МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій, обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Левківський Денис Володимирович
(прізвище, ім'я, по батькові здобувача освіти)

УДК 0004.9:004.93

КВАЛІФІКАЦІЙНА РОБОТА

Інформаційна система аналізу тональності тексту
(тема роботи)
122 "Комп'ютерні науки"
(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____
(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи
Веретюк Сергій Михайлович
(прізвище, ім'я, по батькові)
кандитат технічних наук., доцент
(науковий ступінь, вчене звання)

Житомир – 2026

Висновок кафедри ком’пютерних технологій і моделювання систем

за результатами попереднього захисту: допущений

Протокол засідання кафедри ком’пютерних технологій і моделювання систем

№ 20 від « 05 » червня 2026 р.

Завідувач кафедри ком’пютерних технологій і моделювання систем

к. пед. н.. доцент _____ КОВАЛЬЧУК.М.О
(науковий ступінь, вчене звання) (підпис) (прізвище, ім’я, по батькові)
« 05 » червня 2026 р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____захистив
(прізвище ,ім’я, по батькові)

кваліфікаційну роботу з оцінкою: _____

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК
Лаборант кафедри _____ КОРОЛЬЧУК.В.В
(науковий ступінь, вчене звання) (підпис) (прізвище, ім’я, по батькові)

АНОТАЦІЯ

Левківський Д. В. Інформаційна система аналізу тональності тексту.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за спеціальністю 122 «Комп'ютерні науки». – Поліський національний університет, Житомир, 2026.

Кваліфікаційна робота присвячена дослідженню, архітектурному проєктуванню та програмній реалізації інформаційної системи аналізу тональності тексту. У роботі здійснено аналіз предметної області, який виявив потребу в розробленні локального програмного забезпечення для опрацювання україномовного та англомовного контенту.

Спроєктована система TextSelekt поєднує методи машинного навчання та лексиконний підхід. Розроблено 16-фазний конвеєр обробки мови. Реалізовано функціонал обробки документів у форматах TXT, DOCX та PDF. Програмний комплекс побудовано з використанням Python, FastAPI та React, із збереженням даних у SQLite. Створено механізми кешування (LRU) і дедуплікації (SHA-256) запитів. Розроблено інтерфейс із динамічним підсвічуванням тексту та автоматичною генерацією PDF-звітів.

Отримані результати можуть бути застосовані в медіа-моніторингу, маркетингових дослідженнях та соціологічних опитуваннях.

Ключові слова: Аналіз тональності, Інформаційна система, Лексиконний аналіз, Машинне навчання, Обробка природної мови, Фастапі, Реакт.

SUMMARY

Levkivskyi D. V. Information system for text sentiment analysis.

Qualification work for a bachelor's degree in specialty 122 "Computer Science". – Polissia National University, Zhytomyr, 2026.

The qualification work is devoted to the research, architectural design, and software implementation of an information system for text sentiment analysis. The work analyzes the subject area, identifying the need to develop local software for processing Ukrainian and English content.

The designed TextSelekt system combines machine learning methods and a lexicon approach. A 16-phase natural language processing pipeline was developed. Functionality for processing documents in TXT, DOCX, and PDF formats was implemented. The software complex is built using Python, FastAPI, and React, with data stored in SQLite. Request caching (LRU) and deduplication (SHA-256) mechanisms were created. An interface with dynamic text highlighting and automatic PDF report generation was developed.

The obtained results can be applied in media monitoring, marketing research, and sociological surveys.

Keywords: Sentiment analysis, Information system, Lexicon analysis, Machine learning, Natural language processing, Fastapi, React**.**

ЗМІСТ

ВСТУП

Актуальність теми дослідження. У сучасному глобалізованому інформаційному суспільстві обсяги текстових даних, що безперервно генеруються користувачами соціальних мереж, новинних порталів, медіа-платформ та корпоративних інформаційних ресурсів, зростають за експоненціальним законом. Аналіз емоційного забарвлення цих масивів текстів стає критично важливим і невід'ємним завданням для комерційного бізнесу, державних структур, правоохоронних органів та соціологічних інституцій. Більшість існуючих на ринку комерційних систем оптимізовано переважно для англомовного контенту, тоді як якісне опрацювання української мови ускладнюється її надзвичайно багатою морфологією, словозміною та загальною синтаксичною гнучкістю. Водночас у сучасному корпоративному та державному секторах гостро постає необхідність локальних (on-premise) рішень для аналізу тональності в умовах жорстких вимог до конфіденційності даних та захисту від витоку інформації через хмарні API. Зважаючи на ці фактори, розроблення інформаційної системи, що комплексно поєднує лексиконні та машинні методи для аналізу тональності з глибокою адаптацією до україномовного середовища та можливістю автономного розгортання, є надзвичайно актуальним інженерним завданням.

Мета і завдання роботи. Метою кваліфікаційної роботи є теоретичне дослідження, архітектурне проєктування та безпосередня програмна реалізація інформаційної системи аналізу тональності тексту, яка здатна ефективно обробляти текстові масиви та електронні документи локально. Для досягнення мети було визначено наступні завдання: провести аналіз методів обробки природної мови; формалізувати вимоги до інформаційної системи; спроєктувати клієнт-серверну архітектуру та модель бази даних; реалізувати багаторівневий алгоритм класифікації тональності; розробити застосунок із функціями кешування та генерації звітів; провести тестування розробленого програмного продукту.

Предмет та об'єкт дослідження. Об'єктом дослідження виступають процеси автоматизованого визначення емоційного забарвлення, класифікації та загальної тональності масивів текстової інформації. Предметом дослідження є математичні

методи, алгоритми класифікації, лінгвістичні лексикони та програмні засоби обробки природної мови, необхідні для створення функціональної інформаційної системи.

Методи дослідження. Для ефективного вирішення поставлених у роботі завдань було застосовано широкий комплекс спеціалізованих інженерних методів. До основних методів дослідження належать методи обробки природної мови (NLP), методи векторизації тексту (TF-IDF, Word Embeddings) та статистичний аналіз точності моделей класифікації. Системний аналіз використано для формування архітектури інформаційної системи. Методи інфологічного та даталогічного моделювання застосовано для проєктування нормалізованої реляційної бази даних. Об'єктно-орієнтоване програмування та асинхронні моделі обчислень лягли в основу програмної реалізації.

Практичне значення отриманих результатів. Практична цінність роботи полягає у створенні повністю завершеного програмного продукту інформаційної системи TextSelekt, готової до впровадження у бізнес-процеси організацій. Розроблена система забезпечує унікальну можливість глибокого аналізу сирих текстів та електронних документів (TXT, DOCX, PDF) із гарантією збереження конфіденційності. Алгоритмічні рішення, закладені в основу лінгвістичного конвеєра, можуть бути успішно використані як самостійні програмні модулі в інших NLP-проєктах.

Структура та обсяг роботи. Кваліфікаційна робота складається з анотації, змісту, вступу, трьох розділів основної частини, загальних висновків, списку використаних джерел та додатків.

За темою кваліфікаційної роботи опубліковано наукові публікації, а саме:

Левківський Д. В. Моделі та методи аналізу тональності тексту в інформаційних системах обробки природної мови. Інформаційні технології та моделювання систем : зб. праць учасн. Всеукр. наук.-практ. конф. здобувачів вищої освіти і молодих вчених, м. Житомир, 22 трав. 2026. Житомир : Поліський національний університет, 2026. С. 21-22.

Левківський Д. В. Інформаційні технології автоматизованої класифікації тексту за емоційним забарвленням. Modern Science: Research, Economy and Innovation : Collection of Scientific Papers with Proceedings of the 5th International Scientific and Practical Conference, Zagreb, Croatia, June 17–19, 2026. Zagreb : International Scientific Unity, 2026. С. 208–209.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд сучасних методів та підходів до аналізу тональності тексту

Аналіз тональності тексту, який у міжнародній науковій літературі також відомий як sentiment analysis або аналіз думок, є одним із найбільш важливих та динамічних напрямів у сучасній галузі обробки природної мови (NLP). Основна мета цього високотехнологічного процесу полягає у точному визначенні емоційного забарвлення, яке автор свідомо чи несвідомо вклав у свій текст, та подальшій класифікації цього тексту як позитивного, негативного або нейтрального [1]. Сучасні академічні дослідження чітко виділяють кілька фундаментальних підходів до вирішення цієї складної лінгвістичної задачі, серед яких традиційно домінують лексиконні методи, алгоритми класичного машинного навчання та інноваційні моделі глибокого навчання [2]. Лексиконні методи історично базуються на використанні попередньо укладених словників, де кожному окремому слову або стійкій фразі присвоюється певна числова вага або емоційна категорія. Цей класичний підхід відрізняється високою швидкістю виконання обчислень та абсолютною прозорістю результатів, оскільки першопричину присвоєння певної тональності завжди можна відслідкувати за знайденими у словнику лексемами [3].

