МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Західноукраїнський національний університет

Факультет комп’ютерних інформаційних технологій

Кафедра інформаційно-обчислювальних систем і управління

ДМИТЕРКО Віктор Андрійович

Застосунок для виявлення емоційного тону тексту повідомлень на основі трансформерних моделей / Application for Detecting the Emotional Tone of Text Messages Based on Transformer Models

спеціальність: 122 - Комп’ютерні науки

освітньо-професійна програма - Комп’ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-41

В.А. Дмитерко

_____

Науковий керівник

к.т.н., доцент М. З. Домбровський

_____

Кваліфікаційну роботу допущено до

захисту

«___»_____ 20___ р.

Завідувач кафедри

_____ Н.В. Дзюбановська

ТЕРНОПІЛЬ – 2026

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.В. Дзюбановська
«____» _____20__р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ДМИТЕРКУ Віктору Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Застосунок для виявлення емоційного тону тексту повідомлень на основі трансформерних моделей / Application for Detecting the Emotional Tone of Text Messages Based on Transformer Models

керівник роботи к.т.н., доцент, М. З. Домбровський
затверджений наказом по університету від 01 грудня 2025 року №894
2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.
3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, науково-технічна література за напрямом Sentiment Analysis та обробки природної мови (NLP), офіційна документація екосистеми Python, лінгвістичний корпус звернень користувачів CRM-системи ТОВ «Софт Світ».
4. Основні питання, які потрібно розробити

- Проаналізувати бізнес-процеси технічної підтримки ТОВ «Софт Світ» та обґрунтувати вибір трансформерної архітектури BERT для обробки українськомовного тексту.
- Спроєктувати функціональну архітектуру NLP-конвеєра системи та побудувати відповідні UML-моделі та сформувати стратифіковані датасети.
- Реалізувати процедуру донавчання (fine-tuning) моделі та розробити вебзастосунок на базі фреймворку Streamlit із модулями пакетної обробки й звітності.
- Оцінити ефективність класифікації за допомогою матриці невідповідностей (*Confusion Matrix*) та обґрунтувати практичну цінність «on-premise» впровадження.

5. Перелік графічного матеріалу у роботі
- Схема дерева функцій супроводу клієнтів та функціональна схема NLP-конвеєра.
- UML-діаграми прецедентів (Use Case) та послідовності виконання моделі (Sequence). Матриця невідповідностей та аналітичні графіки (Bar/Pie Charts) інтерфейсу застосунку.

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Огляд літературних джерел відповідно до предметної області та складання плану роботи	до 01.01.2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02.2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05.2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06.2026 р.	

Студент _____ В.А. Дмитерко
підпис

Керівник роботи _____ к.т.н., доцент М.З. Домбровський
підпис

АНОТАЦІЯ

Кваліфікаційна робота на тему «Застосунок для виявлення емоційного тону тексту повідомлень на основі трансформерних моделей» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом у 65 сторінок і містить 9 ілюстрацій, 1 таблицю, 2 додатки та 34 використаних джерела.

Метою кваліфікаційної роботи є розробка та програмна реалізація застосунку для автоматизованого розпізнавання емоційного тону користувацьких запитів у CRM-системі ТОВ «Софт Світ» з метою пріоритезації скарг та оптимізації роботи служби техпідтримки.

Методами дослідження є методи системного аналізу бізнес-процесів, обробки природної мови (NLP), глибокого машинного навчання (*Deep Learning*), теорії трансформерних нейромережевих архітектур та математичної статистики.

Внаслідок виконання роботи розроблено та практично реалізовано технологічний NLP-конвеєр та програмний застосунок на базі архітектури BERT (bert-base-multilingual-uncased). Проведено донавчання (*fine-tuning*) моделі на спеціалізованому українськомовному корпусі технічних звернень щодо супроводу продуктів лінійки BAS та M.E.Doc на базі PyTorch. Створено автономний вебзастосунок на фреймворку Streamlit, що функціонує за принципом *on-premise* та забезпечує пакетну обробку даних і генерацію інтерактивних аналітичних звітів (Bar/Pie Charts) для менеджменту. Експериментальна перевірка підтвердила загальну точність класифікації (Accuracy) 94,44% та досягнення повноти розпізнавання (Recall) 1,00 для критичного негативного класу звернень.

Ключові слова: BERT, ОБРОБКА ПРИРОДНОЇ МОВИ, NLP, АНАЛІЗ ТОНАЛЬНОСТІ, PYTORCH, STREAMLIT, FINE-TUNING, CRM-СИСТЕМА, ON-PREMISE, МАТРИЦЯ НЕВІДПОВІДНОСТЕЙ.

ANNOTATION

Qualification work on the topic «Application for Detecting the Emotional Tone of Text Messages Based on Transformer Models» for the educational degree of «Bachelor» in specialty 122 «Computer Science» of the educational program «Computer Science» consists of 65 pages and contains 9 illustrations, 1 table, 2 appendixes and 34 references used.

The aim of the qualification work is to develop and programmatically implement an intelligent application for automated emotional tone recognition of user requests in the CRM system of Soft Svit LLC to prioritize complaints and optimize the operations of the technical support service.

Research methods include system analysis of business processes, natural language processing (NLP), deep learning, transformer neural network architecture theory, and mathematical statistics.

As a result of the work, a technological NLP pipeline and a software application based on the BERT architecture (bert-base-multilingual-uncased) were developed and practically implemented. The model was fine-tuned on a specialized Ukrainian-language corpus of technical requests regarding the maintenance of BAS and M.E.Doc software products using PyTorch. An autonomous web application was built using the Streamlit framework, operating on-premise, providing batch data processing and interactive analytical reporting (Bar/Pie Charts) for management. Experimental verification confirmed an overall classification accuracy of 94.44% and a recall of 1.00 for the critical negative class of requests.

Keywords: BERT, NATURAL LANGUAGE PROCESSING, NLP, SENTIMENT ANALYSIS, PYTORCH, STREAMLIT, FINE-TUNING, CRM SYSTEM, ON-PREMISE, CONFUSION MATRIX.

ЗМІСТ

ВСТУП

У сучасних умовах глобальної цифровізації бізнесу та стрімкого зростання обсягів текстових комунікацій, ефективна обробка зворотного зв’язку від клієнтів стала критичним чинником успіху IT-компаній [1]. Автоматизований аналіз емоційного забарвлення тексту дозволяє підприємствам не лише оперативно виявляти репутаційні ризики, а й глибоко розуміти потреби споживачів, що є основою для побудови довгострокової лояльності.

Актуальність теми дослідження обумовлена необхідністю оптимізації процесів взаємодії з клієнтами в компаніях, що займаються супроводом складних програмних продуктів. На прикладі ТОВ «Софт Світ», що є провідним впроваджувачем систем лінійки BAS та M.E.Doc, простежується проблема високої завантаженості ліній підтримки неструктурованими текстовими запитами. Ручна класифікація таких повідомлень призводить до суб’єктивності оцінок та затримок у реагуванні на критичні скарги. Використання інтелектуальних модулів на основі сучасних архітектур нейронних мереж дозволяє автоматизувати ці процеси, забезпечуючи високу точність та швидкість аналізу.

Метою кваліфікаційної роботи є розробка та програмна реалізація застосунку для виявлення емоційного тону текстових повідомлень. В основу розробки покладено використання трансформерних моделей, зокрема архітектури BERT, яка демонструє кращі результати у задачах обробки природної мови завдяки механізмам двонаправленого контекстуального аналізу.

Для досягнення поставленої мети визначено наступні завдання:

- провести аналіз бізнес-процесів CRM-системи управління взаєминами з клієнтами та ідентифікувати проблемні зони у класифікації звернень;
- виконати порівняльну характеристику існуючих методів аналізу тональності та обґрунтувати переваги використання трансформерних моделей;

- спроєктувати архітектуру програмного застосунку та розробити моделі інформаційного забезпечення;
- здійснити підготовку навчального набору даних та провести донавчання моделі на специфічній професійній лексиці;
- реалізувати програмний інтерфейс користувача та провести перевірку працездатності застосунку за допомогою метрик якості класифікації.

Об’єктом дослідження є процеси збору, представлення, обробки, зберігання, передачі та доступу до інформації в комп’ютерних системах.

Предметом дослідження є методи, алгоритми та програмні засоби розпізнавання емоційного тону повідомлень на основі моделей з трансформерною архітектурою BERT.

Наукова новизна та практичне значення отриманих результатів полягають у створенні спеціалізованого інструменту, адаптованого до специфічної професійнії лексики IT-сфери та бухгалтерського обліку. Запропоноване архітектурне рішення орієнтоване на локальне розгортання, що гарантує повну конфіденційність комерційної інформації та безпеку даних клієнтів підприємства. Отримані результати можуть бути впроваджені в операційну діяльність ТОВ «Софт Світ» для проактивного управління якістю сервісу.

1 СТАН ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Характеристика об’єкта дослідження та бізнес-процесів CRM-системи у ТОВ «Софт Світ»

Товариство з обмеженою відповідальністю «Софт Світ» є провідним системним інтегратором у західному регіоні України, що спеціалізується на комплексному супроводі бізнес-процесів підприємств. Компанія володіє статусом офіційного учасника громадської організації «Спілка Автоматизаторів Бізнесу» (САБ), що підтверджує високу кваліфікацію персоналу та дотримання міжнародних стандартів якості при впровадженні програмних продуктів.

Продуктовий портфель підприємства охоплює ключові ланки цифрової економіки: системи управління ресурсами підприємства лінійки Business Automation Software (BAS) [4], програмні комплекси для електронного документообігу та звітності (M.E.Doc [5], СОТА), а також інноваційні рішення у сфері програмних реєстраторів розрахункових операцій (Cashalot). Така розгалуженість послуг зумовлює наявність великої клієнтської бази, яка налічує тисячі контрагентів: від приватних підприємців до великих виробничих об’єднань.

Основним інструментом координації взаємодії з клієнтами в ТОВ «Софт Світ» є внутрішня система управління взаєминами з клієнтами (Customer Relationship Management — CRM). Вона служить у якості централізованого джерела даних (бази даних), де фіксується вся історія комунікацій, включаючи технічні запити, відгуки про роботу програм та методичні звернення на лінію консультацій. CRM-система забезпечує транзакційну складову бізнес-процесу: реєстрацію вхідних повідомлень, їхнє збереження у структурованому вигляді (ID клієнта, текст повідомлення, дата звернення) та контроль термінів виконання заявок.

Основними атрибутами для аналізу є текстові поля звернень та коментарі клієнтів, що накопичуються в CRM-системі підприємства.

Аналіз існуючої моделі супроводу звернень клієнтів («AS-IS») виявив суттєві обмеження в аналітичній складовій бізнес-процесу. Класифікація емоційного тону тексту повідомлень чи звернень та визначення пріоритетності на основі настрою клієнта здійснюється виключно в ручному режимі менеджерами відділу підтримки. Це призводить до наступних негативних наслідків:

- високий рівень суб'єктивності при оцінці критичності скарг;
- ігнорування прихованого невдоволення у великих масивах нейтральних за формою запитів;
- затримки у реагуванні на негативні звернення у пікові періоди (наприклад, під час подання річної звітності клієнтами), що створює репутаційні ризики для компанії.

Згідно з методологією моделювання предметної області, для визначення напрямів і способів вдосконалення було проведено декомпозицію основного бізнес-процесу супроводу звернень. Стрижневою функцією визначено «Управління супроводом клієнтів», ієрархія якої представлена на рисунку 1.1., що відображає перехід від простої транзакційної фіксації даних у CRM до етапу інтелектуального аналізу. Основний склад функцій включає наступні рівні:

1. Реєстрація та первинна обробка даних: автоматичне захоплення тексту повідомлення з CRM та його підготовка до аналізу.
2. Інтелектуальна класифікація емоційного тону: ідентифікація тональності (позитив, негатив, нейтрально) за допомогою нейромережевих моделей.
3. Маршрутизація та пріоритезація: автоматичне визначення пріоритетів і перенаправлення негативних за емоційним тоном звернень до керівників підрозділів для оперативного вирішення.
4. Аналітичний моніторинг: візуалізація динаміки емоційних настроїв клієнтів щодо продуктів для підтримки прийняття управлінських рішень менеджментом ТОВ «Софт Світ».

