліджень та впровадження їх результатів складе:

$$E_B = \sqrt[4]{1 + \frac{683887,18}{177563,54}} - 1 = \sqrt[4]{1 + 3,85} - 1 = \sqrt[4]{4,85} - 1 = 1,48 - 1 = 0,48 \text{ або } 48\%.$$

Оскільки $E_B = 48\% > \tau_{min} = 0,4 = 40\%$, то інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

6-й крок. Розраховують термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ можна розрахувати за формулою:

$$T_{ок} = \frac{1}{E_B}$$

Якщо $T_{ок} < 3 \dots 5$ -ти років, то фінансування даної наукової розробки в принципі є доцільним. В інших випадках потрібні додаткові розрахунки та обґрунтування.

$$T_{ок} = \frac{1}{0,48} = 2,1 \text{ рік}$$

$T_{ок} < 3$ років, що свідчить про доцільність фінансування даної наукової розробки.

Рівень комерційного потенціалу системи оцінювання тональності інформаційного повідомлення є вище середнього. Загальні витрати на створення нового програмного продукту склали 177502,47 грн. Абсолютна ефективність капіталовкладень для даної розробки за 3 роки 683948,25 грн. Показники ефективності показують, що даний метод є доцільним і буде цікавий для інвестора. Термін окупності розробленого проекту 2 роки та 1 місяць, що підтверджує доцільність вкладання коштів в дану розробку.

ВИСНОВКИ

Проаналізувавши різні методи оцінювання тональності та протидії деструктивному впливу, було обрано та удосконалено по одному з методів для кожного напрямлення. На основі обраних методів створено систему, що дозволяє в автоматизованому режимі оцінювати емоційне забарвлення тексту та надає рекомендації щодо протидії на вплив, якщо він був виявлений.

Система реалізована у вигляді ПЗ та має свої позитивні та негативні аспекти. З позитивних можна назвати, те, що система працює справно та коректно в межах, поставлених завдань та при низьких оцінках рекомендації є досить доречними. Негативними сторонами є обмеженість самого оцінювання тональності, як інструменту виявлення деструктивного впливу. Обмеженість полягає у тому, що деструктивний може бути реалізовано в короткій формі «вкидання» певної інформації без емоційних забарвлень, але будуть нести значний вплив на свідомість.

Проведене дослідження несе значну практичну цінність для систем захисту, що використовуються в державних закладах та на підприємствах приватної форми власності. А саме його цінність полягає у захисті людської свідомості та функціонування існуючої системи від впливу емоційно забарвлених текстів, який як правило несе деструктивний вплив на зазначені сфери.

Шляхом проведення економічних розрахунків доведено доцільність розробки системи оцінювання тональності інформаційного повідомлення. У ході дослідження визначено термін окупності – 2 роки і 1 місяць.

Дослідження є завершеним в межах даної роботи. Та вже може бути застосовано, як додатковий елемент систем захисту на підприємствах. У процесі використання може доповнюватись словник, за допомогою якого здійснюється оцінювання тональності. А також відповідно до індивідуальних потреб тієї чи іншої організації доповнювати функціонал розробленої системи та безпосередньо готового ПЗ.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Manning Christopher D., Schütze Hinrich. Foundations of Statistical Natural Language Processing // Second printing with corrections. — The MIT Press, 2000. — 680 p.
2. Клековкина М. Метод автоматической классификации текстов по тональности, основанный на словаре эмоциональной лексики. / Клековкина М., Котельников Е.// Переславль-Залесский, Россия: конференция. — 2012.
3. Moor, James H. The Status and Future of the Turing Test.// 2001. – p. 77–93.
4. Vyrva, N. Sentiment Analysis in Social Media. Master’s Thesis in Computer Science / N. Vyrva. – Halden: Ostfold University College, 2016. – 82 p.
5. Вормсбехер В. 100 страниц в час. / Вормсбехер В., Кабин В – Кемерово: Кемеровское книжное издательство, 1976. — 116 с.
6. Дудатьев А.В. Комплексный метод противодействия информационно-психологическим операциям // Проблемы управления и информатики. - Киев: Институт кибернетики им. В. М. Глушкова НАН Украины, 2017. - № 1.
7. Лоза В. Визначення тональності текстової інформації з використанням методу штучних нейронних мереж в задачі виявлення інформаційно-психологічних впливів / Лоза В.М., Лалетін С.П., Дяченко І.М // Матер. ІХ Всеукр. наук.-практ. конф. «Актуальні проблеми управління інформаційною безпекою держави». – К.: Нац. акад. СБУ, 2018. - с.233-235..
8. Wiebe, J. Learning subjective adjectives from corpora / J. Wiebe // In Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence / J. Wiebe., 2000. – p. 735–740.
9. Cortes, C. Support-vector networks. / C. Cortes, V. Vapnik // In Machine Learning / C. Cortes, V. Vapnik. – Hingham: Kluwer Academic Publishers, 1995. – V.20. – p. 273–297.
10. Mitchell, T. Machine Learning / T. M. Mitchell. – New York: McGraw-Hill, 1997. – 414 p.

11. Наивный байесовский классификатор [Электронный ресурс]. : URI: <http://bazhenov.me/blog/2012/06/11/naive-bayes>.
12. Narayanan, V. Fast and accurate sentiment classification using an enhanced naive bayes model. / V. Narayanan, I. Arora, A. Bhatia // Intelligent Data Engineering and Automated Learning IDEAL / V. Narayanan, I. Arora, A. Bhatia. – Berlin: Springer, 2013. – V. 8206 of Lecture Notes in Computer Science. – p.194–201.
13. Support Vector Machines: A Simple Explanation [Электронный ресурс]. : URI: <https://www.kdnuggets.com/2016/07/support-vectormachines-simple-explanation.html>.
14. Logistic Regression [Электронный ресурс]. : URI: <http://faculty.cas.usf.edu/mbrannick/regression/Logistic.html>.
15. Lexicon-based methods for sentiment analysis / [M. Taboada, J. Brooke, M. Tofiloski et al]. // Computational linguistics. – V.37. – p. 267–307.
16. Д.В. Шингалов, Є.В. Мелешко, Р.М. Минайленко, В.А. Резніченко. Методи автоматичного аналізу тональності контенту у соціальних мережах для виявлення інформаційно-психологічних впливів // Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. / Шингалов Д., Мелешко Є., Минайленко Р., Резніченко В. – К.: Кропивницький, 2017. - №30. – с. 196-202.
17. Ткачук Т. Ю. Механізми протидії інформаційним загрозам зовнішніх джерел. // ВІСНИК НТУУ «КПІ». Політологія. Соціологія. Право. – Випуск 1/2 (33/34) – 2017 – с. 242–246.
18. Лужецький В. А., Дудатьєв А. В. Концептуальна модель системи інформаційного впливу // Український науковий журнал з інформаційної безпеки. - 2017. - Т. 23, № 1.
19. Вакуленко Р. В. Огляд та аналіз методів протидії інформаційним впливам супротивникам в умовах інформаційної війни.// Актуальні задачі та досягнення у галузі кібербезпеки. Матеріали Всеукраїнської науково-практичної конференції 23—25 листопада 2016 року, м. Кропивницький. — с. 186—187. [Електронний ресурс]. : URL: <https://core.ac.uk/download/pdf/84825417.pdf>.