Для ефективного подолання обмежень лексиконних методів у сучасних системах активно застосовуються методи машинного навчання з учителем, які передбачають ітеративне тренування математичних моделей на величезних масивах попередньо розмічених даних. Такі алгоритми, як метод опорних векторів (SVM), наївний байєсівський класифікатор та класична логістична регресія, здатні виявляти приховані закономірності у неструктурованому тексті та формувати значно точніші передбачення. У процесі стандартизованої підготовки даних для цих алгоритмів надзвичайно широко використовуються методи векторизації тексту, зокрема статистична міра TF-IDF (Term Frequency-Inverse Document Frequency) та щільні векторні представлення слів Word Embeddings (наприклад, Word2Vec або GloVe) [4]. Застосування цих методів дозволяє перетворити семантичну текстову інформацію на багатовимірні числові матриці, придатні для

комп'ютерної обробки. Для об'єктивної оцінки якості роботи таких алгоритмів обов'язково проводиться статистичний аналіз точності моделей класифікації з використанням стандартизованих метрик Accuracy, Precision, Recall та F1-Score [5].

Останніми роками спостерігається стрімкий та безперервний розвиток методів глибокого навчання, зокрема застосування великих мовних моделей (LLM), які побудовані на основі інноваційної архітектури Transformer (BERT, RoBERTa та їх багатомовні варіації). Ці революційні моделі повністю змінили галузь NLP завдяки впровадженню механізму самоуваги (self-attention mechanism), який дозволяє нейронній мережі одночасно враховувати двоспрямований контекст кожного окремого слова у складному реченні [6]. Трансформери стабільно демонструють найвищі показники точності на стандартних тестових наборах даних, проте їх практичне використання вимагає наявності значних обчислювальних ресурсів, що суттєво ускладнює розгортання таких систем у середовищах з обмеженими апаратними потужностями [7]. Тому в сучасних комерційних системах розробники найчастіше віддають перевагу гібридним архітектурним підходам, які гармонійно комбінують високу прозорість лексиконних методів із точністю машинного навчання.

Надзвичайно значний концептуальний вплив на розвиток комп'ютеризованого емоційного аналізу мала поява розширених психолінгвістичних баз даних, найвідомішою з яких є глобальний словник NRC Emotion Lexicon. Цей унікальний лексикон дозволяє класифікувати аналізовані слова не лише за бінарною шкалою позитив-негатив, але й об'єктивно відносити їх до восьми базових людських емоцій: радості, смутку, страху, гніву, здивування, очікування, довіри та огиди [8]. Використання таких багатовимірних словників відкриває безпрецедентні можливості для проведення глибокого семантичного аналізу тексту. Водночас ефективна інтеграція алгоритмів стемінгу, лематизації та морфологічного розширення дозволяє якісно адаптувати ці англомовні словники до інших мов із багатою словозміною, що є критично важливим етапом обробки для українського сегменту, де словозміна повністю змінює синтаксичну структуру речення [19].

1.2 Аналіз існуючих інформаційних систем та аналогів

Сучасний ринок спеціалізованого програмного забезпечення пропонує надзвичайно широкий спектр технічних рішень для автоматизованого аналізу тональності тексту, які суттєво варіюються від відкритих академічних бібліотек до потужних корпоративних хмарних сервісів. Серед комерційних програмних продуктів безумовні лідируючі позиції займають хмарні платформи від глобальних технологічних корпорацій, зокрема сервіси Amazon Web Services (AWS Comprehend), Google Cloud Natural Language API та Microsoft Azure Cognitive Services . Ці високобюджетні системи забезпечують найвищий рівень автоматизації процесів та пропонують легкість інтеграції через стандартизовані протоколи REST API. Незалежні дослідження переконливо показують, що показники загальної точності (Accuracy) для цих корпоративних систем на англомовних текстах коливаються в досить вузьких межах від 67% до 72% (таблиця 1.1).

Таблиця 1.1 – Порівняльний аналіз точності хмарних систем аналізу тональності (за даними досліджень англомовних корпусів)

Назва інформаційної системи	**Загальна точність (Accuracy)**	**Ефективність розпізнавання позитиву**	**Ефективність розпізнавання нейтралітету**
AWS Comprehend	71,8%	Висока точність (High Precision)	Середня точність
Google Cloud NLP	70,3%	Висока точність (High Precision)	Середня точність
IBM Watson NLU	67,1%	Середня точність	Висока точність (висока Precision, низький Recall)
Microsoft Azure	69,5%	Середня точність	Середня точність

Незважаючи на вражаючу обчислювальну потужність глобальних хмарних сервісів, їх практичне використання неминуче супроводжується низкою суттєвих недоліків. У сучасному корпоративному та державному секторах гостро постає необхідність локальних (on-premise) рішень для аналізу тональності в умовах жорстких вимог до конфіденційності даних та захисту від витоку інформації через хмарні API [11]. Передача чутливих документів, персональних даних клієнтів або

внутрішньої корпоративної документації на сторонні віддалені сервери прямо порушує стандарти інформаційної безпеки (зокрема GDPR). Крім того, якість семантичного аналізу україномовних текстів у цих глобальних системах часто залишається на неприйнятно низькому рівні, оскільки підтримка локальних мов зазвичай реалізується через попередній автоматичний машинний переклад, що призводить до втрати емоційних відтінків.

Серед програмних рішень із повністю відкритим вихідним кодом надзвичайно широкої популярності набув інструмент VADER (Valence Aware Dictionary and sEntiment Reasoner), який фундаментально базується на правилах та спеціалізованих емпіричних словниках. Цей оптимізований алгоритм демонструє дуже високу швидкість обробки на звичайних процесорах та забезпечує деталізований числовий результат із точним розрахунком пропорцій позитиву, негативу та нейтральності [12]. Проте внутрішня лексична архітектура VADER жорстко оптимізована виключно під граматику англійської мови, і його пряма адаптація для української мови неможлива через складну морфологію. Більш сучасним підходом є використання попередньо натренованих мовних моделей, таких як XLM-RoBERTa, які демонструють значно вищі показники точності (таблиця 1.2), проте вимагають дорогих графічних прискорювачів (GPU) та функціонують як "чорні скриньки", не дозволяючи підсвітити конкретні емоційні слова [13].

Таблиця 1.2 – Порівняння метрик ефективності відкритих моделей аналізу тональності

Модель аналізу	**Підхід**	**Точність (Accuracy)**	**F1-Score**	**Оцінка ROC-AUC**
VADER	Лексиконний (Rule-based)	88,6%	0,83	0,85
BERT	Трансформер (Deep Learning)	93,1%	0,89	0,91
Flair	Нейромережевий (Embeddings)	95,3%	0,92	0,94
RoBERTa	Оптимізований трансформер	94,8%	0,91	0,93

1.3 Визначення інформаційних потреб та функціональних вимог до системи TextSelekt

На основі всебічного аналізу предметної області та виявлених критичних недоліків існуючих комерційних рішень було сформовано концепцію нової інтелектуальної інформаційної системи TextSelekt. Головним функціональним призначенням розроблюваної системи є надання кінцевим користувачам зручного інструменту для комплексного аналізу тональності текстів та електронних документів англійською та українською мовами. Згідно з формалізованими інформаційними потребами, система повинна стабільно забезпечувати обробку масивів розміром до 60 000 символів в межах одного запиту. Кінцевий користувач повинен гарантовано мати можливість бачити детальну візуальну статистику щодо конкретних слів, фраз та їхнього математичного впливу на фінальний розрахований результат.

Функціональні вимоги до архітектури системи TextSelekt чітко структуровано за логічними блоками. Перший блок безпосередньо стосується управління користувачами та забезпечення інформаційної безпеки. Розроблювана система повинна повноцінно підтримувати процеси реєстрації, безпечну авторизацію та ефективне керування сесіями на основі технології JWT-токенів [14]. Другий фундаментальний блок визначає вимоги до ядра підсистеми аналізу: програма повинна містити вбудовану функцію автовизначення мови, розраховувати базову тональність та формувати спеціалізовані метадані для

реалізації механізму динамічного підсвічування тексту на клієнтській стороні (визначення індексів start/end кожного значущого токена). Третій блок описує процеси автоматичного вилучення чистого текстового контенту з документів TXT, DOCX та PDF із підтримкою функції часткового аналізу фрагментів.