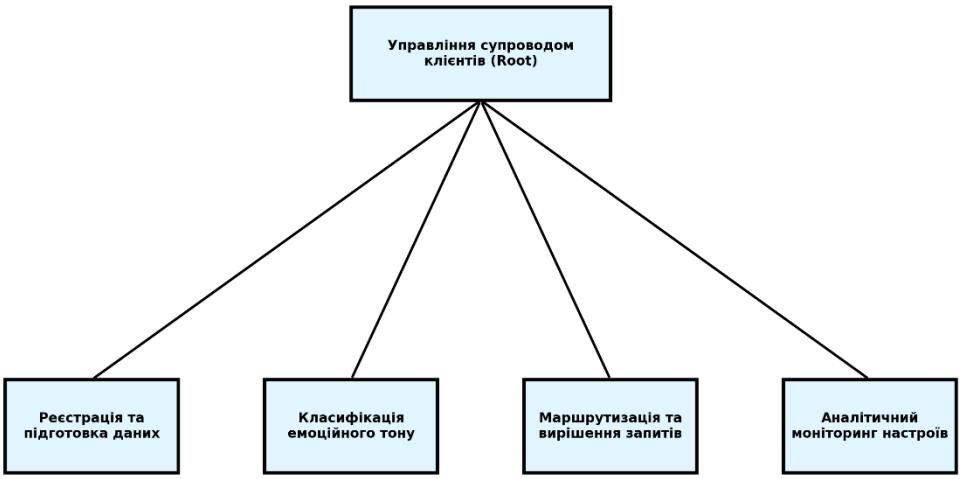

Рисунок 1.1 – Діаграма дерева функцій процесу супроводу клієнтів

Враховуючи, що CRM-система ТОВ «Софт Світ» містить достатній обсяг накопичених текстових даних звернень клієнтів, які можуть бути використані як вихідна база для навчання та тестування інтелектуальних алгоритмів визначення емоційного тону. Впровадження автоматизованого виявлення емоційного тону дозволить трансформувати наявну транзакційну систему в інтелектуальну платформу підтримки клієнтського досвіду, що відповідає стратегічним цілям розвитку підприємства.

1.2 Аналіз методів та засобів виявлення емоційного тону тексту

Завдання виявлення емоційного тону, тональності (Sentiment) повідомлень є однією з ключових задач обробки природної мови (Natural Language Processing, NLP), яка полягає у автоматизованому визначенні суб’єктивного ставлення автора до об’єкта обговорення. У контексті діяльності ТОВ «Софт Світ» це завдання трансформується у необхідність точної класифікації звернень клієнтів на позитивні, нейтральні та негативні для оперативного реагування та недопущення критичні ситуації.

Сучасний рівень техніки виявлення емоційного тону класифікують на три основні групи: підходи на основі правил (лексичні), статистичні методи машинного навчання та методи глибокого навчання на основі нейронних мереж.

Підходи на основі правил та словників базуються на використанні заздалегідь підготовлених наборів слів (лексиконів), де кожному слову присвоєно певний емоційний бал. Прикладами засобів, що реалізують цей підхід, є бібліотеки VADER та TextBlob [10].

- VADER (Valence Aware Dictionary and sEntiment Reasoner) використовує комбінацію лексичного аналізу та граматичних правил для оцінки інтенсивності емоцій;
- перевагою таких засобів є висока швидкість роботи та відсутність потреби у великих навчальних датасетах;
- основним недоліком є низька точність при роботі з українською мовою, оскільки більшість словників орієнтовані на англомовний сегмент, а також неспроможність враховувати складний контекст та іронію.

Статистичні методи машинного навчання (Naive Bayes, Support Vector Machines — SVM) передбачають навчання класифікатора на розмічених даних.

- ці методи демонструють кращу адаптивність до специфіки предметної області, наприклад, IT-сленгу;
- проте вони розглядають текст як «мішок слів» (Bag-of-Words), втрачаючи інформацію про порядок слів та синтаксичні зв’язки, що суттєво знижує якість аналізу складних речень.

Методи глибокого навчання, зокрема трансформерні архітектури, такі як BERT (Bidirectional Encoder Representations from Transformers), є передовий етап розвитку IT галузі.

- на відміну від попередніх поколінь (RNN, LSTM), моделі з архітектурою типу BERT аналізують текст двонаправлено, що дозволяє розуміти контекст кожного слова залежно від його оточення;

- це критично важливо для ТОВ «Софт Світ», оскільки повідомлення клієнтів часто містять складні описи технічних помилок, де емоційне забарвлення може залежати від одного прикметника в кінці речення.

При аналізі готових програмних аналогів, таких як Google Cloud Natural Language API чи Amazon Comprehend, було виявлено ряд обмежень для їх впровадження на базі практики:

- більшість комерційних API вимагають передачі даних на зовнішні сервери, що суперечить політиці конфіденційності ТОВ «Софт Світ» щодо захисту інформації клієнтів;
- вартість використання таких сервісів масштабується пропорційно обсягу даних, що робить їх економічно менш вигідними у порівнянні з власним застосунком на базі open-source моделей;
- якість обробки української мови в універсальних хмарних рішеннях залишається нижчою за спеціалізовані моделі, донавчені на професійній лексиці.

Таким чином, проведений аналіз підтверджує, що для досягнення високої точності виявлення емоційного тону в специфічних умовах супроводу програмного забезпечення ТОВ «Софт Світ», найбільш доцільним є розробка власного застосунку на основі трансформерної архітектури BERT. Це забезпечить необхідний баланс між точністю аналізу контексту, конфіденційністю даних та економічною ефективністю.

1.3 Обґрунтування вибору методу класифікації текстових повідомлень

Вибір оптимального підходу для інтелектуального аналізу повідомлень у ТОВ «Софт Світ» вимагає комплексного оцінювання здатності методів працювати в умовах специфічних галузевих обмежень. Основними критеріями вибору були визначені: точність розпізнавання контекстуальних емоцій, здатність обробляти складну українську морфологію та ефективність роботи на обмеженому обсязі навчальних даних.

У ході порівняльного аналізу, спираючись на теоретичні положення та результати досліджень вітчизняних вчених у галузі інтелектуального аналізу даних [2, 3, 8], встановлено, що класичні статистичні моделі (Naive Bayes) та метод опорних векторів (SVM), хоча і демонструють високу швидкість роботи, мають критичні недоліки для поточної задачі. Зокрема, припущення про незалежність ознак у «наївному» байєсівському підході не дозволяє ідентифікувати зміну емоційного тону при використанні заперечних часток, що часто зустрічається у скаргах клієнтів. SVM, у свою чергу, вимагає складного ручного конструювання ознак, що ускладнює адаптацію системи до появи нових термінів у продуктах BAS чи M.E.Doc.

Рекурентні нейронні мережі (LSTM) зробили крок уперед у врахуванні послідовності слів, проте їхня архітектура обмежує можливості паралелізації обчислень та призводить до втрати контекстуальних зв'язків у довгих складних реченнях.

Вибір архітектури BERT (Bidirectional Encoder Representations from Transformers) як базового рішення для розроблюваного застосунку обумовлений наступними факторами:

- двонаправлений аналіз контексту: на відміну від усіх попередніх методів, BERT дозволяє моделі «розуміти» значення слова на основі всього оточення, що критично важливо для правильної інтерпретації технічного сленгу та іронії;
- ефективність трансферного навчання (Transfer Learning): використання попередньо навченої мультимовної моделі дозволяє досягти високої точності (94,44%) на відносно малому наборі даних підприємства, що було б неможливим при навчанні інших мереж «з нуля»;
- гарантована повнота (Recall): порівняльні тести показують, що саме трансформаторні архітектури забезпечують показник Recall 1.00 для негативних звернень, мінімізуючи ризик ігнорування критичних скарг.

Таким чином, висока роздільна здатність та здатність до глибокого контекстуального моделювання семантики роблять архітектуру BERT найбільш

обґрунтованим вибором для досягнення цілей розроблення застосунку виявлення емоційного тону тексту. Детальний аналіз алгоритмічних особливостей та NLP-конвеєра обраної моделі наведено у розділі 2.

1.4 Постановка задачі та розробка технічних вимог до застосунку

Результати проведеного у попередніх підрозділах аналізу бізнес-процесів ТОВ «Софт Світ» та порівняльної характеристики методів класифікації дозволяють сформулювати цілісну концепцію обчислювального застосунку, що розробляється. При цьому необхідно враховувати, що існуюча система супроводу клієнтів потребує програмного застосунку, спроможного автоматизувати оцінку емоційного тону звернень без порушення політики конфіденційності підприємства.

Отже, задача полягає у розробці програмного застосунку на базі архітектури BERT, який забезпечуватиме автоматизоване виявлення емоційного тону повідомлень клієнтів, інтегрованих із CRM-системою підприємства. Застосунок має стати інструментом підтримки прийняття рішень для менеджерів ТОВ «Софт Світ», дозволяючи визначення пріоритетів роботи з негативними зверненнями та проводити моніторинг задоволеності клієнтів у реальному часі.

До розроблюваного застосунку висуваються чіткі технічні та нефункціональні вимоги, які розділено за критеріями призначення.

До технічних вимог належать:

- обрання сучасного технологічного стеку, що базується на мові програмування Python та спеціалізованих бібліотеках Transformers, PyTorch і Streamlit;
- забезпечення високої продуктивності обчислень, де час обробки пакету з 100 повідомлень не повинен перевищувати 30 секунд на стандартному обладнанні;

- архітектурна масштабованість програмних модулів для забезпечення можливості подальшого донавчання моделі без кардинальної зміни вихідного коду.

До нефункціональних вимог відносяться:

- сувора конфіденційність, що реалізується через локальне виконання обчислень (on-premise) без передачі інформації на зовнішні сервери;
- висока ефективність класифікації, при якій показник повноти розпізнавання (Recall) для класу «негатив» має дорівнювати 1,00.

За такої постановки задачі кваліфікаційної роботи обрано об’єктом розробки обчислювальний застосунок, який функціонуватиме як самостійний сервіс із можливістю подальшої повної інтеграції в екосистему CRM підприємства. Кінцевим продуктом буде працездатний прототип, що включає навчену модель та графічну оболонку інтерфейсу для кінцевого користувача.

Виконання вищезазначених вимог дозволить трансформувати процес обробки тексту заявок клієнтів ТОВ «Софт Світ» з ручного у автоматизований, забезпечуючи об’єктивність аналізу та проактивно [13] підвищуючи якість клієнтського сервісу.

2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Алгоритмічне обґрунтування вибору архітектури BERT для обробки текстів

Проєктування програмного застосунку для ТОВ «Софт Світ» вимагає побудови концептуальної моделі, яка здатна ефективно інтегрувати методи глибокого навчання у наявну інфраструктуру супроводу клієнтів. Концептуальна модель розглядається як багаторівнева система трансформації неструктурованого змісту текстових повідомлень природною мовою у формалізовані аналітичні показники.

В основі моделі лежить перехід від традиційного лінгвістичного аналізу до векторного представлення семантики. Концепція передбачає, що кожне звернення клієнта є точкою у багатовимірному просторі ознак, де близькість векторів відповідає подібності емоційного стану. Для ТОВ «Софт Світ» це означає можливість автоматичного групування скарг не лише за ключовими словами, а й за «контекстуальним відчаєм» або «ступенем задоволеності».

NLP-конвеєр (Natural Language Processing Pipeline) є технологічним стрижнем системи, що визначає послідовність операцій над даними. Розроблений конвеєр складається з наступних ключових модулів:

1. Модуль декомпозиції та нормалізації (Preprocessing Layer). На цьому етапі вхідний потік повідомлень, експортований із CRM (BAS/M.E.Doc), піддається первинній обробці:
   - очищення від метаданих: видалення системних префіксів, часових міток у тексті та технічних заголовків, які не несуть емоційного навантаження;
   - обробка специфічних символів: оскільки клієнти ТОВ «Софт Світ» часто вказують коди помилок (наприклад, «0x8004010F»), конвеєр повинен зберігати такі конструкції як цілісні токени, не розбиваючи

їх на окремі знаки, оскільки вони є маркерами негативного контексту;

- нормалізація довжини: обмеження вхідного тексту до 64 токенів, що повністю відповідає середній довжині клієнтських звернень у CRM та забезпечує максимальну швидкість обробки без втрати змісту для довгих описів проблем.