20. Дослідження ефективності визначення вагових коефіцієнтів важливості / Медиковський М., Шуневич О. // Вісник Хмельницького національного університету. – 2011. № 5. – с. 176–182.
21. Каліон В. Основи інформатики. Структурне програмування на Паскалі. / Каліон В., Черняк О. // Центр навчальної літератури – 2007. – 248 с.
22. Виссер Дж. Разработка обслуживаемых программ на языке С# // ДМК Пресс, 2017. — 192 с.
23. Chowdhury Kunal. Mastering Visual Studio 2019 // 2nd Edition. — Packt Publishing, 2019 — 374 p.
24. Vanier Eric, Shah Birju, Malepati Tejaswi. Advanced MySQL 8 // Packt Publishing, 2019. — 378 p.
25. Засоби аналізу тональності інформаційного повідомлення/ А. В. Дудатьєв, Д. В. Лиськов // Матеріали XLIX науково-технічної конференції підрозділів ВНТУ, Вінниця, 2020 р.

ДОДАТКИ

Додаток А

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

ЗАТВЕРДЖУЮ

Зав. кафедри ЗІ, д. т. н., проф.

_____ В. А. Лужецький

«_____» _____ 2021 року

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

на тему: «Система оцінювання тональності інформаційного повідомлення»

08-20.МКР.007.00.000 ТЗ

Керівник магістерської кваліфікаційної роботи

доц. каф. ЗІ, к. т. н., доц.

Дудатьєв А. В.

1 Підстави для проведення робіт

Робота проводиться на підставі наказу ректора ВНТУ від 24 вересня 2021 року № 277.

Дата початку роботи 09.09.21 р.

Дата закінчення роботи 18.12.21 р.

2 Мета та призначення НДР

Мета – зменшення частки ручної праці при аналізі повідомлень, а також пришвидшення процесу протидії на джерела інформації з деструктивним впливом, що досягається шляхом розробки ефективної системи для оцінювання тональності інформаційних повідомлень.

Об'єктом дослідження є процеси визначення тональності інформаційного повідомлення та протидії інформаційному впливу.

Предметом дослідження є методи та засоби, що дозволяють здійснювати визначення тональності інформаційного повідомлення та протидії впливу.

Актуальність теми. З кожним днем люди зустрічаються з великими потоками інформації з різноманітних джерел. Джерелами виступають загальновідомі ЗМІ, такі як телебачення, веб-сторінки офіційних новинних каналів, радіовіщання, а також різноманітні інтернет ресурси з чатами (в тому числі і групи соціальних мереж), на яких спільнота обговорює різні тематики.

В зв'язку з ризиком можливого впливу на людину, шляхом представлення тієї чи іншої інформації в певному вигляді виникає необхідність у фільтрації данної інформації. Одним з інструментів, що допомагає автоматизувати даний процес є оцінювання тональності інформаційних повідомлень.

3 Вихідні дані для проведення НДР

Вихідними даними для проведення НДР є:

- 3.1. Дудатьев А.В. Комплексный метод противодействия информационно-психологическим операциям // Проблемы управления и информатики. - Киев: Институт кибернетики им. В. М. Глушкова НАН Украины, 2017. - № 1.
- 3.2. Дудатьев А.В., Лужецкий В.А., Коротаев Д.О. Метод оценки информационной устойчивости социотехнических систем в условиях информационной войны / А.В. Дудатьев, В.А. Лужецкий, Д.А. Коротаев // Восточно-Европейский журнал передовых технологий. – 2016. – № 1.
- 3.3. Дудатьев А.В. Моделі для організації протидії інформаційним атакам / А.В. Дудатьев // Захист інформації. – 2015. – № 2.
- 3.4. Лужецький В. А., Дудатьев А. В. Концептуальна модель системи інформаційного впливу // Український науковий журнал з інформаційної безпеки. - 2017. - Т. 23, № 1.

- 3.5. Ткачук, Т. Ю. Механізми протидії інформаційним загрозам зовнішніх джерел // Вісник НТУУ «КПІ». Політологія. Соціологія. Право : збірник наукових праць. – 2017. – № 1/2 (33/34).
- 3.6. Stone, P. A computer approach to content analysis: Studies using the general inquirer system // Spring Joint Computer Conference, AFIPS '63 (Spring) – New York: ACM, 1963. – p. 241–256.

4 Виконавці НДР

Студент групи ІБС-20м Лиськов Дмитро Васильович

5 Вимоги до виконання НДР

Для пришвидшення процесів оцінювання тональності тесту та реакції на виявлений вплив необхідно розв'язати такі задачі:

- науково-досліджене обґрунтування необхідності дослідження та розробки системи оцінювання тональності інформаційного повідомлення;
- аналіз наукової літератури щодо існуючих методів та засобів визначення тональності тексту;
- удосконалення лексемного методу оцінки тональності, шляхом розгляду не тільки окремих слів, а й словосполучень, що складаються з двох слів;
- розробка алгоритмів функціонування модулів системи;
- розробка системи згідно розроблених алгоритмів функціонування;
- експериментальне дослідження розробленої системи.

6 Вимоги до супровідної документації

Графічна і текстова документація повинна відповідати діючим стандартам України.

7 Етапи НДР

Робота з теми виконується в чотири етапи.

Зміст етапу	Початок	Закінчення	Очікувані результати	Звітна документація
Аналіз завдання. Вступ	09.09.2021	14.09.2021	Вступ	Чернетка вступу
Розробка технічного завдання	15.09.2021	21.09.2021	Технічне завдання	Проект технічного завдання
Аналіз літературних джерел за напрямком магістерської кваліфікаційної роботи	22.09.2021	29.09.2021	Аналіз відомих методів. Постановка завдання.	Чернетка першого розділу
Удосконалення лексемного методу.	30.09.2021	11.10.2021	Удосконалений лексемний метод визначення тональності	Чернетка другого розділу

Зміст етапу	Початок	Закінчення	Очікувані результати	Звітна документація
Створення макету системи.	11.10.2021	01.11.2021	Макет системи для протидії інформаційному впливу	Чернетка третього розділу
Експериментальні дослідження	02.11.2021	07.11.2021	Система, яка реалізує розроблений метод	Чернетка четвертого розділу
Розробка економічного розділу	08.11.2021	15.11.2021	Економічні показники дослідження	Чернетка економічного розділу
Оформлення Пояснювальної записки	16.11.2021	10.12.2021	Пояснювальна записка	Пояснювальна записка

8 Очікувані результати та порядок реалізації НДР

Передбачається розробка удосконалення існуючого методу визначення тональності інформаційного повідомлення. Заплановане створення програмного засобу, який може бути використаний як в державних установах так і на підприємствах.

9 Матеріали які подаються після закінчення НДР

По завершенню роботи подається пояснювальна записка та ілюстративна частина.

10 Порядок приймання НДР та її етапів

Апробація на науково-технічних конференціях та семінарах. Результати роботи будуть розглядатися на засіданні ДЕК із захисту магістерських кваліфікаційних робіт.

Попередній захист та доопрацювання МКР грудень 2021 р.

Представлення МКР до захисту 18 грудня 2021 р.

Захист МКР 23.12.21.