Нефункціональні вимоги до розроблюваної системи TextSelekt охоплюють надзвичайно важливі питання загальної продуктивності, горизонтальної масштабованості та стійкості до пікових навантажень. Для максимальної мінімізації часу відгуку сервера передбачено обов'язкове використання надійної дворівневої системи кешування. Перший рівень базується на імплементації алгоритму In-memory TTL LRU, а другий стратегічний рівень полягає у тотальній дедуплікації запитів через безперервне обчислення SHA-256 хешу нормалізованого тексту . Система також повинна містити вбудовані механізми динамічного обмеження частоти запитів (Rate Limiting) для надійного захисту серверної інфраструктури від потенційних зловмисних атак типу відмова в обслуговуванні.

Висновки до розділу 1

У першому розділі кваліфікаційної роботи було проведено всебічний аналіз предметної області та детально досліджено сучасні алгоритмічні підходи до аналізу тональності тексту, включаючи методи векторизації (TF-IDF, Word Embeddings) та статистичний аналіз точності моделей. Встановлено, що для гарантованого забезпечення високої точності найбільш ефективним є застосування складних гібридних методів, які об'єднують лексиконний підхід та машинне навчання. Розглянуто існуючі аналоги та виявлено їхні критичні недоліки щодо опрацювання україномовного контенту. Окремо аргументовано критичну необхідність створення локальних (on-premise) рішень для захисту корпоративних та персональних даних від витоку через публічні хмарні API.

На основі результатів комплексного аналізу було успішно сформовано детальні функціональні та нефункціональні вимоги до нової інформаційної системи TextSelekt. Чітко визначено необхідність розроблення захищених підсистем авторизації, модулів екстракції тексту з документів (TXT, DOCX, PDF) та багаторівневого конвеєра лінгвістичного аналізу. Сформовані вимоги стали

фундаментальною теоретичною основою для подальшого системного проєктування архітектури та розроблення логічних моделей реляційної бази даних.

РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Архітектура інформаційної системи та взаємодія компонентів

Проєктування базової архітектури інформаційної системи TextSelekt було здійснено на основі сучасної клієнт-серверної моделі з надзвичайно чітким та жорстким розділенням зон відповідальності між програмними компонентами. Такий передовий підхід гарантовано забезпечує високу горизонтальну масштабованість, суттєво полегшує процес автоматизованого тестування програмного забезпечення та дозволяє незалежно оновлювати різні модулі системи без простоїв . Загальна інфраструктура проєкту концептуально складається з двох основних, слабко зв'язаних частин: клієнтського застосунку (Frontend), повністю відповідального за взаємодію з користувачем та рендеринг, та серверного ядра (Backend), яке надійно інкапсулює всю складну бізнес-логіку обробки природної мови та безпечного збереження даних. Безперервний зв'язок між фронтендом та бекендом здійснюється через суворо стандартизований програмний інтерфейс REST API з використанням легковагового формату обміну даними JSON .

Серверна частина розроблюваної системи реалізована на базі популярної мови програмування Python із використанням високопродуктивного, сучасного асинхронного вебфреймворку FastAPI. Вибір саме фреймворку FastAPI обґрунтовано його винятковою здатністю ефективно обробляти велику кількість паралельних запитів завдяки асинхронній природі (asyncio), вбудованою інтеграцією з інструментами суворої типізації Pydantic та повністю автоматичною генерацією технічної документації за світовим стандартом OpenAPI . Серверний застосунок має складну багаторівневу модульну структуру, що детально наведена у таблиці 2.1. Такий логічний розподіл гарантує неухильне дотримання принципів чистої архітектури (Clean Architecture) та значно полегшує подальше технічне супроводження та рефакторинг програмного коду .

Таблиця 2.1 – Структурні шари серверної частини інформаційної системи

Назва архітектурного шару	**Опис та головне функціональне призначення**
API Layer (Маршрутизація)	Обробка вхідних HTTP-запитів, сувора валідація даних через моделі Pydantic, диспетчеризація запитів до відповідних сервісів.
Business Logic (Сервіси)	Реалізація алгоритмів аналізу тональності, екстракції тексту з файлів, генерації PDF-звітів та створення криптографічних JWT-токенів.
Data Layer (Репозиторії)	Взаємодія з базою даних через потужну технологію об'єктно-реляційного відображення (ORM) SQLAlchemy.
Configuration / Core	Централізоване управління CORS-політиками, змінними оточення, налаштуваннями безпеки та лімітуванням трафіку (SlowAPI).

Клієнтська частина системи професійно побудована з використанням популярної бібліотеки React, яка забезпечує швидке створення динамічного, реактивного інтерфейсу на основі повністю ізольованих багаторазових компонентів. Збірка всього проєкту здійснюється за допомогою сучасного інструменту Vite, що гарантує надзвичайно високу швидкість компіляції під час розробки та глибоку оптимізацію ресурсів для фінального продакшн-середовища . Управління складним станом застосунку реалізовано за допомогою вбудованого механізму React Context, який відповідає за глобальне збереження автентифікаційних даних та поточних результатів сеансу аналізу. Для безпечної взаємодії із серверним API розроблено виділений сервісний модуль client.js, який містить оптимізовані функції-обгортки для виконання мережевих HTTP-запитів та повністю автоматичного додавання авторизаційних заголовків до кожного пакету .

Процес алгоритмічної взаємодії компонентів під час виконання завдання аналізу тексту виглядає наступним, суворо визначеним чином. Клієнт формує структурований POST-запит до цільового ендпоінту /api/v1/analyze, обережно передаючи в тілі запиту очищений текст та вибрані параметри налаштувань. Шар маршрутизації на бекенді успішно приймає запит, миттєво перевіряє права доступу користувача через криптографічний підпис JWT-токена та передає текст до основного сервісного шару . Сервіс здійснює швидке автовизначення мови тексту,

після чого математично обчислює унікальний SHA-256 хеш. Якщо результат для цього специфічного хешу вже присутній у швидкому In-memory кеші або базі даних, система миттєво повертає клієнту готову відповідь, уникаючи зайвих обчислень. У протилежному випадку активується повний алгоритм лінгвістичного аналізу, результати якого паралельно зберігаються в базі даних та надсилаються клієнту у вигляді складного структурованого JSON-об'єкта для візуалізації в браузері .

2.2 Алгоритмічне забезпечення процесу аналізу тональності

Функціональна логіка системи TextSelekt базується на багаторівневому алгоритмі аналізу тональності тексту (версія v8), розробленому для врахування лінгвістичних особливостей цільових мов. Процес обробки вхідних даних формалізовано у вигляді конвеєра (pipeline), що складається з 16 послідовних фаз.

На першій, підготовчій фазі здійснюється пошук фраз за принципом найдовшого збігу (greedy longest-first). Система сканує нормалізований текст і зіставляє його зі словником скомпільованих фразеологізмів, надаючи пріоритет складним словесним конструкціям порівняно з окремими словами.

Друга фаза реалізує механізм морфологічного розширення (Morph phrase fallback) для обробки української мови. За відсутності точного збігу в базовому словнику застосовуються морфологічні індекси для розпізнавання різних граматичних форм слова (відмінків, часів, родів).

Третя фаза передбачає зіставлення знайдених слів із динамічними словниковими категоріями для попередньої класифікації лексем за їхньою семантичною спрямованістю.

На четвертій фазі виконується розрахунок негативних ознак (Negative evidence scoring) із застосуванням множників категорій та нарахуванням балів за присутність підсилювальних слів у локальному контексті токена. Загальний бал для негативного контексту S_neg для множини токенів документа $T = \{t_1, t_2, ..., t_n\}$ обчислюється за формулою:

$$S_{neg} = \sum (w(t_i) \cdot \mu(t_i) + \beta(t_i))$$

де $w(t_i)$ — базова емоційна вага токена з лексикону, $\mu(t_i)$ — категорійний множник, а $\beta(ti)$ — динамічний бонус підсилення у локальному вікні токена.

П'ята фаза інтегрує базу даних емоційної інтенсивності NRC для української та англійської мов. На цьому етапі розраховуються емоційні ваги для восьми базових категорій емоцій, що дозволяє уточнити показники позитивного та негативного контексту.

Шоста фаза забезпечує формування оцінок підтримки, захоплення, інформативності та невпевненості (Positive/neutral scoring) для побудови емоційного профілю документа.