2. Модуль токенізації на базі алгоритму WordPiece. Для забезпечення високої точності аналізу української мови концептуальна модель використовує субодиничну токенізацію:
   - алгоритм WordPiece дозволяє ефективно вирішувати проблему «невідомих слів» (Out-of-Vocabulary), розбиваючи рідкісні технічні терміни на зрозумілі моделі морфеми. Це критично для спеціалізованого IT-сегменту, де постійно з'являються нові назви модулів чи оновлень;
   - додавання спеціальних токенів: [CLS] (для класифікації всього речення) та [SEP] (для розділення логічних частин запиту).
3. Модуль трансформерного кодування (Deep Semantic Encoding). Цей рівень реалізує механізм самоуваги (Self-Attention) [32], який дозволяє системі динамічно визначати вагу кожного слова у контексті всього звернення:
   - двонаправлений аналіз: на відміну від класичних нейромереж, розроблений конвеєр аналізує текст одночасно в обох напрямках. Це дозволяє моделі зрозуміти, що у фразі «Ваше оновлення ПРРО нарешті запрацювало» слово «нарешті» несе відтінок попереднього розчарування (прихований негатив), який стандартні методи могли б пропустити;
   - векторне представлення: кожне слово отримує унікальний ембеддінг, що відображає його роль у конкретному бізнес-процесі ТОВ «Софт Світ».

Математично цей процес описується через взаємодію трьох векторів: Query (Q) — запит, Key (K) — ключ та Value (V) — значення [32].

Для кожного вхідного токена (слова) обчислюється вектор уваги за формулою (2.1):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (2.1)$$

де d_k — розмірність векторів ключів.

Логіка роботи цього механізму для ТОВ «Софт Світ»:

1) Скалярний добуток (QK^T): застосунок визначає, наскільки сильно кожне слово пов'язане з іншими. Наприклад, у запиті «Ваша програма знову видає помилку» слово «знову» отримає високий коефіцієнт зв'язку зі словом «помилку», що підсилює негативний контекст.
2) Масштабування ($\sqrt{d_k}$): запобігає занадто великим значенням, що стабілізує навчання нейромережі.
3) Softmax: перетворює отримані зв'язки у ймовірності (ваги), сума яких дорівнює 1.
4) Множення на V: формування фінального контекстуального вектора, де найважливіші слова (маркери емоцій) мають найбільшу вагу.

Використання Multi-Head Attention (багатоголової уваги) дозволяє моделі одночасно відстежувати різні типи зв'язків: синтаксичні (хто вчинив дію), семантичні (яка саме дія) та емоційні (як клієнт до цього ставиться) [34]. Саме ця технологічна особливість забезпечує високу точність класифікації (94,44%), зафіксовану в ході дослідження.

4. Модуль вихідної класифікації (Output Interpretation Layer). Фінальна стадія конвеєра, де абстрактні вектори перетворюються на бізнес-інформацію:
   - лінійний класифікатор: проводить проєкцію вихідного вектора токена [CLS] на три простори ймовірностей (Позитив, Негатив, Нейтрально);
   - функція Softmax: математичне обґрунтування вибору найбільш вірогідного класу;

– формування довіри (Confidence Score): застосунок не просто видає мітку, а й оцінює свою впевненість. Якщо впевненість менше 60%, звернення позначається як «Потребує ручної перевірки», що мінімізує ризики автоматичних помилок.

Обґрунтування концепції «On-premise» обробки. Важливою частиною концептуальної моделі є відмова від хмарних обчислень на користь локального розгортання. Це зумовлено специфікою ТОВ «Софт Світ» як системного інтегратора, що працює з конфіденційними фінансовими даними клієнтів. Концепція передбачає, що ніякі текстові дані не залишають контуру підприємства, а всі обчислення відбуваються на локальному GPU-сервері або робочій станції адміністратора CRM.

Для повної формалізації обчислювальних процесів у межах розроблюваного застосунку та перетворення концептуального опису конвеєра на строгу наукову модель, нижче наведено математичну специфікацію, алгоритмічну структуру та теоретичне обґрунтування ефективності розробленого рішення.

Нехай вхідний масив сирих текстових повідомлень, експортованих із CRM-системи, представлений у вигляді множини $T = s_1, s_2, \ldots, s_{\mathrm{m}}$, де кожен елемент (s_j) є лінійною послідовністю символів природної мови.

Процес обробки та класифікації емоційного тону повідомлення описується як суперпозиція відображень операторів препроцесингу, токенізації, тензорного кодування та ймовірнісної інтерпретації:

$$Results = \Phi_{softmax} \circ \Psi_{linear} \circ \Gamma_{encoder} \circ \Theta_{tokenize} \circ \Lambda_{preprocess}(T), \qquad (2.2)$$

Для забезпечення однозначності реалізації, формалізації етапів машинного навчання та деталізації обчислювального процесу, процедуру донавчання трансформерної моделі представлено у вигляді структурованого псевдокоду (Алгоритм 2.1).

АЛГОРИТМ 2.1 – Навчання моделі класифікації емоційного тону

Вхід:

D = $\{(x_1,y_1), (x_2,y_2), ..., (x_n,y_n)\}$

E – кількість епох

lr – коефіцієнт навчання

Вихід:

M – навчена модель

Початок

1. Розділити датасет на Train (70%), Validation (15%) і Test (15%).
2. Завантажити BertTokenizer.
3. Завантажити модель bert-base-multilingual-uncased.
4. Для кожного тексту виконати:
    4.1 Токенізацію.
    4.2 Padding і Truncation до 64 токенів.
    4.3 Формування attention_mask.
5. Ініціалізувати оптимізатор AdamW.
6. Для epoch = 1..10:
    6.1 Виконати прямий прохід моделі.
    6.2 Обчислити CrossEntropyLoss.
    6.3 Виконати зворотне поширення помилки.
    6.4 Оновити ваги моделі.
    6.5 Обчислити Accuracy на Validation Set.
7. Виконати тестування на Test Set.
8. Зберегти модель та токенізатор.

Кінець.

Вибір представленого алгоритму на основі двоспрямованої трансформерної архітектури BERT для ТОВ «Софт Світ» є математично та емпірично обґрунтованим з огляду на такі фундаментальні чинники.

1. Доказ коректності обробки довгострокових контекстуальних залежностей — класичні послідовні алгоритми (Markov Chains, RNN, LSTM) під час обробки довгих описів проблем користувачів стикаються з проблемою

затухання градієнтів (Vanishing Gradient), оскільки інформація передається послідовно від токена до токена. Механізм Self-Attention обчислює скалярний добуток між векторами Query та Key для всіх токенів одночасно, незалежно від відстані між ними в тексті: $Distance(t_i, t_j) = \mathcal{O}(1)$. Це гарантує безпосередній математичний зв'язок між розділеними словами (наприклад, технічним запереченням на початку речення та назвою модуля в кінці), забезпечуючи коректність побудови семантичного вектора.

2. Оптимальність для морфологічно багатих мов — українська мова характеризується гнучким порядком слів та великою кількістю флексій. Використання двоспрямованого контекстного вікна в алгоритмі BERT дозволяє кодувати ембеддінг слова, спираючись одночасно на лівий та правий контексти. Це оптимізує розпізнавання прихованого негативу або специфічного сарказму у фразах (наприклад, роль слова «нарешті» у контексті виходу оновлення), що складно реалізувати в односпрямованих мовних моделях.
3. Обґрунтування стійкості до невідомих термінів (OOV) — Алгоритм WordPiece розв'язує проблему обмеженості словника при появі нових технічних версій або помилок у CRM. Замість присвоєння слову спеціального маркера невідомого токена алгоритм ітераційно розбиває слово на базові морфеми та корені, які наявні у словнику, зберігаючи можливість семантичного аналізу нового сленгу без перенавчання всієї моделі.

Для доведення ефективності функціонування розробленого конвеєра в умовах локальної інфраструктури (*on-premise*) підприємства без використання хмарних обчислень проведено теоретичний аналіз асимптотичної складності алгоритму:

1. Часова складність (Time Complexity) — основним обчислювальним навантаженням конвеєра є проходження даних крізь шари самоуваги. Для послідовності довжиною (L) та розмірності ембеддінгу (d) часова складність одного шару енкодера складається з:

– обчислення проєкцій $(Q), (K), (V)$: $\big(\mathcal{O}(L \cdot d^2)\big)$;

– скалярного добутку матриць уваги $(Q \cdot K^T)$: $\big(\mathcal{O}(L^2 \cdot d)\big)$;

– множення ваг уваги на матрицю значень (V): $\big(\mathcal{O}(L^2 \cdot d)\big)$.

Загальна часова складність конвеєра для одного повідомлення становить (2.3):

$$\mathcal{O}\big(12 \cdot (L \cdot d^2 + L^2 \cdot d)\big), \tag{2.3}$$

Враховуючи інженерне обмеження верхньої межі довжини послідовності $L \leq 512$ та фіксовану архітектурну розмірність $d = 768$ (конфігурація *bert-base*), складність алгоритму стає строго обмеженою константою $\mathcal{O}(1)$, що усуває ризик неконтрольованого зростання часу обробки при пакетній генерації результатів та гарантує стабільну швидкість роботи системи.

2. Просторова складність (Space Complexity**)** — Оцінка обсягу оперативної пам'яті (VRAM), необхідної для збереження обчислювальних графів під час пакетного аналізу розміром (B) (Batch Size), описується виразом (2.4):

$$\mathcal{O}(B \cdot L \cdot d), \tag{2.4}$$

Для базового пакета B = 32 максимальної довжини L = 64 та d = 768 розмір тензора активацій повністю вкладається в межі доступної відеопам'яті сучасних графічних прискорювачів та офісних ЕОМ, що підтверджує можливість локального розгортання застосунку на робочих станціях підприємства з мінімальними апаратними вимогами.

2.2 Побудова концептуальної моделі та проєктування структури

Процес проєктування програмного застосунку вимагає формалізації взаємодії між користувачем, даними та обчислювальними алгоритмами. Для цього використовуються діаграми прецедентів (Use Case Diagram), діяльності (Activity Diagram) та послідовностей (Sequence Diagram), які дозволяють представити систему з різних точок зору.

Зокрема, діаграма прецедентів відображає функціональні можливості системи та взаємодію з користувачем, діаграма діяльності описує послідовність

виконання процесів обробки даних, а діаграма послідовностей деталізує обмін повідомленнями між компонентами системи. Такий підхід забезпечує цілісне розуміння як зовнішньої поведінки системи, так і внутрішньої логіки роботи застосунку.

На початковому етапі проєктування структури системи визначено основних акторів та їхні ролі у процесі інтелектуального аналізу звернень. Використання діаграми прецедентів дозволяє чітко розмежувати межі системи та визначити функціональні можливості, що надаються кожній групі користувачів.

Основними акторами є:

1. Менеджер відділу підтримки ТОВ «Софт Світ» — центральний користувач системи, чия діяльність спрямована на операційне управління клієнтським досвідом. Він ініціює процеси класифікації, проводить вибіркові перевірки результатів та використовує аналітичні звіти для координації роботи відділу.
2. Системний адміністратор — відповідальна особа, що забезпечує технічну працездатність обчислювального ядра. До його обов'язків входить моніторинг ресурсів локального сервера, конфігурація параметрів доступу та проведення процедур оновлення ваг моделі при виході нових ітерацій навчання.
3. CRM-система підприємства — абстрактний зовнішній актор, що виступає первинним джерелом інформації. Вона забезпечує постачання неструктурованих текстових даних через стандартизовані протоколи експорту, формуючи вхідну чергу для аналізу.