11 Вимоги до розроблення документації

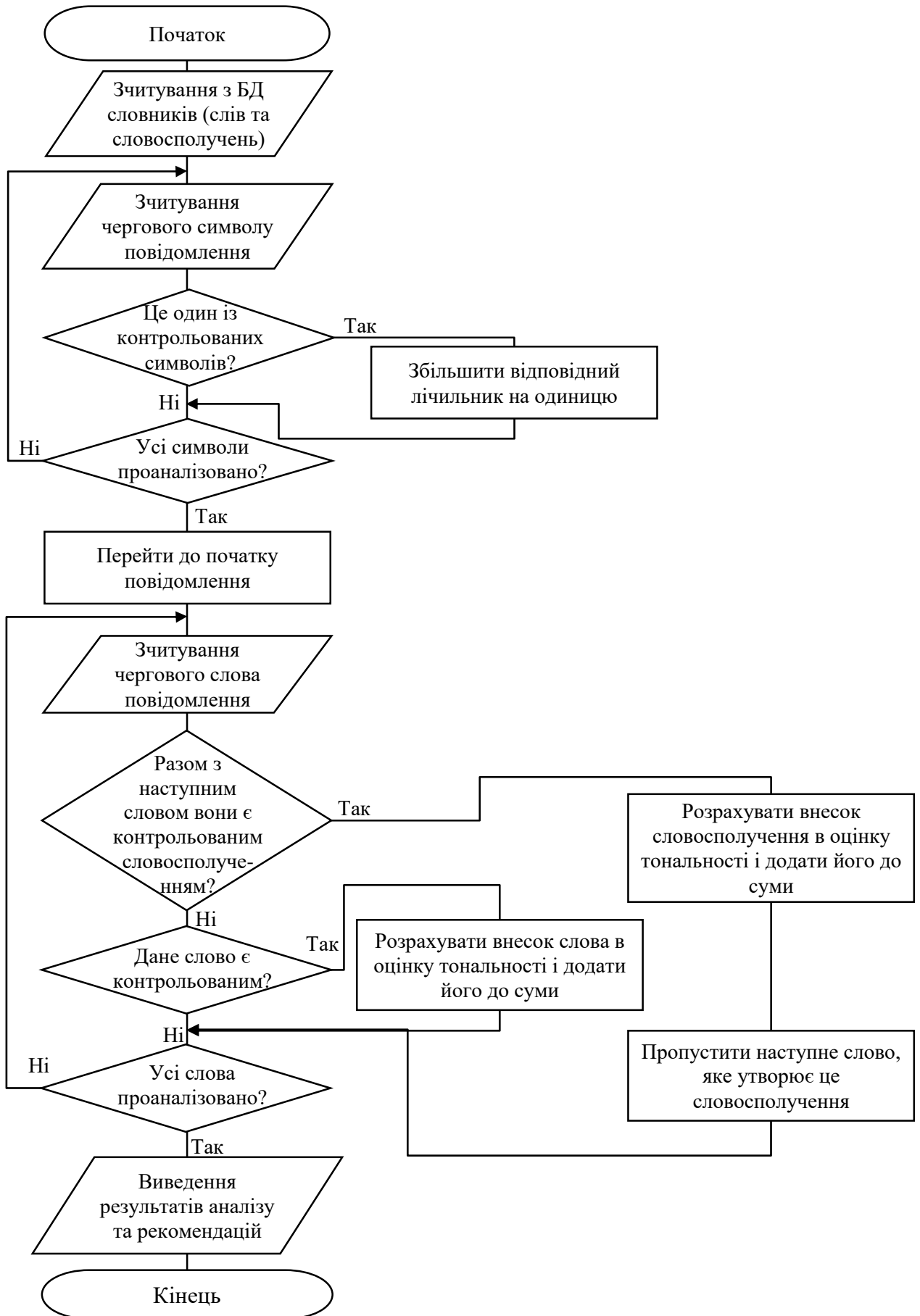
Документація буде виконуватись за допомогою комп'ютерного набору у відповідності вимог ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання».

12 Вимоги щодо технічного захисту інформації з обмеженим доступом

У зв'язку з тим, що дана робота не містить інформації, що потребує захисту у відповідності до законів України, заходи з її технічного захисту не передбачаються.

Додаток Б

Схема роботи програми



Додаток В

Хід встановлення ODBC

Завантаження цього програмного продукту виконується з офіційного сайту самого програмного продукту MySQL. Після завантаження переходимо до встановлення драйвера (рис. В.1).

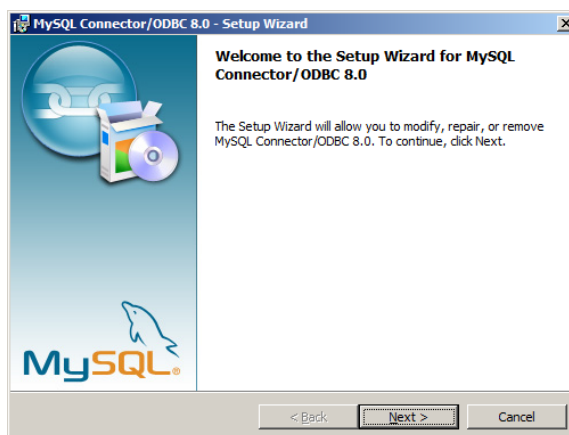


Рисунок В.1 – Установник драйвера ODBC для СУБД MySQL

Після встановлення в систему цей драйвер треба активувати засобами ОС Windows (далі розглядається на прикладі ОС Windows 7). Для цього необхідно перейти до Панелі управління (з меню «Пуск») та обрати у ній пункт ODBC, після чого має з'явитися вікно «Адміністратор источников данных ODBC» (рис. В.2). У ньому слід натиснути кнопку «Добавить...», та у новому вікні «Создание нового источника данных» обрати встановлений драйвер MySQL (рис. В.3).

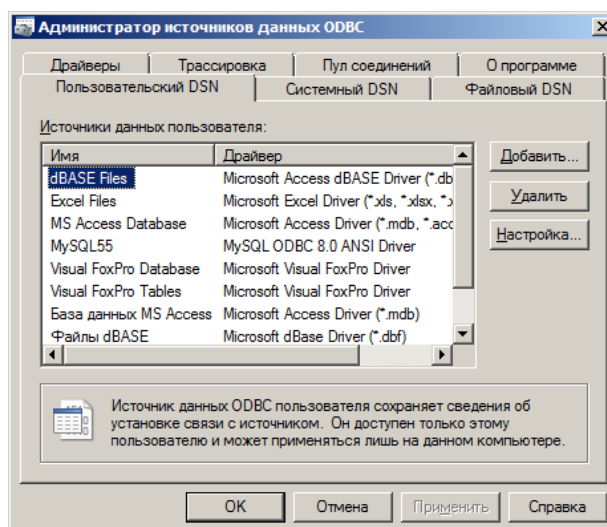


Рисунок В.2 – Вікно «Адміністратор источников данных ODBC»

Після цього у вікні «Администратор источников данных» з'явиться нове джерело даних із назвою на зразок «MySQL ODBC 8.0 ANSI Driver». Це джерело ще необхідно налаштувати, для чого виділяємо ім'я джерела даних MySQL55 та натискаємо кнопку «Настройка...», після чого відкривається нове діалогове вікно (рис. В.4).