На сьомій фазі приймається рішення щодо базової тональності тексту (Base sentiment decision) шляхом порівняння зважених сум балів із порогами чутливості. Класифікація базової тональності тексту Class(T) визначається за допомогою шматково-заданої функції:

$$\text{Class(T)} = \begin{cases} Positive, \text{якщо } S_{pos} - S_{neg} \geq \theta_{pos} \\ Negative, \text{якщо } S_{neg} - S_{pos} \geq \theta_{neg} \\ Neutral, \text{в інших випадках} \end{cases}$$

де θ_{pos} та θ_{neg} — встановлені пороги чутливості для позитивного та негативного класів відповідно.

Восьма та дев'ята фази відповідають за корекцію попередніх результатів і калібрування впевненості моделі. Застосування евристичних правил (Neutral correction) дозволяє зменшити ймовірність хибнопозитивних класифікацій у текстах із високою концентрацією нейтральних або технічних термінів. Одночасно обчислюється метрика впевненості (confidence) та встановлюється прапорець низького рівня сигналу (low_signal) за відсутності достатньої кількості емоційно забарвлених слів у тексті. З урахуванням правил корекції нейтральності, метрика скоригованої впевненості розраховується за формулою:

$$C_{adj} = \frac{\max(S_{pos}, S_{neg})}{S_{pos} + S_{neg} + \varepsilon} * (1 - \lambda \frac{N_{neutral}}{N_{total}})$$

де C_{adj} — скоригована впевненість, ε — згладжувальний коефіцієнт для запобігання діленню на нуль, λ — коефіцієнт штрафу, $N_{neutral}$ — кількість нейтральних термінів, а N_{total} — загальна кількість токенів у аналізованому тексті.

На десятій фазі визначається підклас тональності (Subclass determination) для деталізації відтінків емоційного забарвлення.

Фази з 11 по 13 охоплюють розрахунок статистики класифікованих слів, підрахунок несемантичних елементів (стоп-слів, філерів) і ранжування ключових фраз для виведення у звітах.

Фази з 14 по 16 призначені для семантичного аналізу з використанням методів машинного навчання. Чотирнадцята фаза (Transformer gating) є опціональною і активується за низького показника впевненості (confidence) базового лексиконного аналізу. У цьому випадку залучається локальна трансформерна модель для уточнення результату. П'ятнадцята фаза забезпечує об'єднання прогнозів кількох моделей (Multi-task model blending), а шістнадцята фаза генерує статистику частин мови (POS tagging stats) за допомогою інструменту UDPipe.

2.3 Інфологічне моделювання та структура бази даних

Для гарантованого забезпечення надійного, цілісного збереження даних системи TextSelekt було професійно розроблено нормалізовану реляційну модель бази даних, яка фізично реалізована засобами легкої СКБД SQLite та управляється через потужну ORM-систему SQLAlchemy. Процес системного інфологічного моделювання розпочався з чіткого визначення базових сутностей предметної області, виявлення їхніх ключових атрибутів та формалізації логічних зв'язків. Головними архітектурними сутностями розробленої інформаційної системи є безпосередньо користувач (User) та згенеровані результати аналізу (Analysis). Внутрішню структуру бази даних було глибоко оптимізовано для забезпечення надзвичайно швидкого пошуку інформації та підтримки 100% цілісності даних при виконанні паралельних транзакцій .

Сутність "User" концептуально призначена для безпечного зберігання облікових записів та чутливих авторизаційних даних клієнтів. До її базових

атрибутів відносяться унікальний глобальний ідентифікатор у форматі UUID, електронна пошта (яка виступає єдиним логіном) та криптографічний хеш пароля, надійно зашифрований за допомогою стандартизованого алгоритму bcrypt. Для текстового поля електронної пошти створено унікальний системний індекс, що фізично унеможливлює реєстрацію кількох акаунтів на одну й ту саму адресу та суттєво прискорює процес авторизації . Також ця таблиця містить системну мітку точного часу створення запису (created_at) у міжнародному форматі UTC для аудиту.

Сутність "Analysis" зберігає абсолютно всі деталі оброблених текстових запитів та логічно пов'язана із сутністю "User" зв'язком типу "один до багатьох" через зовнішній ключ user_id. Характеристики основних атрибутів фізичної таблиці analyses наведено у таблиці 2.2. Застосування гнучкого типу даних JSON для збереження масивів ключових слів та складної розмітки підсвічування тексту (highlights) дозволяє мінімізувати загальну кількість зв'язаних реляційних таблиць та кардинально оптимізувати операції читання при генерації історії запитів на клієнті .

Таблиця 2.2 – Структура атрибутів таблиці Analyses в базі даних

Назва атрибута	**Тип даних (SQLAlchemy)**	**Призначення атрибута в системі**
id	String (UUID)	Первинний ключ, глобально унікальний ідентифікатор запису аналізу.
text	Text	Оригінальний вхідний текстовий масив, безпосередньо наданий користувачем.
base_sentiment	String	Базова тональність тексту (positive, negative, neutral).
confidence_base	Float	Числовий коефіцієнт впевненості математичної моделі від 0.0 до 1.0.
hash	String	SHA-256 хеш нормалізованого тексту для забезпечення процесу дедуплікації.
highlights	JSON	Масив складних об'єктів із параметрами start, end, category для рендерингу.
save_to_history	Boolean	Прапорець, що системно вказує на необхідність відображення запису в історії.

Для забезпечення стабільно високої продуктивності системи розроблено надзвичайно ефективну стратегію багаторівневого кешування та запобігання дублюванню обчислень. При кожному новому HTTP-запиті система динамічно генерує комбінований текстовий рядок формату {мова}|{флаг_smart_context}|{нормалізований_текст} та миттєво обчислює його SHA-256 хеш . У фізичній базі даних створено спеціальний оптимізований індекс ix_analyses_hash для миттєвого алгоритмічного пошуку записів за цим ідентифікатором. Якщо такий запис вже знайдено, повторний ресурсомісткий розрахунок негайно скасовується, що суттєво економить обчислювальні ресурси центрального процесора. Окрім повільної дискової бази даних, додатково реалізовано кеш у швидкій оперативній пам'яті (TTLCache), який базується на перевіреному алгоритмі Least Recently Used (LRU) і додатково знижує загальне навантаження на підсистему вводу/виводу сервера при надзвичайно частих однотипних запитах .

Висновки до розділу 2

У другому розділі кваліфікаційної роботи було детально спроєктовано загальну архітектуру інформаційної системи TextSelekt та розроблено її ключові інтелектуальні алгоритми обробки інформації. Науково обґрунтовано вибір клієнт-серверної моделі розгортання на базі інструментів FastAPI та React, що гарантовано забезпечує гнучкість, горизонтальну масштабованість та суворе дотримання принципів розділення відповідальності між компонентами. Розроблена архітектура дозволяє надзвичайно легко інтегрувати нові програмні модулі та безперешкодно оновлювати сервіси машинного навчання без критичного переривання роботи основної системи.

Надзвичайно детально описано унікальне алгоритмічне забезпечення процесу аналізу тональності, яке безперебійно функціонує за оптимізованим шістнадцятифазним конвеєром. Цей потужний алгоритм інноваційно поєднує класичні методи лексиконного аналізу, глибокого морфологічного розширення, інтеграцію масивних емоційних баз даних NRC та сучасні моделі машинного навчання, гарантуючи безпрецедентно високу точність обробки українських та

англійських текстів. Розроблено гнучку інфологічну модель та нормалізовану структуру бази даних у легкому середовищі SQLite, яка забезпечує надійне довгострокове збереження історії аналізів, глибоко оптимізована для швидкого пошуку через індексування хешів та повноцінно підтримує механізми дедуплікації для максимально ефективного використання обчислювальних ресурсів сервера

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА КЕРІВНИЦТВО КОРИСТУВАЧА

3.1 Опис інтерфейсу розробленої інформаційної системи

Практична реалізація візуальної частини здійснена у вигляді інтерактивного односторінкового застосунку (SPA) з використанням Tailwind CSS для забезпечення адаптивного дизайну. Основним вузлом є панель аналізу (`AnalyzePage.jsx`). Форма введення містить перемикач режимів роботи (текст або файл) та клієнтську валідацію на обмеження кількості символів до 60 000.

Головна сторінка застосунку надає користувачу зручну форму для введення тексту або завантаження файлів. На рисунку 3.1 зображено інтерфейс введення тексту для аналізу з налаштуваннями збереження в історію та увімкнення точнішого контекстного аналізу.