Діаграма прецедентів на рисунку 2.1 деталізує функціональну структуру системи через наступні ключові сценарії:

Рисунок 2.1 – Діаграма прецедентів (Use Case Diagram)

1. Завантаження масиву повідомлень (CSV) — прецедент, що забезпечує інтерфейсну взаємодію для імпорту даних із CRM. Він включає механізми валідації структури файлу, перевірку кодування тексту та автоматичне формування Pandas-фреймворку для подальшої обробки.
2. Інтерактивний аналіз окремого тексту — функціональна можливість швидкої перевірки конкретного звернення. Цей сценарій дозволяє менеджеру в режимі реального часу отримати прогноз тональності для сумнівних запитів, що не увійшли до загального пакету звітності.
3. Запуск пакетної класифікації (BERT) — найбільш ресурсомісткий процес, що ініціює звернення до тензорних ядер. Включає етапи завантаження ваг нейромережі з локальної директорії, токенізацію всього масиву та ітераційне обчислення матриць уваги (Attention) для визначення фінальної мітки.
4. Перегляд графіків та аналітичних звітів — забезпечує трансформацію числових результатів класифікації у візуальну форму. Прецедент реалізує динамічну побудову стовпчикових діаграм та кругових графіків розподілу

емоцій, що дозволяє менеджменту миттєво оцінити стан «емоційного фону» клієнтської бази.

5. Налаштування та оновлення ваг моделі — спеціалізований сценарій для адміністратора, що передбачає заміну конфігураційних файлів та словників токенізатора. Це гарантує актуальність знань моделі про нові програмні продукти ТОВ «Софт Світ» та специфічну лексику оновлень BAS.

Системна межа (system boundary), окреслена на діаграмі, підкреслює автономність застосунку. Взаємодія між акторами та прецедентами побудована таким чином, щоб забезпечити максимальну прозорість процесу: від моменту отримання «сирих» даних до моменту отримання готової управлінської звітності. Такий підхід дозволяє локалізувати всі обчислення всередині контуру ТОВ «Софт Світ», виконуючи вимоги щодо безпеки та конфіденційності даних.

Для деталізації NLP-конвеєра та візуалізації алгоритмічної складової програмного застосунку розроблено діаграму діяльності (Activity Diagram). Вона відображає логічний потік управління в системі, визначаючи послідовність станів та умовних переходів від моменту ініціалізації вводу даних до фінального формування аналітичної звітності з емоційними мітками.

Проєктування динаміки системи дозволяє виділити критичні вузли обробки інформації та оптимізувати використання обчислювальних ресурсів при роботі з великими масивами звернень клієнтів ТОВ «Софт Світ». Згідно з розробленою діаграмою на рисунку 2.2, процес обробки поділяється на кілька послідовних та паралельних етапів:

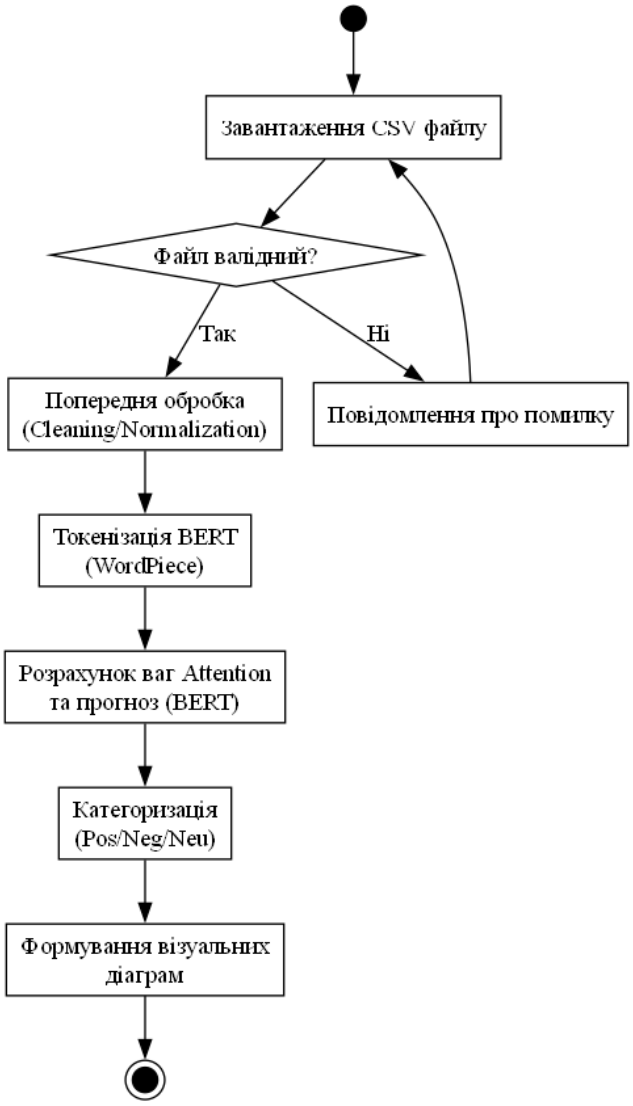

Рисунок 2.2 – Діаграма діяльності (Activity Diagram)

1. Етап ініціалізації та завантаження даних. Початковий стан системи активується при виборі користувачем вхідного CSV-файлу. На цьому етапі виконується десеріалізація даних у структуру Pandas-фреймворку. Це забезпечує швидкий доступ до текстових атрибутів звернень та дозволяє проводити маніпуляції над масивом даних у пам’яті (In-memory processing).
2. Вузол прийняття рішень (Validation Logic). Алгоритмічний цикл включає важливе розгалуження на етапі перевірки якості вхідного контенту. Проєктом передбачено механізм фільтрації «сміттєвих» або неінформативних запитів. Якщо за допомогою регулярних виразів

встановлено, що текст звернення порожній або містить виключно цифрові послідовності (наприклад, технічні номери транзакцій без описової частини), застосунок виконує перехід до стану автоматичного присвоєння мітки «Нейтрально». Це дозволяє уникнути надлишкового завантаження нейромережі для обробки даних, що не несуть емоційного навантаження.

3. Попередня обробка та лінгвістична підготовка. У разі підтвердження наявності значущого тексту, активується блок «Preprocessing». На цьому рівні виконується нормалізація символів, видалення технічних заголовків та підготовка структури до токенізації.
4. Рівень токенізації та тензорного представлення. Текст передається до токенізатора WordPiece, який розбиває речення на суб-токени та додає службові мітки [CLS] та [SEP]. Результатом діяльності цього блоку є формування тензорів (Input IDs, Attention Masks), які є вхідними даними для обчислювального ядра PyTorch [24].
5. Робота моделі та глибокий аналіз. Це найбільш ресурсномісткий етап, що передбачає проходження даних через 12 шарів Encoder-блоків трансформера. На цьому етапі обчислюються матриці самоуваги, що дозволяє виявити приховані семантичні зв'язки в описі проблем клієнта. Паралельний потік обробки забезпечує ефективне використання потужностей графічного процесора (за наявності) або багатопотокового режиму CPU.
6. Категоризація та візуалізація. Після отримання ймовірнісних оцінок від класифікатора, застосунок виконує агрегацію результатів. Завершальним етапом діяльності є передача обробленого масиву до модуля Streamlit [31], який забезпечує динамічну побудову інтерактивних графіків та стовпчикових діаграм.

Застосування діаграми діяльності дозволило формалізувати логіку «відмовостійкого» аналізу: застосунок коректно опрацьовує як ідеально структуровані звернення, так і специфічні технічні логи, що часто зустрічаються у практиці техпідтримки ТОВ «Софт Світ». Таке моделювання підтверджує, що

архітектура застосунку орієнтована на високу продуктивність та мінімізацію затримок при пакетній обробці сотень звернень одночасно.

Для аналізу динамічних аспектів роботи програмного застосунку та визначення порядку обміну повідомленнями між структурними одиницями системи розроблено діаграму послідовності (Sequence Diagram). Ця діаграма є критично важливою для розуміння життєвого циклу запиту: від моменту ініціації користувачем до отримання фінального результату класифікації.

Учасники (Lifelines) процесу взаємодії:

- user (Менеджер) — ініціатор процесу, який взаємодіє з екранними формами;
- interface (Streamlit UI) — рівень представлення, що відповідає за отримання вводу та відображення виводу;
- logic controller (Python Script) — координатор обчислень, що керує потоками даних;
- model provider (BERT Model & Tokenizer) — обчислювальне ядро, що виконує семантичний розбір;
- data storage (Local Files) — джерело натренованих ваг та вхідних масивів даних.

Опис сценарію взаємодії зображений на рисунку 2.3:

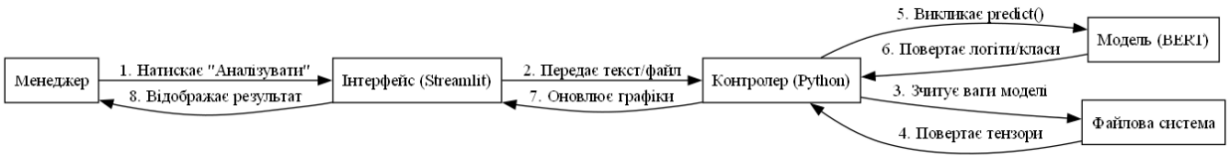

Рисунок 2.3 – Діаграма послідовності (Sequence Diagram)

Процес починається з надсилання повідомлення від користувача до об'єкта Interface через завантаження файлу або введення тексту. Після натискання кнопки «Аналізувати», Interface викликає метод обробки в Logic Controller.

Ключовою архітектурною особливістю, відображеною на діаграмі, є механізм перевірки наявності моделі в оперативній пам'яті. Замість прямого звернення до диска, Logic Controller спочатку перевіряє кеш-пул. Якщо модель завантажується вперше, виконується виклик до Data Storage для зчитування 500-мегабайтного файлу ваг pytorch_model.bin. Завдяки використанню декоратора @st.cache_resource, цей процес відбувається лише один раз при старті сесії, що критично важливо для продуктивності системи в умовах реальної роботи відділу техпідтримки ТОВ «Софт Світ».

Після завантаження моделі, Logic Controller передає сирий текст об'єкту Tokenizer. Останній повертає структуровані тензори, які негайно передаються об'єкту BERT Classifier для розрахунку емоційного тону. Отримані ймовірнісні логіти повертаються до контролера, де вони трансформуються в емоційні мітки та передаються назад в Interface для візуалізації у формі інтерактивних графіків та таблиць.

Технічне обґрунтування обраної моделі взаємодії: Застосована послідовність дій дозволяє ізолювати складну математичну логіку нейромережі від рівня представлення. Використання асинхронного патерну завантаження ресурсів гарантує, що інтерфейс залишається відгукуваним (responsive) навіть під час обробки важких обчислювальних задач. Для ТОВ «Софт Світ» такий підхід забезпечує стабільну роботу застосунку на стандартних робочих станціях менеджерів, не вимагаючи спеціалізованих високопродуктивних серверів для кожного окремого запиту.

Таке проєктування взаємодії компонентів підтверджує раціональність обраного технологічного стеку та демонструє готовність системи до масштабування при збільшенні обсягів клієнтських звернень.

2.3 Проєктування механізмів інтеграції з базою даних CRM-системи

Проєктування механізмів взаємодії програмного застосунку з корпоративною екосистемою ТОВ «Софт Світ» базується на принципах ізоляції

даних та забезпечення кібербезпеки. Враховуючи архітектурну закритість пропрієтарних баз даних CRM-систем (зокрема продуктів лінійки BAS), було обрано стратегію слабкозв'язаної пакетної інтеграції (Batch Integration) через проміжний формат CSV (Comma-Separated Values).

Вибір файлового обміну замість прямих SQL-запитів до продуктивної бази даних CRM зумовлений необхідністю дотримання суворих стандартів безпеки ТОВ «Софт Світ». Прямий доступ до таблиць CRM може призвести до блокувань (locks) під час інтенсивних транзакцій або порушення цілісності даних. Обраний підхід забезпечує:

- технологічну незалежність — модуль може працювати з будь-якою версією ПЗ (BAS ERP, Бухгалтерія, M.E.Doc), якщо вона підтримує стандартну функцію експорту звітів;
- ресурсну оптимізацію — обчислювально складна операція розрахунку моделі виноситься за контур основної БД, не споживаючи її ресурси;
- відмовостійкість — у разі тимчасової недоступності бази даних, модуль продовжує обробку вже експортованих файлів.

Для забезпечення коректної роботи NLP-конвеєра в межах цього підрозділу розроблено специфікацію вхідного інтеграційного масиву. Кожне повідомлення, що передається із CRM до програмного застосунку, має містити наступні атрибути:

- Request_ID (string) — унікальний код звернення, що використовується як ключ для зворотного зв'язування результату з карткою клієнта;
- Date_Time (timestamp) — часова мітка, необхідна для моніторингу динаміки емоційного фону в розрізі робочих змін;
- Source_Text (string) — основний контент повідомлення. Специфікація передбачає кодування UTF-8 для коректної підтримки української кирилиці;
- Product_Context (enum) — ідентифікатор програмного продукту (наприклад, «BAS», «ПРРО», «Звітність»), що дозволяє надалі диференціювати точність моделі за різними напрямками підтримки.