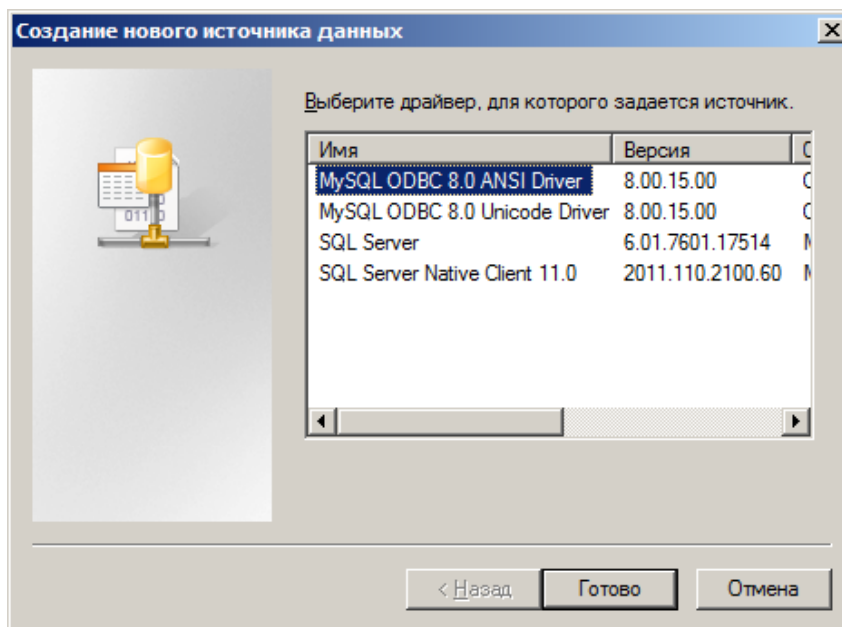


Рисунок В.3 – Вікно «Создание нового источника данных»

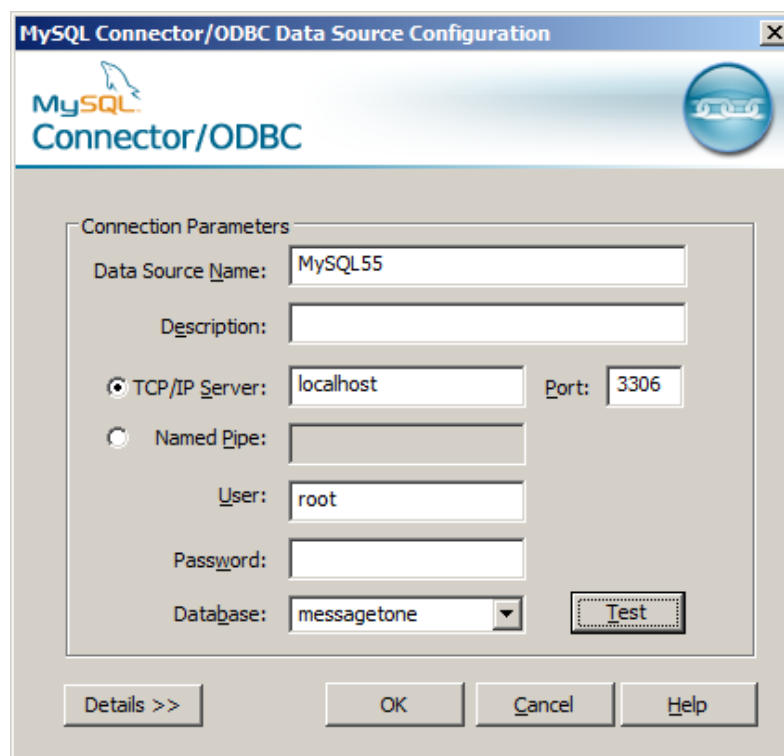


Рисунок В.4 – Вікно налаштувань сервера БД MySQL

У даному вікні слід ввести основні налаштування сервера БД:

– у якості TCP/IP Server слід задати localhost (оскільки використовується локальний сервер баз даних);

– користувач – root;

– поле пароль – залишити порожнім (або, за необхідності, задати деякий пароль);

– у полі Database можна вибрати базу даних messagetone і натиснути кнопку Test (це слід робити, якщо база даних з таким ім'ям уже була створена при розробці даної програми, а якщо СУБД встановлюється вперше, то цей крок слід відкласти і виконати тоді, коли база даних буде створена) – система має відповісти, що зв'язок встановлено (спливе повідомлення «Connection Successful»).

Додаток Г

Текст програми

```

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

namespace TextAnalyzer
{
    struct CWord
    {
        public string word;
        public int tone;
        public int weight;
    };
    struct CCombination
    {
        public string word1;
        public string word2;
        public int tone;
        public int weight;
    };
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void вихідToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        private void проПрограмуToolStripMenuItem_Click(object sender, EventArgs e)
        {
            FormAbout about = new FormAbout();
            about.ShowDialog();
        }
        private void відкритиToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if (openFileDialog1.ShowDialog() == DialogResult.Cancel)
                return;
            string filename = openFileDialog1.FileName;
            string fileText = System.IO.File.ReadAllText(filename);
            textBox1.Text = fileText;
        }
        private void button1_Click(object sender, EventArgs e)
        {
            textBox1.Clear();
            textBox2.Clear();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            textBox2.Text = "";
            var WordsList = new List<CWord>();
            var CombinationsList = new List<CCombination>();
            try
            {
                string MyConString = "Driver={MySQL ODBC 8.0 ANSI
Driver};Server=localhost;Database=messagetone;uid=root;pwd=";

```



```

OdbcConnection MyConnection = new OdbcConnection(MyConString);
MyConnection.Open();
string querystr = "";
querystr = @"SELECT WText,WTone,WWeight FROM words;";
OdbcCommand MyCommand = new OdbcCommand(@querystr, MyConnection);
OdbcDataReader MyDataReader;
MyDataReader = MyCommand.ExecuteReader();
if (MyDataReader.HasRows)
{
    int i = 0;
    CWord w1;
    while (MyDataReader.Read())
    {
        w1.word = MyDataReader.GetString(0).Trim();
        w1.tone = Convert.ToInt32(MyDataReader.GetString(1));
        w1.weight = Convert.ToInt32(MyDataReader.GetString(2));
        WordsList.Add(w1);
        i++;
    }
}
else
    MessageBox.Show(this, "Вибачте, сигнальні слова у базі даних не знайдені!
Зверніться, будь-ласка, до адміністратора системи");
MyDataReader.Close();
MyCommand.Dispose();

querystr = @"SELECT WCWord1,WCWord2,WCTone,WWeight FROM wordcombinations;";
OdbcCommand MyCommand2 = new OdbcCommand(@querystr, MyConnection);
MyDataReader = MyCommand2.ExecuteReader();
if (MyDataReader.HasRows)
{
    int i = 0;
    CCombination cw1;
    while (MyDataReader.Read())
    {
        cw1.word1 = MyDataReader.GetString(0).Trim();
        cw1.word2 = MyDataReader.GetString(1).Trim();
        cw1.tone = Convert.ToInt32(MyDataReader.GetString(2));
        cw1.weight = Convert.ToInt32(MyDataReader.GetString(3));
        CombinationsList.Add(cw1);
        i++;
    }
}
else
    MessageBox.Show(this, "Вибачте, сигнальні словосполучення у базі даних не
знайдені! Зверніться, будь-ласка, до адміністратора системи");
MyDataReader.Close();
MyCommand2.Dispose();

MyConnection.Close();
}
catch (OdbcException MyOdbcException)
{
    MessageBox.Show(this, "Вибачте, не можу підключитися до бази даних! Перевірте,
чи активний сервер баз даних");
    for (int i = 0; i < MyOdbcException.Errors.Count; i++)
    {
        MessageBox.Show(this, "ERROR #" + i + "\n" +
            "Message: " + MyOdbcException.Errors[i].Message + "\n" +
            "Native: " + MyOdbcException.Errors[i].NativeError.ToString() + "\n" +
            "Source: " + MyOdbcException.Errors[i].Source + "\n" +
            "SQL: " + MyOdbcException.Errors[i].SQLState + "\n");
    }
}
float numpl = 0, nump2 = 0, numexc = 0, numintr = 0, numsym = 0;
float totalres = 0;
int numfound = 0;
string str = textBox1.Text;
numsym = textBox1.Text.Length;
int prevsym=0;