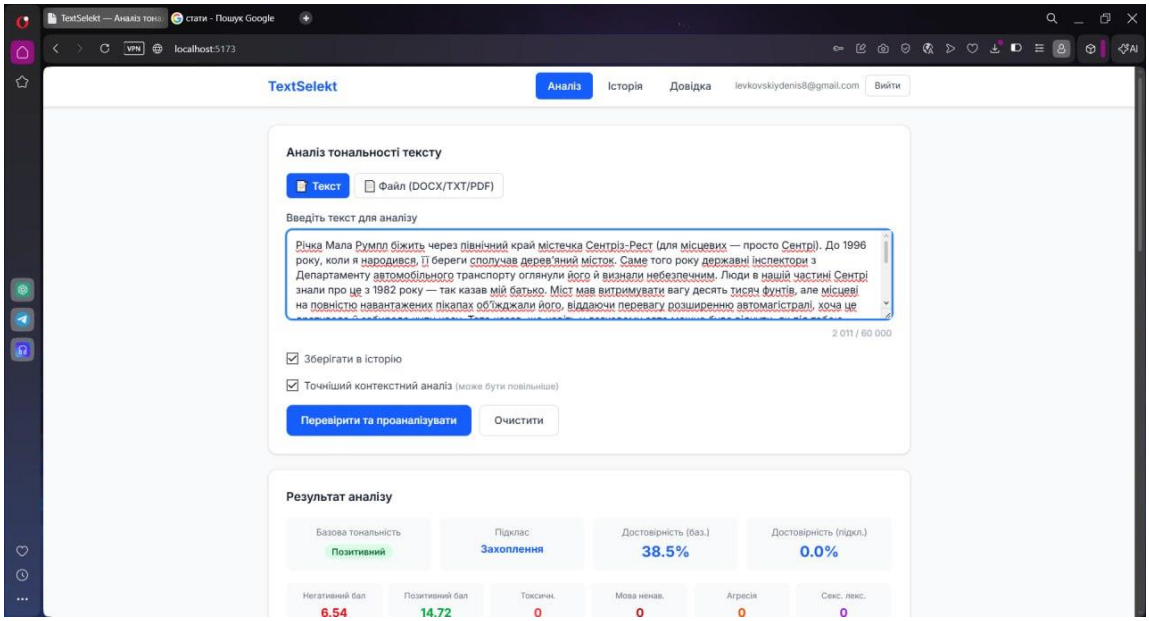


Рисунок 3.1 – Інтерфейс введення тексту для аналізу

У разі необхідності обробки цілих документів, користувач може перемкнутися на вкладку завантаження файлів. На рисунку 3.2 зображено інтерфейс завантаження електронних документів формату DOCX, TXT або PDF. Система підтримує опцію `extract_only` та можливість фрагментарного аналізу тексту через параметри `sliceStart` та `sliceEnd`.

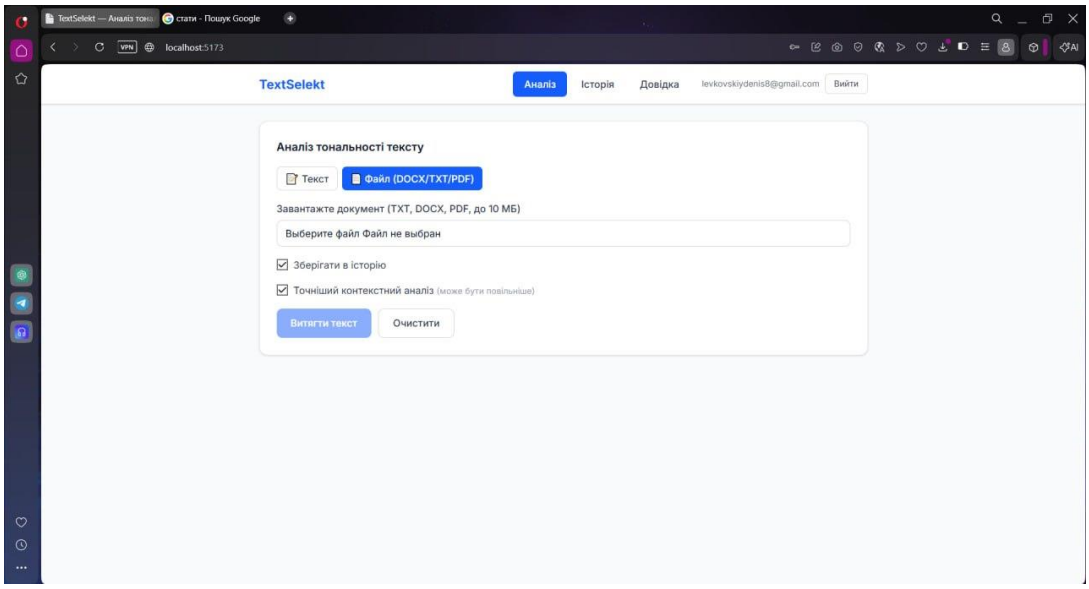


Рисунок 3.2 – Вкладка завантаження електронних документів

Після виконання серверного аналізу система повертає деталізований результат, який відмальовується у спеціальному компоненті `HighlightedText.jsx`. Цей компонент будує підсвічування на основі неперекривних діапазонів (spans) згідно з категоріями: phrase-first пріоритет, базова тональність, словники `sentiment_ua`, несемантичні елементи та POS-теги. На рисунку 3.3 зображено сторінку з результатами аналізу, де виведено базову тональність, підклас, рівень достовірності, а також реалізовано динамічне підсвічування тональних категорій у тексті.

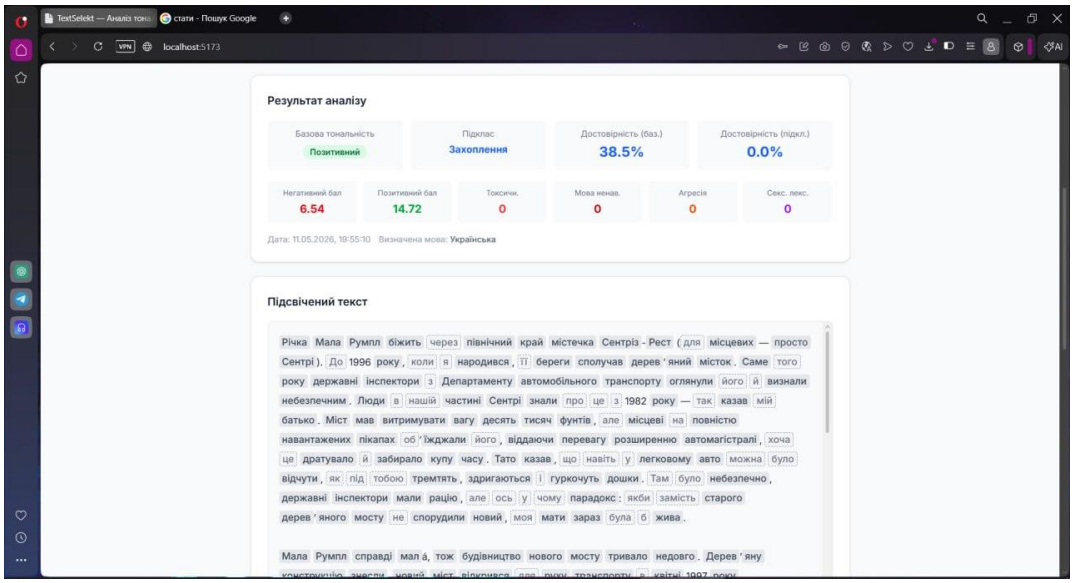


Рисунок 3.3 – Візуалізація результатів аналізу з динамічним підсвічуванням тональностей

Для забезпечення глибшого розуміння структури тексту застосунок формує графічну статистику. На рисунку 3.4 зображено статистику слів, типи збігів та кругову діаграму розподілу токенів, побудовану за допомогою графічної бібліотеки Chart.js.