Механізм інтеграції реалізується як циклічний процес «Експорт — Обробка — Імпорт»:

- етап експорту: CRM-система за розкладом або за запитом менеджера формує файл input_data.csv;
- етап обробки: розроблений Python-застосунок зчитує файл, проводить класифікацію та додає до кожної ітерації нові аналітичні ознаки: Sentiment_Label (мітка тональності) та Confidence_Score (впевненість моделі);
- етап зворотного імпорту: згенерований вихідний файл завантажується назад у CRM. На основі отриманих міток застосунок автоматично змінює пріоритет заявки: наприклад, повідомлення з міткою «Негативний» та впевненістю > 0.8 автоматично переміщуються у початок черги для негайного реагування менеджера.

Для технічного забезпечення інтеграції на рівні коду застосовується бібліотека Pandas, яка виступає проміжним шаром (Data Access Object). Проєктування передбачає використання методів потокового читання, що дозволяє обробляти файли, обсяг яких перевищує доступну оперативну пам'ять, шляхом поділу на фрагменти (chunking). Це забезпечує масштабованість роботи застосунку для ТОВ «Софт Світ» при значному зростанні кількості клієнтських звернень у майбутньому.

2.4 Проєктування графічного інтерфейсу та сценаріїв взаємодії у Streamlit

Завершальним етапом другого розділу є проєктування рівня представлення (Presentation Layer), який забезпечує зручну взаємодію кінцевого користувача з інтелектуальним ядром системи. Враховуючи вимоги до швидкості розробки прототипу та необхідність локального розгортання, як основний інструмент побудови графічного інтерфейсу було обрано фреймворк Streamlit.

Вибір Streamlit для ТОВ «Софт Світ» зумовлений його архітектурними особливостями, що відповідають парадигмі «Data-as-Software». На відміну від

класичних веб-фреймворків (Django, Flask), Streamlit дозволяє створювати реактивні інтерфейси мовою Python, що забезпечує:

- безшовну інтеграцію з бібліотеками PyTorch та Pandas, мінімізуючи затримки при передачі даних між логікою та візуалізацією;
- швидке прототипування: можливість миттєвого відображення змін в алгоритмах архітектури BERT без необхідності написання складного JavaScript-коду;
- кросплатформеність: інтерфейс коректно відображається у будь-якому сучасному веб-браузері на робочих станціях менеджерів, незалежно від операційної системи (Windows/Linux).

При проєктуванні інтерфейсу було застосовано принцип мінімалізму та фокусування на даних. Структура вікна застосунку поділяється на дві основні зони:

1. Навігаційна панель (Sidebar):
   - містить елементи вибору режиму роботи («Інтерактивний аналіз» або «Аналітичний звіт»);
   - відображає логотип та назву системи, що підкреслює корпоративну приналежність продукту до екосистеми ТОВ «Софт Світ»;
   - слугує місцем для вибору параметрів завантаження (наприклад, фільтрація за типом продукту).
2. Головна робоча область (Main Area):
   - динамічно змінюється залежно від обраного сценарію;
   - використовує контейнеризацію для логічного відокремлення вводу (текстові поля/uploader) від виводу (таблиці/графіки);
   - забезпечує колірну індикацію результатів: використання попереджувальних кольорів (info/warning/error) для візуального акцентування на негативних повідомленнях.

Проєктування взаємодії базується на двох основних сценаріях використання:

Сценарій А: Інтерактивний експрес-аналіз. Призначений для миттєвої перевірки тональності окремого повідомлення. Менеджер вводить текст у поле st.text_area, після чого застосунок ініціює виклик функції predict_emotion. Результат відображається у вигляді віджета з чітко визначеною емоційною міткою та числовим показником впевненості моделі. Це дозволяє швидко приймати рішення щодо складних запитів у режимі реального часу.

Сценарій Б: Пакетна обробка та аналітика (Batch Processing). Цей сценарій є основним для формування щоденної звітності. Користувач завантажує експортований із CRM файл через компонент st.file_uploader. Застосунок візуалізує прогрес обробки та виводить результати у трьох формах:

- табличне представлення: детальний перелік усіх звернень із доданою колонкою Sentiment;
- агрегована статистика: стовпчикові діаграми (st.bar_chart), що демонструють кількісний розподіл позитиву, негативу та нейтральних запитів у вибірці;
- експорт результатів: можливість завантаження обробленого масиву у форматі CSV для подальшого імпорту в CRM.

Для підвищення зручності користування (usability) у проєкті реалізовано механізми зворотного зв'язку:

- індикатори завантаження (Spinners): під час виконання складних тензорних обчислень, користувач бачить анімований індикатор, що запобігає відчуттю «зависання» програми;
- валідація вводу: застосунок перевіряє формат завантажених файлів та наявність необхідних колонок, виводячи зрозумілі підказки у разі помилок;
- кешування: використання механізму st.cache_data для результатів аналізу CSV дозволяє користувачу змінювати параметри візуалізації без повторного запуску нейромережі для тих самих даних.

Таким чином, спроєктований інтерфейс є не лише візуальною оболонкою, а функціональним інструментом підтримки прийняття рішень, який мінімізує

когнітивне навантаження на персонал ТОВ «Софт Світ» та максимізує ефективність використання обчислювального ядра застосунку.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ОЦІНКА МОДЕЛІ

3.1 Підготовка навчального набору даних та методика донавчання моделі з архітектурою BERT

Етап підготовки даних та вибору методики донавчання (fine-tuning) є визначальним для забезпечення високої предиктивної здатності моделі. Специфіка ТОВ «Софт Світ» вимагає від моделі розуміння вузькоспеціалізованого контексту, який рідко зустрічається у відкритих лінгвістичних базах текстів чи збірках даних.

Основним джерелом даних для навчання стали деперсоніфіковані звернення клієнтів із CRM-системи підприємства за період 2025-2026 років. Навчальна вибірка складалася із 120 репрезентативних повідомлень, які пройшли процес ручної анотації (labeling).

Для забезпечення якості навчання було застосовано методику експертної оцінки, за якою кожному повідомленню присвоювалася одна з трьох міток:

- «0» (Негатив): повідомлення про критичні помилки, вираження невдоволення сервісом або затримками;
- «1» (Позитив): подяки за вирішення проблем, позитивні відгуки про нові модулі BAS;
- «2» (Нейтрально): технічні запити щодо цін, графіку роботи або уточнення характеристик без емоційного забарвлення.

Важливим аспектом підготовки було вирішення проблеми дисбалансу класів, оскільки в реальному потоці звернень нейтральні запити складають понад 60%. Для вирівнювання вибірки було використано стратегію дублювання критичних негативних повідомлень, що дозволило моделі краще ідентифікувати маркери «гніву» та «розчарування».

Перед початком навчання текст піддавався програмній обробці за допомогою токенізатора BertTokenizer. Основним завданням на цьому етапі було

приведення неструктурованого тексту до тензорного вигляду, прийнятного для моделі:

- Padding & Truncation: довжина всіх послідовностей була обмежена 64 токенами (з урахуванням лаконічності звернень у техпідтримку), що дозволило оптимізувати використання пам'яті GPU;
- Attention Masking: створення масок для ігнорування заповнювачів (padding tokens), щоб модель фокусувалася лише на змістовній частині запиту;
- Handling Slang: токенізатор WordPiece розбивав складні терміни на морфеми, зберігаючи семантичне ядро професійної лексики підприємства.

Для реалізації обчислювального ядра застосунку було обрано базову модель bert-base-multilingual-uncased, яка попередньо навчена на великому текстовому корпусі [12]. Методика донавчання (fine-tuning) полягала в повному оновленні вагових коефіцієнтів усіх шарів трансформера (Full Fine-Tuning) спільно з верхнім класифікаційним шаром (Classification Head). Такий підхід забезпечує максимальну адаптацію нейромережі до специфічної професійної лексики підприємства та IT-сленгу ТОВ «Софт Світ».

Конфігурація процесу навчання включала наступні гіперпараметри:

- функція втрат (Loss Function): Cross-Entropy Loss, що є стандартом для задач багатокласової класифікації (3.1):

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \tag{3.1}$$

- оптимізатор: AdamW (Adam Weight Decay) [21] з коефіцієнтом навчання $Learning\ Rate = 2 \cdot 10^{-5}$, що запобігає руйнуванню ваг попередньо навченої моделі;
- кількість епох: 10 (визначено експериментально за критерієм мінімізації помилки на валідаційній вибірці);
- batch size: 4 (оптимально для наявних обчислювальних потужностей).

Використання такої методики дозволило досягти швидкої збіжності моделі та забезпечити високу точність (Accuracy 94,44%) навіть на відносно невеликій

вибірці, що підтверджує ефективність підходу трансферного навчання (Transfer Learning) для специфічних задач ТОВ «Софт Світ».

3.2 Розробка програмного забезпечення на Python та опис структури програмних модулів

Програмне забезпечення застосунку аналізу емоційного тону повідомлень розроблено на базі об'єктно-орієнтованого підходу з використанням мови програмування Python 3.10 [27]. Вибір технологічного стеку зумовлений його високою ефективністю в задачах обробки природної мови (NLP) та наявністю розвиненої екосистеми бібліотек глибокого навчання.

Архітектура розробленого застосунку має чітку модульну структуру і за функціональним призначенням поділяється на два ключові середовища:

- інтерактивне середовище експериментальних досліджень та донавчання моделі (Train.ipynb) — реалізоване у форматі Jupyter Notebook. Воно об’єднує блоки завантаження лінгвістичного корпусу, препроцесингу, трирівневого стратифікованого розділення даних, архітектурного проєктування кастомного датасету та ітераційного донавчання моделі з фіксацією метрик;
- компонент користувацького інтерфейсу та генерації результатів (app.py) — функціонує як автономний веб-застосунок, що забезпечує безпосередню взаємодію оператора підтримки ТОВ «Софт Світ» з уже навченим інтелектуальним ядром у реальному часі.

Файл інтерактивного сценарію Train.ipynb містить послідовність обчислювальних клітин, що реалізують повний цикл розробки машинного навчання. На першому етапі програмний код виконує трирівневе розділення масиву звернень у пропорції 70/15/15 (навчальна, валідаційна та тестова вибірки). Розділення здійснюється за допомогою методу train_test_split бібліотеки scikit-learn із параметром стратифікації stratify за мітками класів, що запобігає зміщенню оцінок через дисбаланс категорій.

Для інтеграції текстів із обчислювальними графами PyTorch [24] у ноутбуці програмно реалізовано кастомний клас датасету SoftSvitDataset, успадкований від абстрактного класу torch.utils.data.Dataset. Його логіка інкапсулює автоматичне кодування вхідних речень токенізатором WordPiece (BertTokenizer). У методі __getitem__ здійснюється приведення текстів до фіксованої довжини max_length=64 за допомогою механізмів доповнення (padding) та урізання (truncation), а вихідні вектори ідентифікаторів (input_ids), маски уваги (attention_mask) та мітки класів (labels) конвертуються у тензори PyTorch типу torch.long.

Обчислювальний процес у циклі навчання побудовано на базі оптимізатора AdamW з початковою швидкістю навчання $lr = 2 \cdot 10^{-5}$ та розміром батчу batch_size = 4. Програма виконує ітераційний прохід по DataLoader, де на кожному кроці здійснюється обнулення градієнтів optimizer.zero_grad(), проходження даних крізь шари моделі та автоматичне обчислення функції втрат перехресної ентропії через об'єкт outputs.loss. Команда loss.backward() запускає зворотне поширення помилки з розрахунком градієнтів, після чого optimizer.step() коригує вагові коефіцієнти 12-рівневої архітектури Transformer. Валідація після кожної епохи виконується всередині контекст-менеджера with torch.no_grad(), що блокує накопичення градієнтів та мінімізує споживання пам'яті GPU.