```

```

foreach (char ch in textBox1.Text)
{
    numsym++;
    switch(ch)
    {
        case "(":
            if (prevsym==1)
            {
                nump1++;
                prevsym=2;
            }
            if (prevsym==2)
                break;
            prevsym=1;
            break;
        case ")":
            if (prevsym==11)
            {
                nump2++;
                prevsym=12;
            }
            if (prevsym==12)
                break;
            prevsym=11;
            break;
        case "!":
            if (prevsym==21)
            {
                numpexc++;
                prevsym=22;
            }
            if (prevsym==22)
                break;
            prevsym=21;
            break;
        case "?":
            if (prevsym==31)
            {
                numpintr++;
                prevsym=32;
            }
            if (prevsym==32)
                break;
            prevsym=31;
            break;
        default:
            prevsym=0;
    }
}
string word1, word2;
int combfound, numwords, j = 0;
numwords = str.Split().Length;
while (j < numwords)
{
    word1 = str.Split()[0];
    j++;
    word1 = word1.TrimStart(new char[] { ' ', '\'', '"', '(' });
    str = str.Substring(str.IndexOf(" ") + 1);
    if (checkBox2.Checked)
    {
        word2 = str.Split()[0];
        word2 = word2.TrimStart(new char[] { ' ', '\'', '"', '(' });
        combfound = 0;
        foreach (CCombination ccomb in CombinationsList)
        {

```



```

else {
    if (x >= -3 && x < -1)
    {
        MessageBox.Show(
            "Віднести джерело з якого надійшло повідомлення до підозрілих.
Тобто необхідно провести аналіз повідомлень, що надходили з даного " +
            "ресурсу та будуть надходити найближчим часом (наприклад на
протязі тижня). Якщо оцінка буде знаходитись в цьому ж діапазоні, " +
            "відповідальній особі необхідно прийняти рішення з приводу чи є
суттєво необхідним надходження інформації від даного джерела. " +
            "Відповідно, провести більш глибокий аналіз вхідних повідомлень за
певний проміжок часу:\r- 2 - 3 тижні, якщо частота надходження " +
            "інформації не дуже велика, знаходиться в діапазоні 1 - 3
повідомлення в день; \r - 1 - 2 тижні, якщо частота надходження інформації " +
            "не дуже велика, знаходиться в діапазоні 4 - 5 повідомлення в
день;\r - тиждень, якщо частота надходження інформації більша " +
            "ніж 6 повідомлень в день.\r При отриманні результатів оцінки з
такими самими значеннями та прийнятним змістом даних повідомлень, " +
            "зробити позначку, що дана оцінка є прийнятною для даного
ресурсу.Та зрозуміти, що немає потреби у постійному аналізі фахівцем " +
            "повідомлень з даного ресурсу з оцінкою, що знаходиться в даному
діапазоні. А у випадку зміни оцінки потрібно насторожитись " +
            "та проаналізувати чому оцінка була змінена.\rЯкщо оцінка постійно
змінюється, як у бік більш негативної оцінки(і це трапляється " +
            "доволі часто) так і у бік більш позитивної оцінки, з'ясувати з
чим пов'язані дані зміни.При необхідності, або при отриманні " +
            "небажаних результатів припинити обмін з даним джерелом.\rУ
випадку, якщо було виявлено, що дане повідомлення або не тільки " +
            "одне повідомлення, а й усі(більшість) повідомлення з даного
джерела містять деструктивний вплив зробити інформативну розсилку " +
            "серед співробітників, що контактували з даною інформацією. Дана
розсилка повинна роз'яснити співробітникам, що це були повідомлення, " +
            "які мали на меті дестабілізувати роботу системи шляхом впливу на
них(людський ресурс).А також розповсюдити у системі нейтралізуюче " +
            "повідомлення, для приведення функціонування системи до норми.",
"Рекомендації для оцінки тональності в проміжку від -3 до " +
            "-1", MessageBoxButtons.OK
        );
    }
    else
    {
        if (x >= -1 && x < 1)
        {
            MessageBox.Show(
                "Зазвичай таку оцінку можна віднести до категорії нейтральних,
але в залежності від змісту того чи іншого повідомлення. " +
                "Бажано, щоб фахівець (відповідальна особа) провів аналіз
вхідних повідомлень за певний проміжок часу, з даного джерела:\r" +
                "- 2 - 3 тижні, якщо частота надходження інформації не дуже
велика, знаходиться в діапазоні 1 - 2 повідомлення в день;\r" +
                "- 1 - 2 тижні, якщо частота надходження інформації не дуже
велика, знаходиться в діапазоні 3 - 4 повідомлення в день;\r" +
                "- тиждень, якщо частота надходження інформації більша ніж 5
повідомлень в день.\rЦе є необхідним для того, щоб зрозуміти " +
                "чи дана оцінка є характерною для інформації з даного джерела
чи ні(наприклад це може бути своєрідний стиль написання автора). " +
                "Якщо оцінка повідомлень перебуває в даному діапазоні або
деколи і у вишому, то дане джерело можна віднести до прийнятних. " +
                "Та реагувати тільки у випадку зміни оцінки у бік нижчого
діапазону.\rЯкщо оцінка постійно змінюється, у тому числі і в " +
                "негативну сторону, з'ясувати з чим пов'язані дані зміни.При
необхідності, повідомити адміністрацію ресурсу про ті чи інші " +
                "зміни, можливо це буде своєчасною реакцією на проникнення до
їхньої системи, якого вони не змогли виявити.", "Рекомендації " +
                "для оцінки тональності в проміжку від -1 до 1",
                MessageBoxButtons.OK
            );
        }
        else
        {if (x >= 1 && x < 3){
            MessageBox.Show(

```


Додаток Д

Протокол перевірки на наявність плагіату



Ім'я користувача:
Каплуш В.А. ЗІ

ID перевірки:
1009704808

Дата перевірки:
17.12.2021 12:11:01 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
17.12.2021 12:15:02 EET

ID користувача:
61408

Назва документа: Лиськов

Кількість сторінок: 76 Кількість слів: 15350 Кількість символів: 114951 Розмір файлу: 1.84 MB ID файлу: 1009703289

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

12.7%

Схожість

Найбільша схожість: 4.13% з Інтернет-джерелом (https://sida.kpi.ua/attachments/article/1442/MD_Za-.pdf)

7.59% Джерела з Інтернету

13

Сторінка 78

6.04% Джерела з Бібліотеки

14

Сторінка 78

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0.76%

Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 15 слів та 0%)