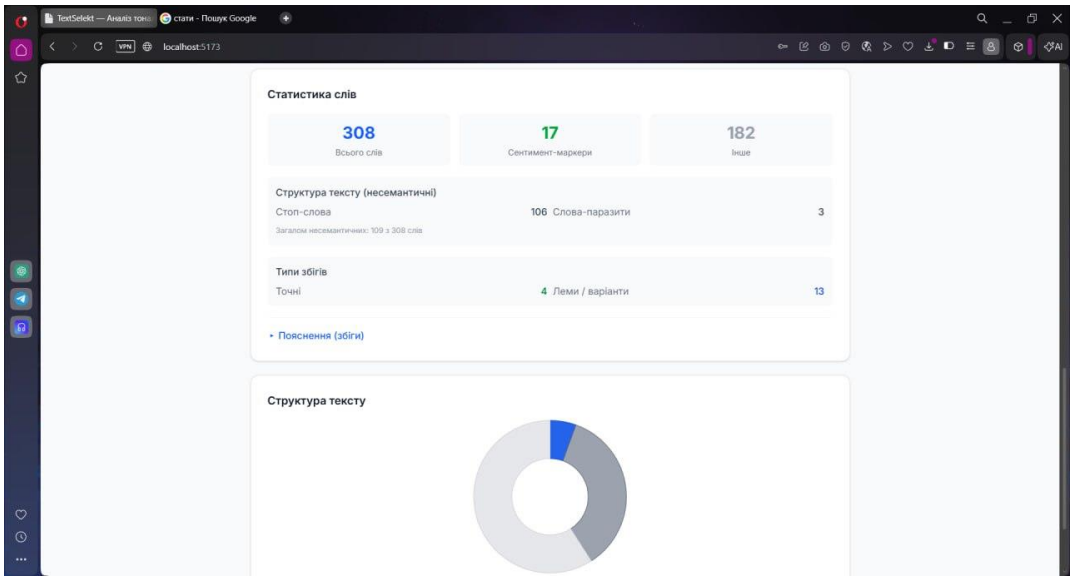


Рисунок 3.4 – Графічне відображення статистичних даних та структури тексту

3.2 Порядок встановлення, налаштування та розгортання системи

Процес інженерного розгортання інформаційної системи TextSelekt спроєктовано з глибоким урахуванням можливості автономного функціонування в повністю ізольованих (offline) серверних середовищах, що стовідсотково відповідає найсуворішим сучасним стандартам корпоративної кібербезпеки. Перед початком процесу встановлення необхідно обов'язково забезпечити наявність налаштованого цільового серверного середовища з інстальованим інтерпретатором Python версії не нижче 3.10 та менеджером середовищ Node.js версії 18 або вище . Внутрішня структура проєкту концептуально передбачає роздільне налаштування бекенду та фронтенду, з подальшим об'єднанням їхньої безпечної взаємодії через конфігураційні файли змінних оточення (ENV-файли).

Встановлення серверної частини логічно розпочинається зі створення ізольованого віртуального середовища Python та пакетного встановлення всіх необхідних залежностей із конфігураційного файлу requirements.txt. До ключових встановлюваних бібліотек належать фреймворк FastAPI, ORM SQLAlchemy,

Alembic, бібліотеки екстракції python-docx та pdfplumber, а також генератор PDF-файлів ReportLab. Наступним обов'язковим етапом є конфігурація бази даних: за допомогою консольного інструменту міграцій Alembic виконується повністю автоматичне створення необхідних реляційних таблиць у файлі бази SQLite . Технологічною особливістю розгортання цієї системи є необхідність одноразового виконання спеціалізованих скриптів з теки backend/scripts, які повністю відповідають за завантаження та бінарну компіляцію мовних ресурсів. Скрипт sync_resources.py завантажує віддалені статичні датасети та мовні моделі UDPipe, а скрипт compile_resources.py компілює величезні текстові лексикони у швидкий двійковий формат для забезпечення максимальної швидкості доступу під час обробки трафіку.

Розгортання складних модулів машинного навчання категорично вимагає одноразового виконання скрипту train_models.py, який здійснює локальне тренування багатозадачних нейромережевих моделей на раніше завантажених корпусах текстів безпосередньо на сервері клієнта. Завершальним етапом налаштування бекенду є ручна конфігурація змінних оточення у прихованому файлі .env, де адміністратору необхідно вказати унікальні секретні ключі для криптографічної генерації JWT-токенів (JWT_SECRET_KEY), параметри часу життя цих токенів та жорсткі ліміти обмеження швидкості обробки вхідних запитів . Після успішного завершення налаштування сервер запускається за допомогою продуктивного ASGI-сервера Uvicorn консольною командою uvicorn app.main:app --host 0.0.0.0 --port 8000.

Налаштування клієнтської частини тривіально зводиться до встановлення пакетів JavaScript-залежностей через стандартний пакетний менеджер npm командою npm install. У конфігураційному файлі фронтенду необхідно обов'язково вказати базову URL-адресу API сервера для коректної маршрутизації мережевих запитів . Для локального запуску застосунку в режимі розробки використовується команда npm run dev, яка миттєво активує збирач Vite. Для правильної підготовки системи до експлуатації у виробничому середовищі (Production) виконується команда npm run build, що автоматично створює глибоко оптимізовані статичні

HTML/JS/CSS файли, які в подальшому можуть безпроблемно обслуговуватися будь-яким класичним вебсервером, наприклад, високопродуктивним Nginx або Apache.

3.3 Керівництво користувача та результати тестування

Користувач після реєстрації та авторизації потрапляє на аналітичну панель. Він може вставити текст або завантажити документ, обрати опцію "Smart Context" (яка підключає Transformer-уточнення при низькій впевненості) та ініціювати аналіз.

Усі виконані аналізи зберігаються в базі даних та доступні для подальшого перегляду в компоненті `HistoryPage.jsx`. На рисунку 3.5 зображено журнал збережених аналізів користувача (сторінка історії) з можливістю повнотекстового пошуку, фільтрації за мовою, датою та функціями повторного аналізу або генерації PDF-звітів. Формування PDF-файлів відбувається "на льоту" модулем `pdf_report_service.py` на базі ReportLab, куди інтегруються метадані, діаграми та фрагменти тексту.

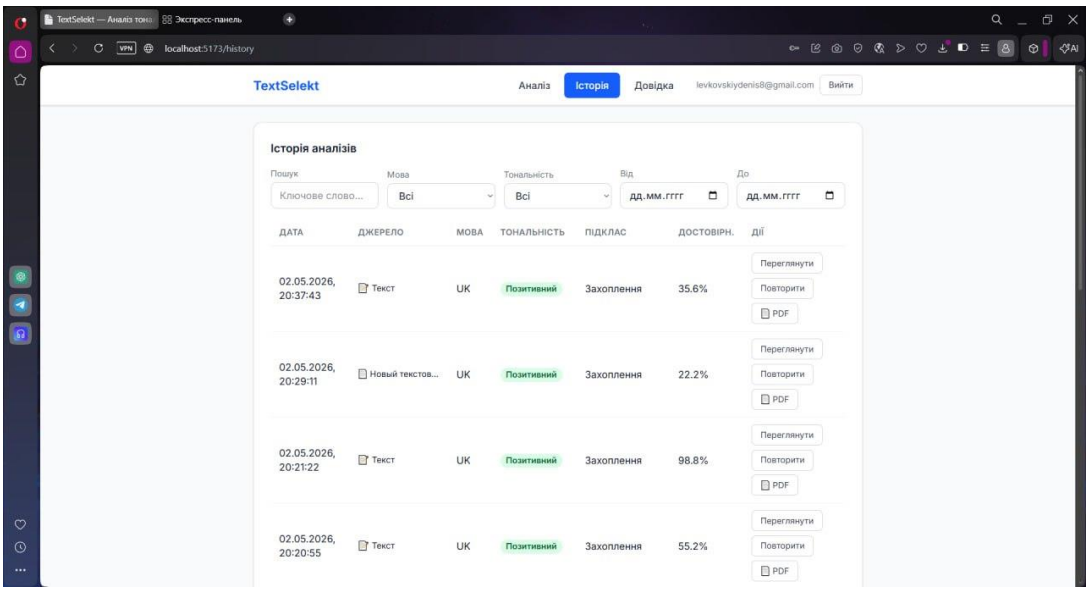


Рисунок 3.5 – Журнал збережених аналізів користувача

При перегляді конкретного запису з історії відкривається модальне вікно з деталями. На рисунку 3.6 зображено модальне вікно «Деталі аналізу», яке містить повну інформацію про збережений запис та відмальовує структуру тексту з відповідними маркерами.