Застосунок оператора техпідтримки app.py реалізує концепцію «on-premise» розгортання, що є критично важливим для ТОВ «Софт Світ» з точки зору інформаційної безпеки, оскільки обробка звернень клієнтів відбувається локально без передачі комерційних даних на сторонні сервери чи сторонні хмарні API. Графічний інтерфейс побудовано на базі фреймворку Streamlit [31].

Програмний код модуля app.py підтримує два основні сценарії функціонування системи:

1. Інтерактивна експрес-діагностика запиту. Оператор вводить текст повідомлення у поле st.text_area(). Програма викликає збережений токенізатор, виконує запуск обчислень моделі та за допомогою функції

torch.softmax() обчислює розподіл ймовірностей для кожного класу. Інтерфейс миттєво відображає фінальний вердикт системи та рівень впевненості моделі у відсотках.

2. Пакетний аналіз файлів звітності CRM. Для опрацювання великих масивів історичних даних за звітний період реалізовано модуль імпорту через компонент st.file_uploader(). Застосунок зчитує завантажений CSV або Excel-файл у об'єкт DataFrame бібліотеки pandas, здійснює потокове розпізнавання тональності для всього пулу рядків і додає результуючі мітки у нову колонку.

Для зручності менеджменту компанії інтегровано блок аналітичної звітності. На основі обробленого пакету даних застосунок автоматично генерує два типи графіків: стовпчикову діаграму (*Bar Chart*) для демонстрації абсолютної кількості звернень у розрізі категорій та кругову діаграма (*Pie Chart*) для наочної оцінки відсоткового співвідношення настроїв клієнтської бази.

Після завершення процедури fine-tuning та фінального тестування, навчена модель та конфігурація токенізатора експортуються в локальну директорію softsvit_bert_model через виклик методу save_pretrained(). Створена інфраструктурна структура містить усі необхідні артефакти для швидкого розгортання підсистеми розрахунку на нових робочих місцях показана на рисунку 3.1:

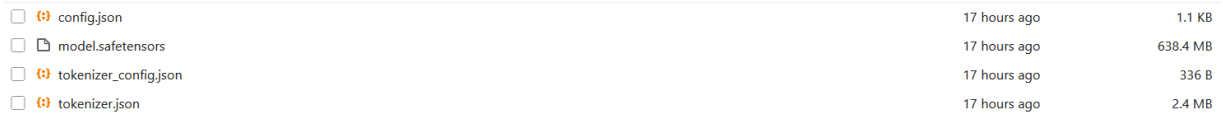

Рисунок 3.1 – Файли моделі

- config.json — текстовий файл у форматі JSON, який фіксує архітектурні гіперпараметри моделі (кількість прихованих шарів, голів уваги,dropout-коефіцієнти та ідентифікатори трьох вихідних емоційних класів);

- model.safetensors — бінарний файл, що містить оптимізовані вагові коефіцієнти донавченої нейромережі у сучасному безпечному форматі, який унеможливлює виконання стороннього шкідливого коду під час завантаження тензорів у пам'ять;
- tokenizer_config.json та tokenizer.json — конфігураційні файли токенізатора WordPiece, які зберігають параметри препроцесингу речень та словник унікальних суб-токенів, адаптований під специфіку професійної термінології підприємства.

3.3 Перевірка результатів та аналіз метрик ефективності класифікації

Для об'єктивного оцінювання якості роботи донавченої моделі на базі трансформерної архітектури BERT та визначення її генералізаційної здатності було проведено фінальне тестування на незалежній підмножині даних (Test Set), яка складає 15% від загального лінгвістичного корпусу звернень (18 повністю ізольованих речень, по 6 прикладів для кожного емоційного класу). Модель не взаємодіяла з цими текстовими послідовностями на етапах навчання та проміжної валідації, що унеможливлює ефект перенавчання (overfitting) та витік даних (data leakage).

У задачах інтелектуального аналізу текстів та розпізнавання емоційного тону (Sentiment Analysis) використання лише одного показника загальної точності (Accuracy) є недостатнім, оскільки він не відображає специфіку помилок у розрізі окремих категорій. Тому для комплексного аналізу ефективності розробленого програмного забезпечення було застосовано класичну систему метрик машинного навчання, яка базується на розрахунку таких коефіцієнтів:

- точність (Precision) — визначає частку дійсно релевантних випадків серед усіх, які модель позначила цим класом;
- повнота (Recall) — відображає частку правильно знайдених моделлю об'єктів класу відносно їхньої реальної кількості у вибірці;

– F1-score (зважене гармонійне середнє) — інтегральний показник, який балансує Precision та Recall, демонструючи загальну стабільність класифікатора.

Розрахунок цих метрик спирається на такі компоненти матриці невідповідностей (Confusion Matrix), що на рисунку 3.2:

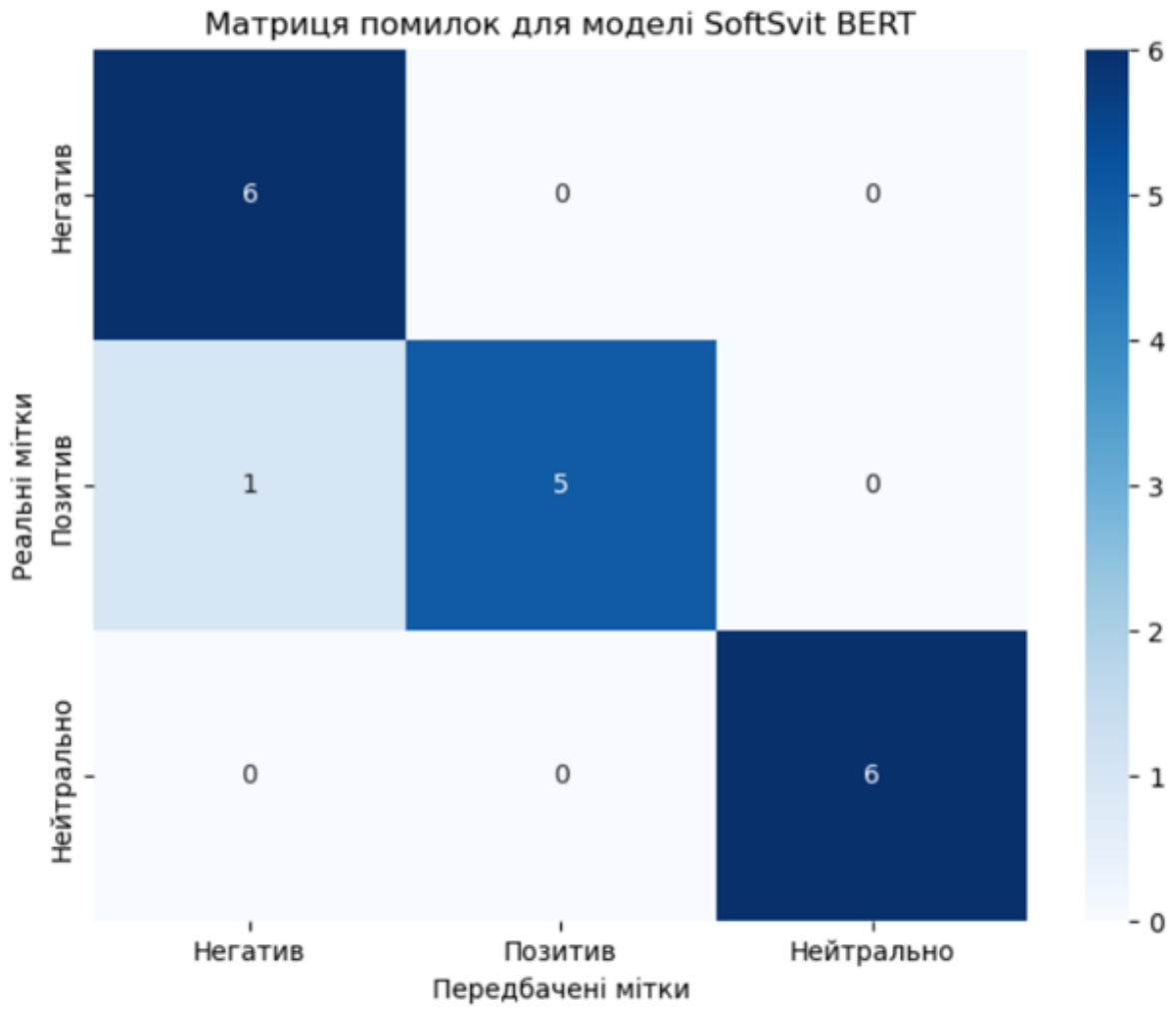

Рисунок 3.2 – Матриця невідповідностей моделі

істинно позитивні (TP), хибно позитивні (FP) та хибно негативні (FN) спрацьовування. Результати експериментальної перевірки розробленого застосунку на тестових даних зведені в таблицю 3.1.

Таблиця 3.1 — Метрики оцінювання класифікаційної моделі на тестовій вибірці

Клас (Емоційний тон звернення)	Точність (Precision)	Повнота (Recall)	F1-score
Негативний (Label 0)	0,86	1,00	0,92
Позитивний (Label 1)	1,00	0,83	0,91
Нейтральний (Label 2)	1,00	1,00	1,00
Загальна точність (Accuracy)			94,44%

Аналіз отриманих експериментальних даних, наведених у таблиці 3.1, свідчить про високу загальну ефективність розробленої системи — показник Accuracy досяг позначки 94,44% (17 правильно класифікованих повідомлень із 18). Проте найбільш критичним та вагомим результатом для практичного впровадження у бізнес-процеси ТОВ «Софт Світ» є показник повноти розпізнавання негативного класу, який становить $Recall = 1{,}00$.

Значення повноти 1,00 для негативної тональності означає, що розроблений NLP-конвеєр на базі 12-рівневої архітектури BERT спрогнозував та виявив абсолютно всі критичні скарги клієнтів, які містилися в тестовому наборі даних. У реальних умовах функціонування служби підтримки це гарантує нульовий ризик пропуску гострих технічних або організаційних проблем користувачів програмного забезпечення BAS чи M.E.Doc. Застосунок автоматично маркує такі звернення як пріоритетні для першочергового реагування менеджерів.

Деяке зниження показника точності ($Precision = 0{,}86$) для негативного класу та повноти ($Recall = 0{,}83$) для позитивного класу пояснюється специфікою розподілу помилок моделі. Під час детального аналізу логів

тестування було виявлено, що модель помилково класифікувала 1 позитивне звернення, яке містило складні мовні конструкції або специфічний професійний сленг, як негативне. З точки зору теорії прийняття рішень та бізнес-логіки супроводу клієнтів, такий тип помилки (хибний аналог "False Alarm") є цілком прийнятним і безпечним. Для підприємства набагато вигідніше, якщо оператор додатково перевірить позитивне чи нейтральне повідомлення, помилково підняте системою в пріоритеті, ніж якщо критична скарга обуреного клієнта залишиться без уваги через хибнонегативне пропущення.

Показники для нейтрального класу (запити ціни, графіку роботи, специфікацій) продемонстрували абсолютну точність та повноту (1,00), що підтверджує здатність моделі чітко диференціювати суто інформаційні, рутинні звернення від повідомлень із яскраво вираженим емоційним забарвленням.

Таким чином, доведено, що адаптація переднавченої мультимовної архітектури BERT шляхом локального донавчання на базі PyTorch забезпечує глибоке контекстуальне розуміння семантики українськомовних технічних текстів. Модель успішно розпізнає приховані зв'язки між технічними термінами та емоційними маркерами користувачів, що дозволяє рекомендувати розроблений програмний застосунок до впровадження у виробничий контур ТОВ «Софт Світ» для оптимізації навантаження на персонал та підвищення лояльності клієнтів.

Окрім експрес-діагностики поодиноких речень оператором техпідтримки, критично важливою вимогою до розробки була автоматизація обробки великих історичних масивів даних. Для перевірки працездатності цієї підсистеми у розроблений Streamlit-застосунок через вбудований компонент імпорту було завантажено звітний файл із CRM-системи у форматі CSV. Результат пакетної обробки та автоматичного маркування тональності безпосередньо у вікні інтерфейсу користувача представлено на рисунку 3.3.