0.67% Вилучення з Інтернету

37

Сторінка 79

0.1% Вилученого тексту з Бібліотеки

49

Сторінка 80

Модифікації

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

Замінені символи

87

Підозріле форматування

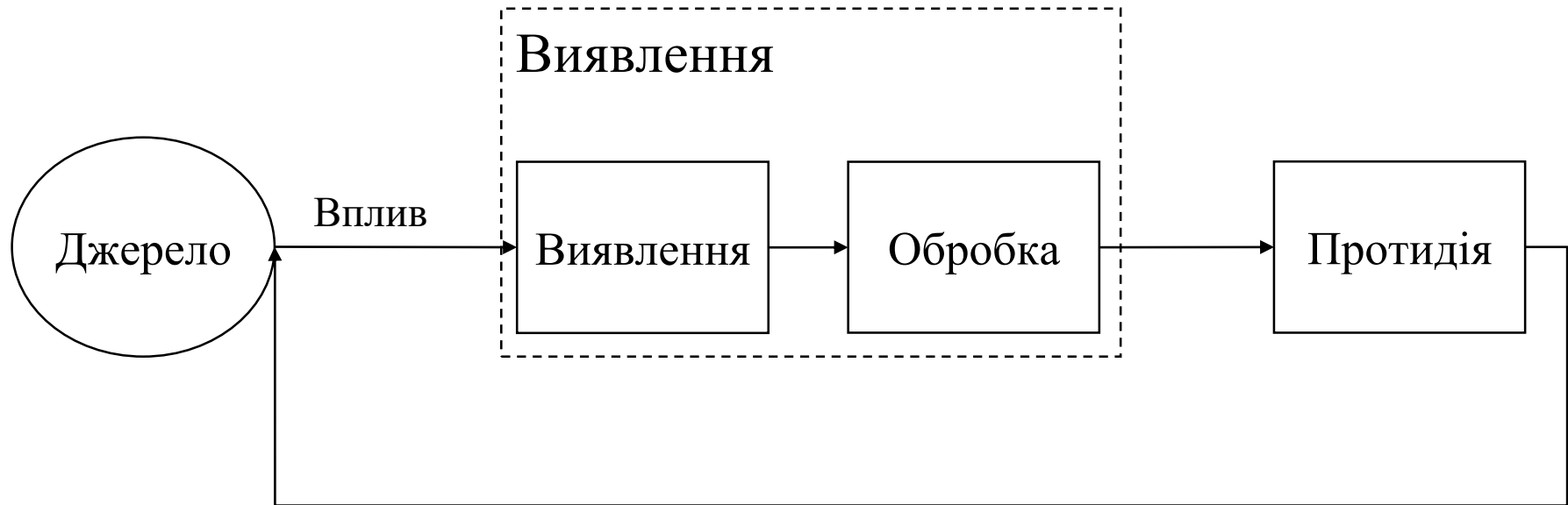
13
сторінок

ІЛЮСТРАТИВНА ЧАСТИНА

СИСТЕМА ОЦІНЮВАННЯ ТОНАЛЬНОСТІ ІНФОРМАЦІЙНОГО ПОВІДОМЛЕННЯ

(Назва магістерської кваліфікаційної роботи)

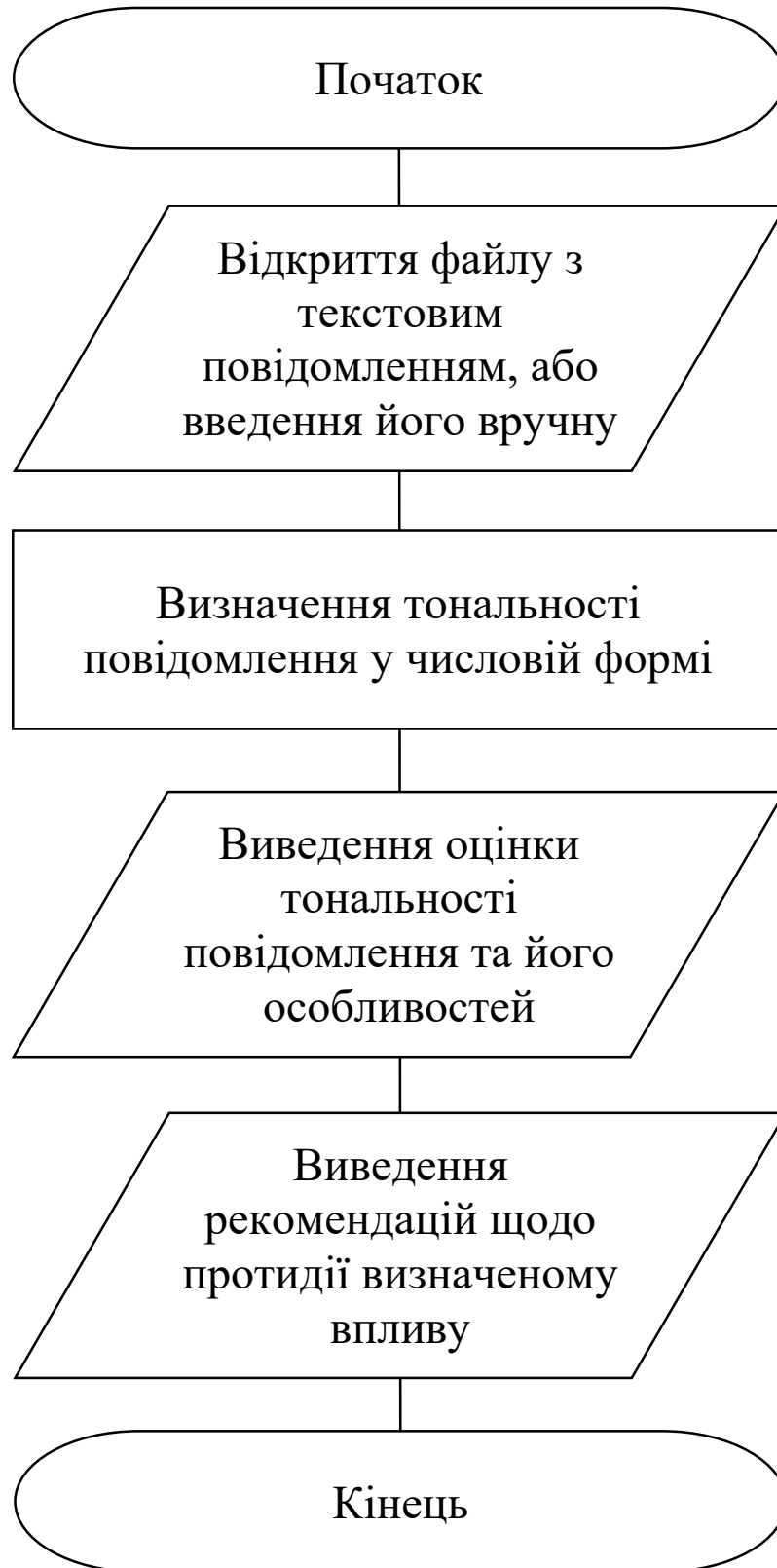
СТРУКТУРНА СХЕМА РОБОТИ СИСТЕМИ



08-20.МКР.007.00.000 ІЧ1

Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Лиськов Д. В.			Система оцінювання тональності інформаційного повідомлення. Структурна схема роботи системи	Літ.	Арк.	Аркуші
Перевір.		Дудат'єв А. В.					1	1
Реценз.		Азарова А. О.				ВНТУ, 1БС-20м		
Н. Контр.		Дудат'єв А. В.						
Затверд.		Лужецький В.А.						

СХЕМА АЛГОРИТМУ РОБОТИ КОРИСТУВАЧА ІЗ ПРОГРАМОЮ



08-20.МКР.007.00.000 142

Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Лиськов Д. В.			Система оцінювання тональності інформаційного повідомлення. Схема алгоритму роботи користувача із програмою	Літ.	Арк.	Аркуші
Перевір.		Дудатьєв А. В.					1	1
Реценз.		Азарова А. О.				ВНТУ, 1БС-20м		
Н. Контр.		Дудатьєв А. В.						
Затверд.		Лужецький В.А.						