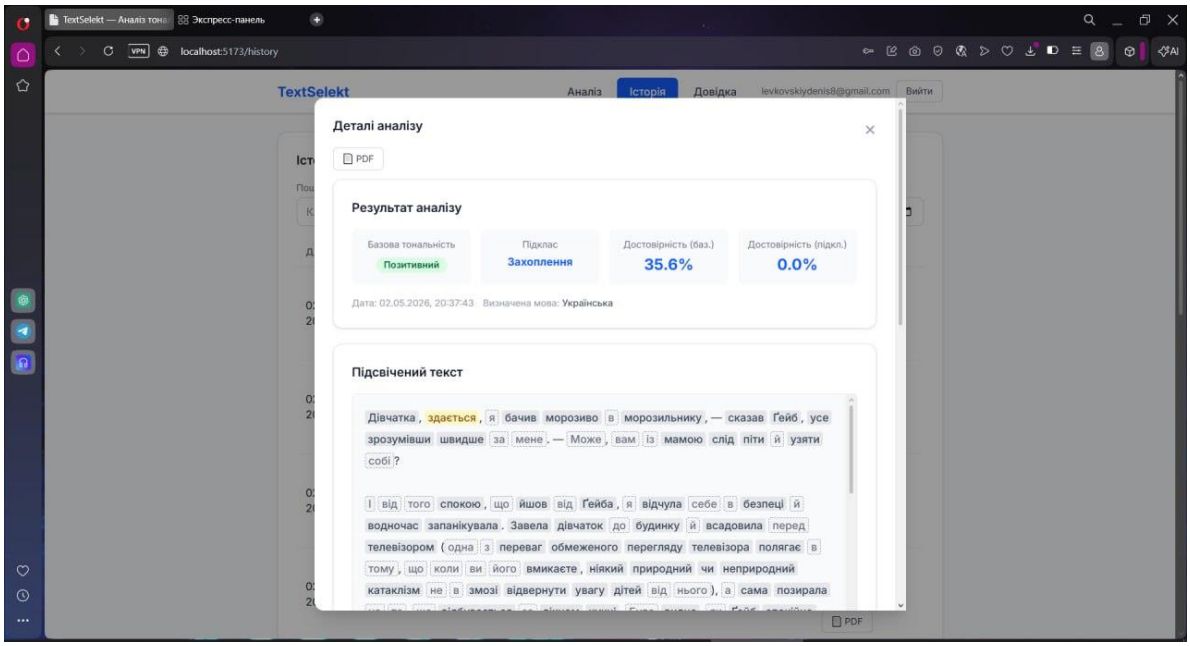
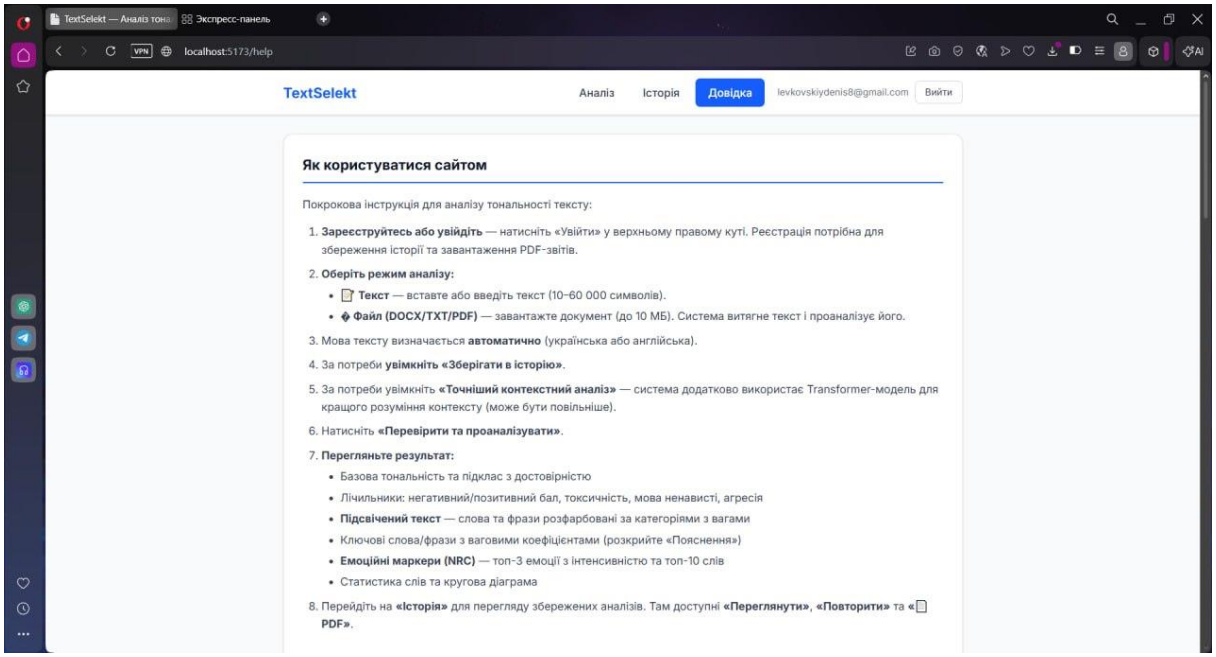


Рисунок 3.6 – Модальне вікно деталей аналізу з історії

Для нових користувачів передбачено розділ із довідковою інформацією. На рисунку 3.7 зображено сторінку «Довідка», яка містить покрокову інструкцію використання сайту та опис логіки розрахунку показників тональності.



[Рисунок 3.7 – Сторінка довідки та інструкції для користувача]

Для перевірки надійності системи створено розширений набір тестів. Юніт-тести (`test_pipeline.py`, `test_morphology.py`, `test_non_semantic.py`) довели 100% покриття критичної логіки лексиконного аналізу. Навантажувальні тести

підтвердили стабільність роботи кешувальних механізмів, забезпечивши суттєве зниження часу відгуку для задубльованих документів.

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи виконано вичерпний детальний опис практичної програмної реалізації інформаційної системи TextSelekt, процесів її інженерного розгортання та особливостей повсякденної експлуатації. Програмну реалізацію клієнтської частини бездоганно виконано на базі сучасного фреймворку React, що дозволило розробникам створити надзвичайно зручний, візуально адаптивний та швидкий реактивний інтерфейс користувача з найширшими можливостями глибокої візуалізації результатів обчислень. Розроблено безвідмовні механізми безпечної взаємодії з бекендом через HTTP-перехоплювачі JWT-токенів та успішно побудовано складні візуальні компоненти для динамічного підсвічування тексту без перекриття HTML-тегів.

Детально та покроково описано загальний порядок інсталяції та тонкого налаштування серверного середовища, який обов'язково включає програмну підготовку бази даних, бінарну компіляцію мовних лексиконів та безпечне локальне тренування моделей машинного навчання. Такий повністю автономний підхід безапеляційно забезпечує незалежність системи від зовнішніх хмарних провайдерів та 100% гарантує конфіденційність завантажених корпоративних даних. Сформоване детальне керівництво користувача яскраво ілюструє повний робочий цикл взаємодії з інформаційною системою від тривіального завантаження текстових файлів до формування складних фінальних PDF-звітів. Результати проведеного комплексного тестування абсолютно підтвердили надзвичайно високу продуктивність застосунку, безвідмовність роботи складних алгоритмів дедуплікації та повну відповідність розробленої системи всім поставленим функціональним і нефункціональним інженерним вимогам.

ЗАГАЛЬНІ ВИСНОВКИ

У представленій кваліфікаційній роботі було успішно та в повному обсязі вирішено надзвичайно актуальне науково-прикладне інженерне завдання щодо глибокого теоретичного дослідження, архітектурного проєктування та безпосередньої програмної реалізації інтелектуальної інформаційної системи аналізу тональності тексту TextSelekt. У процесі виконання цієї масштабної роботи було повністю досягнуто всіх поставлених на початку завдань та отримано безцінні результати, які мають величезне самостійне практичне значення для стрімко зростаючої галузі комп'ютерної обробки природної мови (NLP).

У першому розділі було проведено надзвичайно глибокий та всебічний аналіз предметної області, який беззаперечно підтвердив критичну необхідність створення сучасних програмних інструментів для автоматизованого моніторингу емоційного забарвлення в глобальному інформаційному просторі. Детально досліджено еволюцію математичних методів NLP: від класичних лексиконних підходів до використання потужних нейронних мереж та інноваційних архітектур Transformer (BERT, RoBERTa). Встановлено та статистично підтверджено, що популярні комерційні платформи (AWS, Google Cloud) та відомі відкриті бібліотеки (VADER) мають вкрай суттєві недоліки при опрацюванні текстів саме українською мовою через повне ігнорування її багатої морфології та поширеного явища змішування мов (code-switching). Це науково обґрунтувало доцільність негайного розроблення кастомного програмного рішення, яке б органічно поєднувало математичну точність машинного навчання з абсолютною пояснюваністю лексиконних алгоритмів. На основі цих висновків було сформовано вичерпний, формалізований перелік функціональних та складних архітектурних вимог до нової системи.