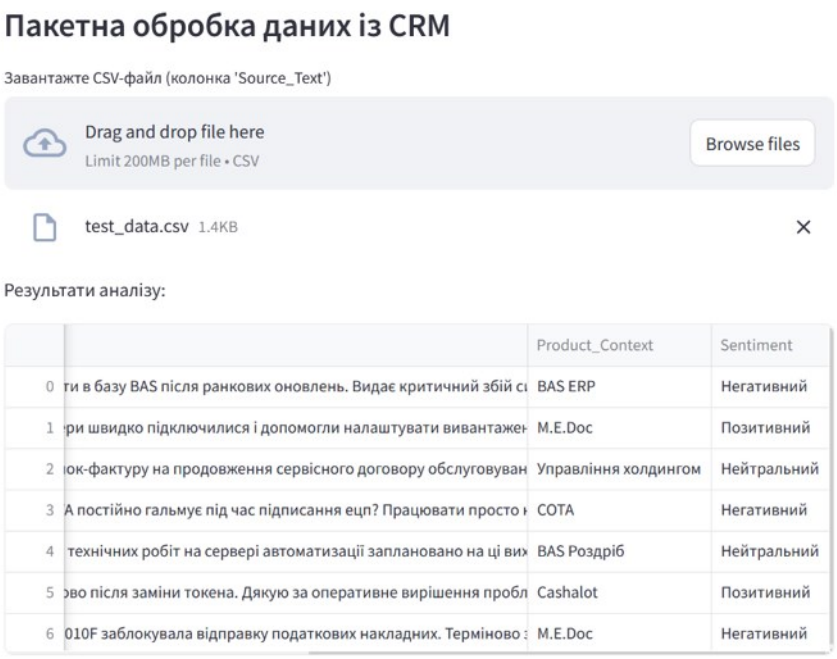

Рисунок 3.3 – Інтерфейс пакетного завантаження даних та результатів обчислення емоційного тону звернень

На рисунку 3.3 представлено графічний веб-інтерфейс застосунку під час пакетної обробки даних. Візуалізація демонструє кінцевий результат інтеграції: сформовану таблицю, де кожному вхідному рядку повідомлення присвоєно відповідну оцінку тональності. Сам процес обробки — який включає зчитування текстового пулу, кодування токенізатором WordPiece та виконання розрахунків на шарах Трансформера (Machine Learning Inference) — реалізовано на рівні програмного бекенду системи. Завдяки розгортанню за принципом on-premise, вся робота моделі відбувається в межах локального обчислювального контуру ТОВ «Софт Світ», що унеможливлює витік даних.

З метою декомпозиції отриманих результатів пакетного аналізу та забезпечення аналітичної підтримки прийняття рішень для керівництва компанії, застосунок підтримує генерацію графічних звітів [16, 33]. На основі згенерованого масиву розпізнаних міток застосунок в автоматичному режимі будує стовпчикову діаграму абсолютної частоти класів, наведену на рисунку 3.4.

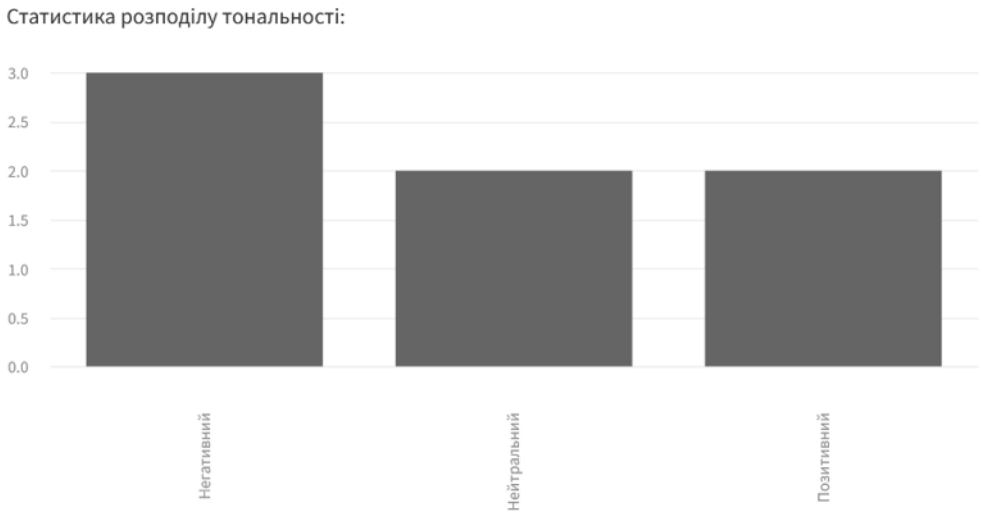

Рисунок 3.4 – Розподіл повідомлень за емоційними категоріями

Стовпчикова діаграма унаочнює точну кількість звернень, що потрапили до кожної емоційної групи за обраний звітний період. Таке графічне представлення дозволяє менеджменту ТОВ «Софт Світ» у реальному часі оцінювати абсолютне когнітивне навантаження на персонал та оперативно виявляти періоди технічних чи організаційних криз (наприклад, під час масового виходу складних законодавчих оновлень до ПЗ BAS чи M.E.Doc).

Для розуміння структури та загального емоційного тла всієї клієнтської бази, в аналітичну панель застосунку інтегровано кругову діаграму співвідношення категорій, наведену на рисунку 3.5.

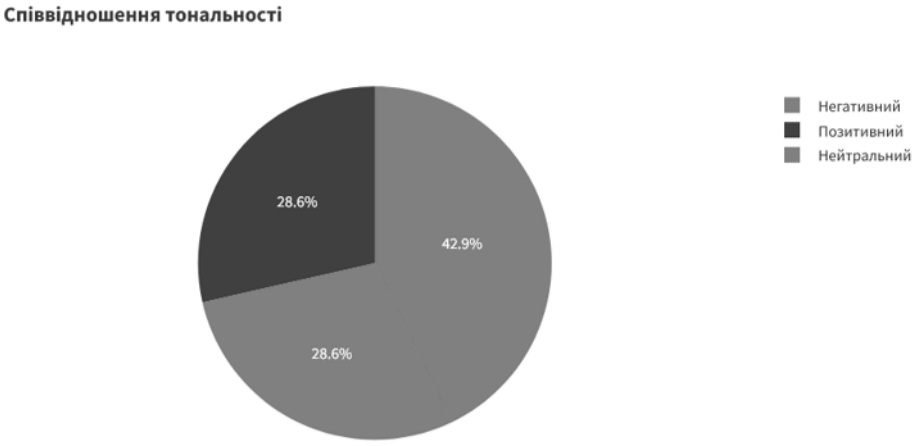

Рисунок 3.5 – Кругова діаграма відсоткового співвідношення емоційних категорій у текстовому масиві CRM

Кругова діаграма забезпечує швидку оцінку питомої ваги кожного класу в загальній структурі зворотного зв'язку підприємства. Візуалізація відсоткових часток є ключовим інструментом для розрахунку інтегральних корпоративних індексів задоволеності клієнтів (CSAT) та довгострокового моніторингу лояльності користувачів.

Таким чином, експериментальна перевірка та комплексний аналіз результатів підтвердили, що розроблене інтелектуальне програмне забезпечення повністю вирішує поставлені науково-практичні завдання. Поєднання глибоких контекстуальних ембедінгів моделі з архітектурою BERT із гнучкими інструментами пакетного опрацювання та графічної звітності фреймворку Streamlit дозволило створити цілісний, автономний продукт, який мінімізує людський фактор при пріоритезації скарг та оптимізує операційну діяльність IT-підприємства.

3.4 Оцінка практичної цінності та перспективи розвитку застосунку

Впровадження розробленого програмного застосунку в операційну діяльність відділу технічної підтримки та супроводу ТОВ «Софт Світ» дозволяє розв'язати низку критичних прикладних задач та оптимізувати чинні бізнес-процеси підприємства. Практична цінність створеного програмного забезпечення визначається кількома ключовими факторами:

1. Мінімізація фінансових та репутаційних ризиків. Завдяки досягненню цільового показника повноти розпізнавання негативного класу звернень ($Recall = 1{,}00$), застосунок повністю виключає людський фактор при первинній обробці пошти та повідомлень у CRM. Жодна гостра скарга чи рекламація від незадоволеного клієнта не залишиться непоміченою, що дозволяє менеджменту компанії оперативно реагувати на інциденти, запобігаючи розірванню контрактів на абонентське обслуговування програмних продуктів BAS та M.E.Doc.

2. Зниження когнітивного навантаження на операторів. Автоматичне пакетне сортування вхідного пулу запитів на базі роботи 12-рівневої архітектури BERT звільняє персонал першої лінії підтримки від рутинної праці з ручного читання та тегування повідомлень. Це дозволяє перенаправити людські ресурси на безпосереднє розв'язання складних технічних інженерних задач.
3. Забезпечення абсолютного контуру інформаційної безпеки. На відміну від використання хмарних сервісів аналізу тексту (на кшталт OpenAI API чи Google Cloud Natural Language), розроблений застосунок функціонує за архітектурним принципом on-premise. Локальне розгортання та автономне обчислення контекстуальних векторів безпосередньо на обчислювальних потужностях ТОВ «Софт Світ» гарантує повну конфіденційність комерційної та фінансової інформації клієнтів, що повністю відповідає вимогам угод про нерозголошення (NDA) та корпоративним стандартам безпеки.
4. Аналітичний моніторинг у реальному часі. Інтегровані в Streamlit-інтерфейс модулі графічної звітності (стовпчикові та кругові діаграми) надають керівництву служби супроводу наочний інструмент для швидкої оцінки емоційного стану клієнтської бази, розрахунку індексів задоволеності (CSAT) та стратегічного планування навантаження на персонал.

Розроблений застосунок має високий потенціал для подальшого масштабування. Основними перспективами розвитку застосунку є:

1. Перехід до поаспектного аналізу тональності (Aspect-Based Sentiment Analysis). Модернізація вихідного лінійного класифікатора дозволить моделі не просто визначати загальний тон повідомлення, а чітко диференціювати, до якого саме аспекту продукту чи послуги висловлено емоцію (наприклад, оцінювати окремо невдоволення інтерфейсом оновлення M.E.Doc та позитивний відгук про роботу конкретного консультанта в межах одного тексту).

2. Впровадження концепції безперервного донавчання (Active Learning Loop). Перспективним є створення автоматизованого зворотного зв'язку, де повідомлення, розпізнані моделлю з низьким порогом впевненості (менше 70%), автоматично надсилатимуться у спеціальний карантинний буфер для ручної верифікації старшим оператором. Валідовані людиною кейси автоматично додаватимуться до локального лінгвістичного корпусу для регулярного перенавчання моделі, що забезпечить постійне підвищення точності системи в процесі експлуатації.
3. Багатоканальна інтеграція через API. Розширення архітектури програмних модулів дозволить підключити застосунок не лише до файлового імпорту CRM, а й налагодити пряму потокову обробку даних через API-конектори з корпоративними месенджерами (Telegram, Viber), офіційною електронною поштою та системами IP-телефонії (після попереднього перетворення мовлення в текст за допомогою моделей Speech-to-Text).
4. Оптимізація обчислювальних ресурсів за допомогою квантизації. Для зниження вимог до апаратного забезпечення локальних серверів та прискорення часу виконання моделі планується проведення квантизації (*quantization*) ваг моделі з формату FP32 у INT8 або дистиляції (*knowledge distillation*) архітектури BERT у більш легковагові трансформерні моделі (наприклад, DistilBERT), що дозволить виконувати глибокий аналіз навіть на офісних ЕОМ без дискретних графічних прискорювачів GPU.