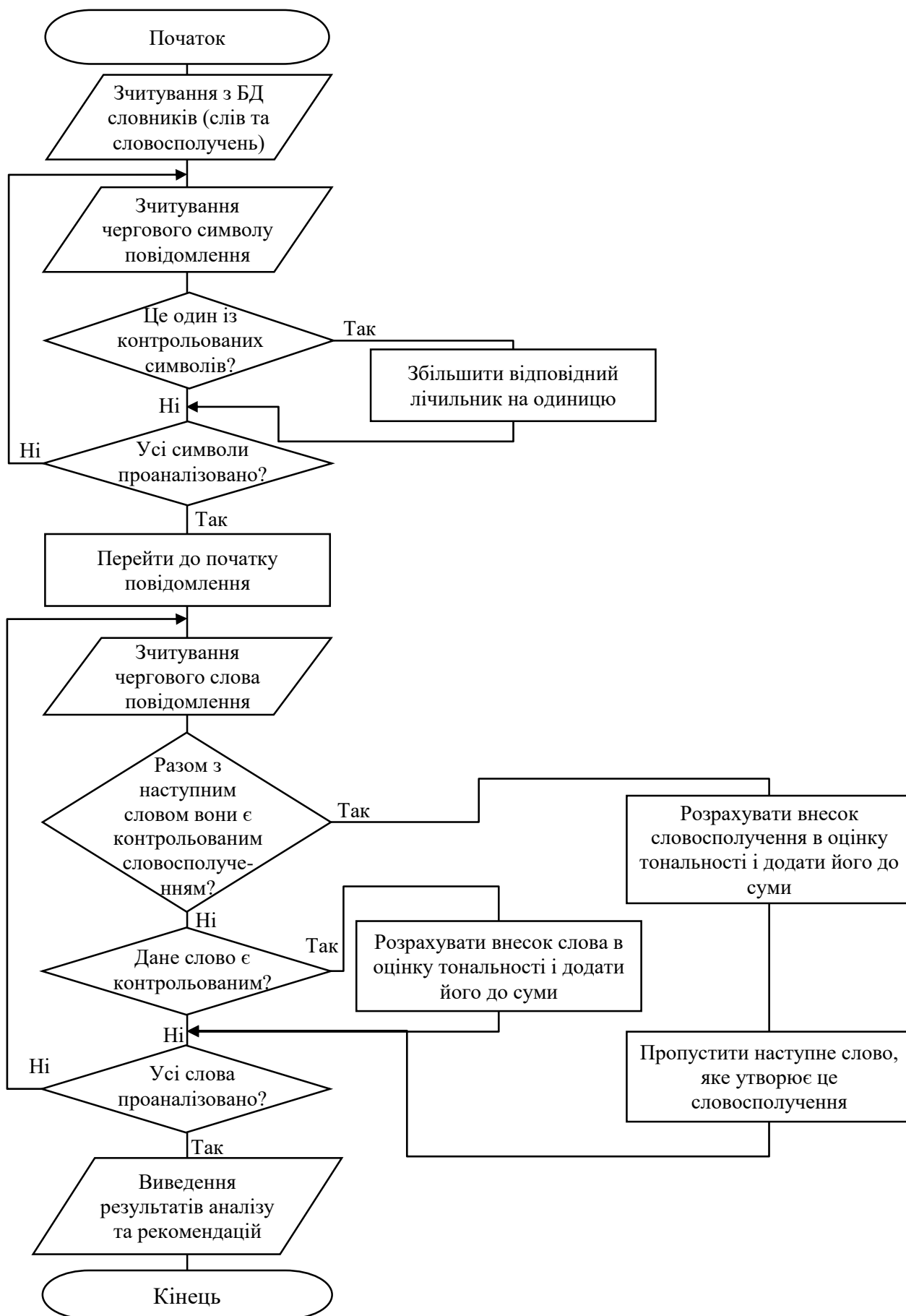
СТРУКТУРНА СХЕМА РОБОТИ ПРОГРАМИ



08-20.МКР.007.00.000 ІЧЗ

Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Лиськов Д. В.			Система оцінювання тональності інформаційного повідомлення. Структурна схема роботи програми	Літ.	Арк.	Аркуші
Перевір.		Дудат'єв А. В.					1	1
Реценз.		Азарова А. О.				ВНТУ, 1БС-20м		
Н. Контр.		Дудат'єв А. В.						
Затверд.		Лужецький В.А.						

СХЕМА РОБОТИ ПРОГРАМИ



08-20.МКР.007.00.000 ІЧ4

Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Лиськов Д. В.			Система оцінювання тональності інформаційного повідомлення. Функціональна схема роботи програми	Літ.	Арк.	Аркуші
Перевір.		Дудат'єв А. В.					1	1
Реценз.		Азарова А. О.				ВНТУ, 1БС-20м		
Н. Контр.		Дудат'єв А. В.						
Затверд.		Лужецький В.А.						

СТРУКТУРА БАЗИ ДАНИХ

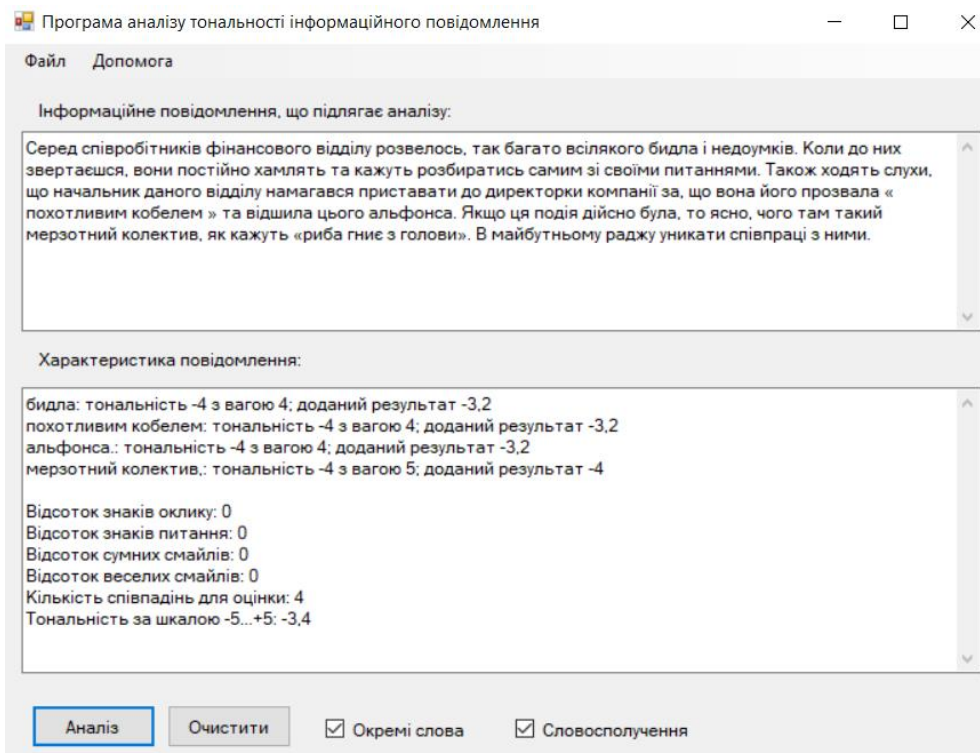


08-20.МКР.007.00.000 145

Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Лиськов Д. В.			Літ.	Арк.	Аркушів
Перевір.		Дудатьєв А. В.				1	1
Реценз.		Азарова А. О.			ВНТУ, 1БС-20м		
Н. Контр.		Дудатьєв А. В.					
Затверд.		Лужецький В.А.					

Система оцінювання тональності
інформаційного повідомлення.
Структура бази даних

ФРАГМЕНТИ ПРОВЕДЕННЯ ТЕСТУВАННЯ



Програма аналізу тональності інформаційного повідомлення

Файл Допомога

Інформаційне повідомлення, що підлягає аналізу:

Серед співробітників фінансового відділу розвелось, так багато всілякого бидла і недоумків. Коли до них звертаєшся, вони постійно хамлять та кажуть розбиратись самим зі своїми питаннями. Також ходять слухи, що начальник даного відділу намагався приставати до директорки компанії за, що вона його прозвала « похотливим кобелем » та відшила цього альфонса. Якщо ця подія дійсно була, то ясно, чого там такий мерзотний колектив, як кажуть «риба гние з голови». В майбутньому раджу уникати співпраці з ними.