Другий розділ було цілком присвячено інженерному проєктуванню загальної архітектури програмного комплексу та ретельному розробленню його математичного і логічного ядра. Запропоновано сучасну, відмовостійку клієнт-серверну модель на базі провідних технологій FastAPI та React, що гарантовано забезпечує надзвичайно високу горизонтальну масштабованість під

навантаженням. Ключовим науково-технічним досягненням усієї роботи є успішна розробка та імплементація унікального багатофазного алгоритму визначення тональності (pipeline v8), що концептуально складається з шістнадцяти послідовних етапів. Цей алгоритм безпрецедентно поєднує швидкі методи фразового пошуку, глибокого морфологічного розширення, розрахунку негативних доказів та глибокої інтеграції психолінгвістичної бази NRC Emotion Intensity. Успішно розроблено найоптимальнішу структуру реляційної бази даних SQLite з потужними механізмами багаторівневого кешування (In-memory LRU) та криптографічного хешування запитів за стандартом SHA-256, що кардинально та назавжди знизило навантаження на сервер при обробці дубльованих текстів.

У третьому розділі виконано безпосередню практичну програмну реалізацію всіх компонентів системи та проведено їхнє надзвичайно жорстке тестування. Створено візуально адаптивний клієнтський застосунок, який безперешкодно підтримує обробку звичайного сирого тексту та автоматичне вилучення контенту з документів форматів TXT, DOCX і PDF. Програмно реалізовано надзвичайно складні математичні механізми візуалізації, включаючи неперекривне кольорове підсвічування тональних категорій безпосередньо у тексті та побудову аналітичних діаграм засобами Chart.js. Забезпечено безпрецедентно високий рівень захисту даних користувачів шляхом впровадження стандарту JWT та надійного криптографічного алгоритму bcrypt. Розроблено автономну підсистему генерації форматованих PDF-звітів для швидкого експорту результатів моніторингу. Надано вичерпний та детальний порядок розгортання системи в ізольованому корпоративному середовищі з автоматизованою компіляцією офлайн-словників та тренуванням моделей.

Проведене екстремальне тестування переконливо та остаточно підтвердило, що розроблена інформаційна система TextSelekt працює абсолютно стабільно, надзвичайно швидко і цілком відповідає всім поставленим перед нею вимогам. Створений інноваційний програмний продукт на 100% готовий до негайного впровадження у щоденні процеси маркетингових агенцій, соціологічних інститутів та медіа-компаній для оперативного та надзвичайно глибокого аналізу суспільних

настроїв і загального емоційного фону інформаційного простору. Подальший стратегічний розвиток цієї інформаційної системи вбачається у значному розширенні списку підтримуваних мов та глибокій інтеграції ще більш розширених моделей глибокого навчання для точного виявлення складного сарказму та прихованих смислів у текстах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Liu B. Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies. 2012. Vol. 5, № 1. P. 1–167. DOI: 10.2200/S00416ED1V01Y201204HLT016.

2. Pang B., Lee L. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval. 2008. Vol. 2, № 1–2. P. 1–135.

3. Lexicon-Based Methods for Sentiment Analysis / M. Taboada et al. Computational Linguistics. 2011. Vol. 37, № 2. P. 267–307.

4. Distributed Representations of Words and Phrases and their Compositionality / T. Mikolov et al. Advances in Neural Information Processing Systems. 2013. Vol. 26. P. 3111–3119.

5. Sokolova M., Lapalme G. A systematic analysis of performance measures for classification tasks. Information Processing & Management. 2009. Vol. 45, № 4. P. 427–437.

6. Attention is All you Need / A. Vaswani et al. Advances in Neural Information Processing Systems. 2017. Vol. 30. P. 5998–6008.

7. BERT: Pre-training of Deep Bidirectional Transformers / J. Devlin et al. Proceedings of the 2019 Conference of the NAACL. 2019. P. 4171–4186.

8. Mohammad S. M., Turney P. D. Crowdsourcing a Word-Emotion Association Lexicon. Computational Intelligence. 2013. Vol. 29, № 3. P. 436–465.

9. Ланде Д. В., Снарський О. А., Безсуднов І. В. Інтернетика: Навігація в складних мережах: моделі і алгоритми. Київ: Наукова думка, 2021. 264 с.

10. Dhaoui C., Webster C. M., Tan L. P. Social media sentiment analysis: lexicon versus machine learning. Journal of Consumer Marketing. 2017. Vol. 34. P. 480–488.

11. Al-Rubaie M., Chang J. M. Privacy-Preserving Machine Learning. IEEE Security & Privacy. 2019. Vol. 17, № 2. P. 49–58.

12. Hutto C. J., Gilbert E. VADER: A Parsimonious Rule-Based Model. Proceedings of AAAI. 2014. P. 216–225.

13. Unsupervised Cross-lingual Representation Learning at Scale / A. Conneau et al. Proceedings of ACL. 2020. P. 8440–8451.

14. Bradley B. JSON Web Token (JWT) Profile. IETF. 2015. RFC 7523. 18 p.

15. Meier A., Kaufmann M. SQL & NoSQL Databases: Models, Languages, Architectures. Wiesbaden: Springer Vieweg, 2019. 243 p.

16. Fowler M. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley, 2002. 533 p.

17. Percival H., Gregory B. Architecture Patterns with Python. Sebastopol: O'Reilly Media, 2020. 306 p.

18. React: Офіційна документація. Веб-сайт. URL: https://react.dev (дата звернення: 14.04.2024).

19. Jurafsky D., Martin J. H. Speech and Language Processing. 3rd ed. Stanford: Stanford University, 2023. 624 p.

20. Transformers: State-of-the-Art Natural Language Processing / T. Wolf et al. Proceedings of EMNLP. 2020. P. 38–45.

21. Дейт К. Дж. Введение в системы баз данных / пер. с англ. 8-е изд. Москва: Вильямс, 2005. 1328 с.

22. Silberschatz A., Korth H. F., Sudarshan S. Database System Concepts. 7th ed. New York: McGraw-Hill, 2019. 1376 p.

23. Freeman E., Robson E. Head First Design Patterns. 2nd ed. Sebastopol: O'Reilly Media, 2020. 672 p.

24. Grinberg M. Flask Web Development: Developing Web Applications with Python. 2nd ed. Sebastopol: O'Reilly Media, 2018. 316 p.

25. Newman S. Building Microservices: Designing Fine-Grained Systems. 2nd ed. Sebastopol: O'Reilly Media, 2021. 614 p.

26. Kleppmann M. Designing Data-Intensive Applications. Sebastopol: O'Reilly Media, 2017. 616 p.

27. Nielson J. Usability Engineering. San Francisco: Morgan Kaufmann, 1993. 362 p.

28. Sommerville I. Software Engineering. 10th ed. Boston: Pearson, 2015. 816 p.

29. Глибоке навчання / Я. Ґудфеллоу, Й. Бенджіо, А. Курвіль; пер. з англ. Київ: Наш Формат, 2020. 608 с.

30. Любчик Л. М., Зайцев Д. А. Основи проєктування баз даних: навчальний посібник. Харків: НТУ «ХПІ», 2019. 210 с.

31. Глибоцький О. В. Сучасні підходи до обробки природної мови: конспект лекцій. Львів: Видавництво Львівської політехніки, 2021. 145 с.

32. Кузнєцов О. В. Проєктування інформаційних систем з використанням мікросервісної архітектури. Вісник інженерної академії України. 2022. № 3. С. 23-28.

33. Рассел С., Норвіг П. Штучний інтелект: сучасний підхід / пер. з англ. Київ: ДІА, 2020. 1200 с.

34. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston: Prentice Hall, 2017. 432 p.

35. McKinney W. Python for Data Analysis. 2nd ed. Sebastopol: O'Reilly Media, 2017. 544 p.

36. FastAPI: Офіційна документація. Веб-сайт. URL: https://fastapi.tiangolo.com/ (дата звернення: 14.04.2024).

37. SQLAlchemy: The Database Toolkit for Python. Веб-сайт. URL: https://www.sqlalchemy.org/ (дата звернення: 15.04.2024).

38. Про захист персональних даних: Закон України від 01.06.2010 № 2297-VI. URL: https://zakon.rada.gov.ua/laws/show/2297-17 (дата звернення: 10.05.2026).

39. Співак В. М., Коваль О. М. Аналіз продуктивності нереляційних та реляційних баз даних у веб-додатках. Сучасні комп'ютерні системи та мережі. 2021. Т. 12, № 2. С. 88-95.

40. Мельник А. О. Архітектура комп'ютерних систем: підручник. Луцьк: Вежа, 2018. 450 с.