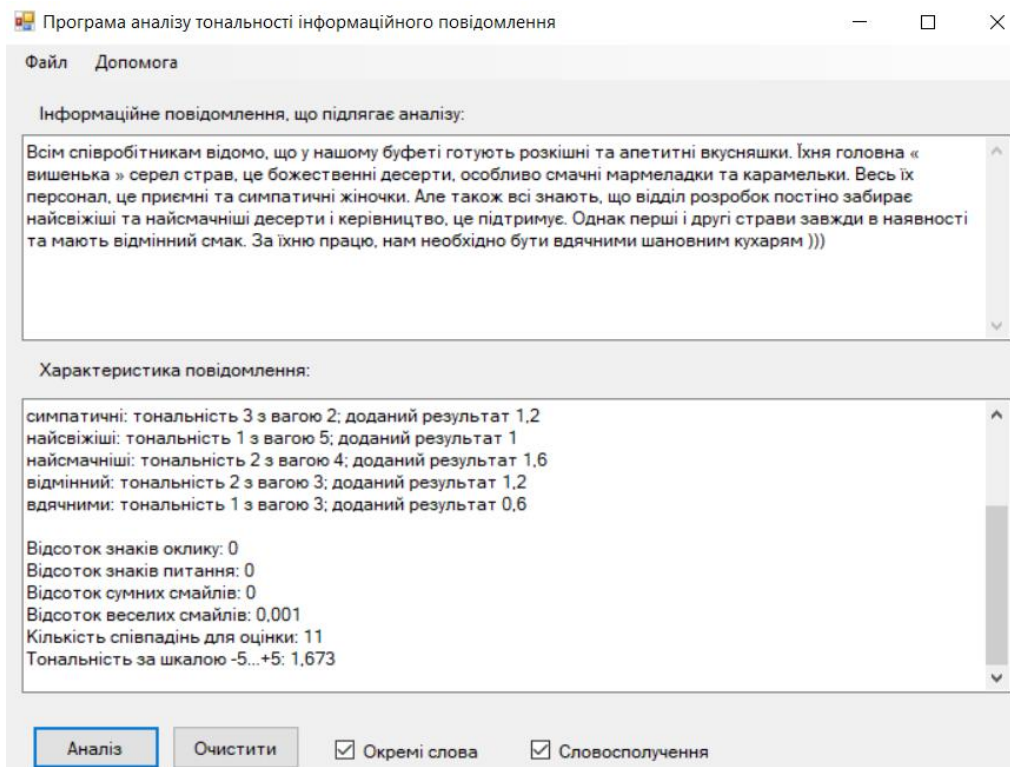
Характеристика повідомлення:

бидла: тональність -4 з вагою 4; доданий результат -3,2
похотливим кобелем: тональність -4 з вагою 4; доданий результат -3,2
альфонса.: тональність -4 з вагою 4; доданий результат -3,2
мерзотний колектив.: тональність -4 з вагою 5; доданий результат -4

Відсоток знаків оклику: 0
Відсоток знаків питання: 0
Відсоток сумних смайлів: 0
Відсоток веселих смайлів: 0
Кількість співпадінь для оцінки: 4
Тональність за шкалою -5...+5: -3,4

 Окремі слова Словосполучення

Результат тестування повідомлення «Текст №1»



Програма аналізу тональності інформаційного повідомлення

Файл Допомога

Інформаційне повідомлення, що підлягає аналізу:

Всім співробітникам відомо, що у нашому буфеті готують розкішні та апетитні вкунсяшки. Їхня головна « вишенька » серел страв, це божественні десерти, особливо смачні мармеладки та карамельки. Весь їх персонал, це приємні та симпатичні жіночки. Але також всі знають, що відділ розробок постійно забирає найсвіжіші та найсмачніші десерти і керівництво, це підтримує. Однак перші і другі страви завжди в наявності та мають відмінний смак. За їхню працю, нам необхідно бути вдячними шановним кухарям)))

Характеристика повідомлення:

симпатичні: тональність 3 з вагою 2; доданий результат 1,2
найсвіжіші: тональність 1 з вагою 5; доданий результат 1
найсмачніші: тональність 2 з вагою 4; доданий результат 1,6
відмінний: тональність 2 з вагою 3; доданий результат 1,2
вдячними: тональність 1 з вагою 3; доданий результат 0,6

Відсоток знаків оклику: 0
Відсоток знаків питання: 0
Відсоток сумних смайлів: 0
Відсоток веселих смайлів: 0,001
Кількість співпадінь для оцінки: 11
Тональність за шкалою -5...+5: 1,673

 Окремі слова Словосполучення

Результат тестування повідомлення «Текст №4»

08-20.МКР.007.00.000 146

Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Лиськов Д. В.			Система оцінювання тональності інформаційного повідомлення. Фрагменти проведення тестування	Літ.	Арк.	Аркуші
Перевір.		Дудатьєв А. В.					1	2
Реценз.		Азарова А. О.				ВНТУ, 1БС-20м		
Н. Контр.		Дудатьєв А. В.						
Затверд.		Лужецький В.А.						

ФРАГМЕНТИ ПРОВЕДЕННЯ ТЕСТУВАННЯ

Рекомендації для оцінки тональності в проміжку від -5 до -3



Тимчасове блокування взаємодії з джерелом. У випадку, якщо це повідомлення надійшло з веб-ресурсу або у групі соц. мережі, або у месенджері повідомити адміністрацію про його негативний характер. Якщо це лист на поштової скриньці тимчасово заблокувати відправника. Відповідальній особі провести аналіз повідомлення і прийняти рішення про те, що робити у конкретно цьому випадку надалі. Можливі набори заходів повинні зазначатись у політиці безпеки установи.

Можна рекомендувати виконання наступних дій:

- максимально виключити дане повідомлення із функціонування системи(для того щоб користувачі, які ще не ознайомились з даною інформацією, з нею не контактували взагалі);
- повідомити користувачів, які взаємодіяли з даною інформацією, що це був сплавнований вплив на те, щоб нанести шкоду функціонуванню системи, в особливості нанести максимальну шкоду самим співробітникам(як правило психічну);
- після усунення даного впливу, сформувані повідомлення для робітників, що були піддані даному впливу, що його нейтралізує та відновить функціонування людського ресурсу у системі;
- відповідно до цієї «атаки» сформувані звіт, та використати його для подальшого формування інструктажів для усього колективу організації.

Якщо надалі з даного джерела будуть надходити повідомлення з оцінкою цього діапазону, виконати ті ж кроки для кожного випадку та при можливості необхідно заблокувати на постійній основі зв'язок з цим ресурсом.

OK

Рекомендації для повідомлення «Текст №1»

Рекомендації для оцінки тональності в проміжку від 1 до 3



Така оцінка є прийнятною. Головне, щоб оцінка повідомлень джерела постійно знаходилась в своєму діапазоні, у такому випадку ніяких дій застосовувати до даного каналу віщання. Якщо оцінка даного джерела постійно змінюється необхідно провести змістовний аналіз даних повідомлень та з'ясувати з чим це пов'язано. У випадку впливу на цю зміну декількох несуттєвих фраз, то можна віднести дане джерело до декількох прийнятних діапазонів. Якщо зміни діапазону спостерігаються дуже рідко, то необхідно проводити аналіз саме тих повідомлень, які виходять за межі встановленого для даного джерела діапазону значень оцінки тональності.

OK

Рекомендації для повідомлення «Текст №4»

					08-20.МКР.007.00.000 146	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		