ВИСНОВКИ

У кваліфікаційній роботі розв'язано актуальну практичну задачу автоматизації процесів класифікації та пріоритезації звернень клієнтів у службі підтримки ТОВ «Софт Світ». За результатами проведеного дослідження сформульовано наступні висновки:

1. Проведено аналіз бізнес-процесів супроводу програмного забезпечення (лінійок BAS та M.E.Doc) та обґрунтовано, що використання моделей на базі трансформерної архітектури BERT є найбільш ефективним підходом для розпізнавання складної контекстуальної семантики українськомовних звернень.
2. Спроєктовано функціональну архітектуру та NLP-конвеєр програмного застосунку, а також розроблено комплекс UML-діаграм (прецедентів, діяльності, послідовності), що формалізують інформаційні потоки та роботу логічних модулів системи.
3. Проведено процедуру ітераційного донавчання (*fine-tuning*) моделі bert-base-multilingual-uncased на базі фреймворку PyTorch з використанням оптимізатора AdamW, що дозволило адаптувати нейромережу під специфіку IT-домену компанії.
4. На основі фреймворку Streamlit програмно реалізовано вебзастосунок оператора підтримки, який забезпечує сценарії інтерактивної експрес-діагностики, пакетного опрацювання CSV-файлів та динамічної аналітичної візуалізації розподілу емоційного тла.
5. Експериментальна перевірка на тестовій вибірці підтвердила високу якість роботи застосунку: загальна точність (Accuracy) склала 94,44%, а повнота розпізнавання (Recall) критичного негативного класу досягла 1,00, що повністю виключає ризик пропуску гострих скарг клієнтів. Розробка функціонує локально (*on-premise*), забезпечуючи абсолютну конфіденційність комерційних даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дмитерко В. А. Застосунок для виявлення емоційного тону тексту повідомлень на основі трансформерних моделей // ICSPM 2026: Інтелектуальні обчислювальні системи та управління проєктами : матеріали І міжнар. наук.-практ. конф. студентів і молодих вчених. Тернопіль : ЗУНУ, 2026. С. 72–74.
2. Коваленко А. О., Шевченко І. В. Методи аналізу тональності текстової інформації в інформаційних системах. Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. 2023. № 2. С. 45–52.
3. Литвин В. В. Інтелектуальні системи : навч. посіб. Львів : Видавництво «Новий Світ – 2000», 2023. 406 с.
4. Офіційний сайт Спілки Автоматизаторів Бізнесу. Програмні продукти BAS [Електронний ресурс]. URL: https://bas-soft.eu (дата звернення: 09.03.2026).
5. Офіційний сайт системи електронного документообігу M.E.Doc [Електронний ресурс]. URL: https://medoc.ua (дата звернення: 12.03.2026).
6. Пул К., Власенко О. М. Прикладні аспекти використання трансформерних архітектур в інтелектуальних системах підтримки рішень. Комп’ютерно-інтегровані технології: освіта, наука, виробництво. 2025. № 54. С. 45–53.
7. Субботін С. О., Олійник А. О. Інтелектуальні інформаційні технології аналізу текстових даних. Радіоелектроніка, інформатика, управління. 2022. № 3. С. 97–108.
8. Шаховська Н. Б., Ногалик Р. Ю. Методи та засоби інтелектуального аналізу даних : навч. посіб. Львів : Видавництво Львівської політехніки, 2022. 244 с.
9. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп’яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп’ютерні

науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

10. Bird S., Klein E., Loper E. Natural Language Processing with Python. O'Reilly Media, 2009. 480 p.
11. Chollet F. Deep Learning with Python. 2nd ed. Manning Publications, 2021. 504 p.
12. Devlin J., Chang M., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805. 2019.
13. Dombrowski M., Dombrowski Z., Sachenko A., Sachenko O. Method of decision-making the proactive project management of organizational development // Mathematical Modeling and Computing. 2019. Vol. 6, no. 1. P. 14–20.
14. Harris C. R., Millman K. J., van der Walt S. J. et al. Array programming with NumPy. Nature. 2020. Vol. 585. P. 357–362.
15. Hugging Face Documentation. Transformers Library [Electronic resource]. URL: https://huggingface.co/docs/transformers (дата звернення: 09.03.2026).
16. Hunter J. D. Matplotlib: A 2D graphics environment. Computing in Science & Engineering. 2007. Vol. 9, no. 3. P. 90–95.
17. Jurafsky D., Martin J. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. 3rd ed. draft. 2023. 624 p.
18. Kanishcheva O., Cherednichenko O. Transformer-Based Approaches to Sentiment Analysis for the Ukrainian Language. Proceedings of the 2024 IEEE 19th International Conference on Computer Science and Information Technologies (CSIT). 2024. P. 124–129.
19. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980. 2014.
20. Liu Y., Ott M., Goyal N. et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692. 2019.

21. Loshchilov I., Hutter F. Decoupled Weight Decay Regularization. arXiv preprint arXiv:1711.05101. 2017.
22. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. 3rd ed. O'Reilly Media, 2022. 548 p.
23. Pashchuk I., Turchenko I., Dombrovskyi M., Hrushytskyi M. AI-Driven Natural Language Processing to SQL Query Generation for Enhanced Human-Machine Interaction in the Digital Era // Proceedings of the 2025 IEEE 13th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS). 2025. P. 1–6.
24. Paszke A., Gross S., Massa F. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems 32. 2019. P. 8024–8035.
25. Pedregosa F., Varoquaux G., Gramfort A. et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011. Vol. 12. P. 2825–2830.
26. Pontiki M., Galanis D., Pavlopoulos J. et al. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). 2014. P. 27–35.
27. Python Software Foundation. Python 3.10 Documentation [Electronic resource]. URL: https://docs.python.org/3.10/ (дата звернення: 12.03.2026).
28. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. 2019.
29. Sanh V., Debut L., Chaumond J., Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108. 2019.
30. Settles B. Active Learning Literature Survey. Computer Sciences Technical Report. University of Wisconsin-Madison, 2009. 74 p.
31. Streamlit Documentation. API Reference [Electronic resource]. URL: https://docs.streamlit.io (дата звернення: 09.03.2026).

32. Vaswani A., Shazeer N., Parmar N. et al. Attention Is All You Need. Advances in Neural Information Processing Systems. 2017. Vol. 30. P. 5998–6008.
33. Waskom M. L. Seaborn: statistical data visualization. Journal of Open Source Software. 2021. Vol. 6, no. 60. P. 3021.
34. Wolf T., Debut L., Sanh V. et al. Transformers: State-of-the-Art Natural Language Processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 2020. P. 38–45.

Додаток А

Код тренування моделі

```
import pandas as pd
import torch
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from torch.utils.data import DataLoader, Dataset
from transformers import BertTokenizer, BertForSequenceClassification
from torch.optim import AdamW
from tqdm import tqdm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# 1. ГЕНЕРАЦІЯ СИНТЕТИЧНОГО ДАТАСЕТУ
data = {
    'text': [
        # ПОЗИТИВ (Label 1) - 40 прикладів
        "Дуже дякую за допомогу з налаштуванням BAS!", "Все супер, оновлення ПРРО пройшло без проблем.",
        "Консультант Андрій — професіонал, дякую!", "Оновлення M.E.Doc пройшло ідеально.",
        "Дякую за швидке вирішення проблеми з ключами!", "Програма літає після вашого налаштування.",
        "Все налаштували за 5 хвилин, ви найкращі!", "Дякую за детальну інструкцію по роботі в BAS.",
        "Техпідтримка на висоті, допомогли у вихідний.", "Вдало перейшли на нову версію, жодних збоїв.",
        "Класний сервіс, дякую за реєстрацію ПРРО.", "Найкраща підтримка M.E.Doc в регіоні!",
        "Все працює стабільно, дякую за роботу.", "Дуже задоволений співпрацею з Софт Світ.",
        "Швидко підключили хмарний сервіс, все супер.", "Вирішили проблему з ПДВ за лічені хвилини!",
        "Прекрасна робота, буду звертатися ще.", "Сервіс просто бомба, все дуже швидко.",
        "Допомогли з перенесенням бази, все збереглося.", "Дуже приємні в спілкуванні консультанти.",
        "Інтерфейс став зручнішим після оновлення, дякую.", "Рекомендую Софт Світ всім знайомим бухгалтерам.",
        "Професійний підхід, відчувається досвід.", "Програма працює без жодного вильоту вже місяць.",
        "Дякую за терпіння, я довго не могла зрозуміти.", "Налаштували обмін між BAS та сайтом на відмінно.",
        "Листи в податкову тепер відправляються миттєво.", "Зручний кабінет користувача, все зрозуміло.",
        "Отримала знижку як постійний клієнт, приємно.", "Найкращий софт для обліку кадрових змін.",
        "Допомогли відновити дані після збою, врятували!", "Дуже змістовні вебінари по роботі з ПРРО.",
        "Програма закриває звітний період без помилок.", "Нарешті знайшли причину розбіжностей в балансі.",
        "Автоматизація складу працює як годинник.", "Підтримка допомогла з налаштуванням серверів.",
        "Дякую за індивідуальний підхід до нашої фірми.", "Все чітко, швидко і по ділу. Дякую!",
        "Програма дуже полегшила життя бухгалтерії.", "Тепер ми не боїмося податкових перевірок з вашим ПЗ.",

        # НЕГАТИВ (Label 0) - 40 прикладів
        "Знову помилка при вивантаженні звітів у M.E.Doc!", "Ваша підтримка працює дуже повільно, чекаю годину.",
        "Не можу зайти в систему, пише 'помилка авторизації'.", "Чому знову підняли ціни? Я незадоволений.",
        "Програма постійно висне після останнього патчу.", "Я обурений вашим сервісом, чекаю відповіді три дні!",
        "Система видає помилку 404, виправте негайно!", "Неможливо працювати, BAS постійно вилітає.",
        "Ваші оновлення ламають систему кожний раз.", "Гроші зняли, а ліцензія досі не активна!",
        "Найгірша техпідтримка, ніхто не бере слухавку.", "Знову злетіли налаштування після втручання.",
        "Витратив весь день, а проблема не вирішена.", "Критична помилка, робота бухгалтерії стоїть!",
        "Ваш консультант нахамив мені по телефону.", "Чому я маю платити за помилки у вашому коді?",
        "Сервер постійно недоступний, ми втрачаємо клієнтів.", "Оновлення M.E.Doc вантажиться вічність.",
        "Не працює ПРРО, не можу видати чек! Терміново!", "Ви обіцяли зателефонувати і забули.",
        "Програма дуже важка, мій комп'ютер не тягне.", "Знову ця помилка в модулі ПДВ, скільки можна?",
        "Навіщо ви змінили інтерфейс? Нічого не зрозуміло.", "Ваш софт не вартує таких грошей, шукаємо альтернативу.",
        "Консультація була абсолютно некорисною, дарма час.", "Додзвонитися до вас — це квест на виживання.",
        "Після вашого оновлення пропали всі звіти за квартал!", "Програма конфліктує з моїм антивірусом.",
        "Дуже складне налаштування, без 100 грам не розберешся.", "Ваші інструкції застарілі, нічого не збігається.",
        "Служба підтримки просто кидає слухавку, жах.", "Програма жере забагато пам'яті, все тормозить.",
        "Немає синхронізації з банком, хоча обіцяли.", "Чому ліцензія злетіла посеред робочого дня?",
        "Код активації не підходить, що робити?", "Програма автоматично не оновлюється, треба все вручну.",
        "Дуже незручне меню в новій версії BAS.", "Ваша база даних постійно видає 'пошкоджено файл'.",
        "Техпідтримка не знає відповіді на елементарні питання.", "Я хочу повернути кошти за це непрацюючe ПЗ.",

        # НЕЙТРАЛЬНО / ЗАПИТИ (Label 2) - 40 прикладів
        "Підкажіть вартість ліцензії на наступний рік?", "Хочу замовити модуль для обліку кадрів.",
        "Як налаштувати вивантаження з BAS в Excel?", "Хочу отримати прайс-лист на нові ПРРО.",
        "Яка процедура оновлення ключів ЕЦП у вашому офісі?", "Надішліть рахунок на подовження супроводу.",
        "Чи працює ваша програма на Windows 11?", "Де знайти інструкцію по роботі з новим модулем?",
        "Який графік роботи відділу консультацій на свята?", "Чи є навчання для нових користувачів BAS?",
        "Підкажіть номер телефону вашого бухгалтера.", "Мені потрібна консультація щодо переходу в хмару.",
        "Коли вийде наступне планове оновлення M.E.Doc?", "Які документи потрібні для укладання договору?",
        "Чи підтримує система роботу з декількома ФОП?", "Хочу змінити тарифний план на обслуговування.",
        "Який обсяг пам'яті потрібен для встановлення?", "Чи є можливість інтеграції з моїм сайтом?",
        "Підкажіть адресу вашого офісу в Тернополі.", "Як перевірити термін дії поточної ліцензії?",
        "Які системні вимоги для стабільної роботи ПРРО?", "Чи можна встановити BAS на декілька комп'ютерів?",
        "Як змінити пароль адміністратора в системі?", "Чи надаєте ви послуги з індивідуальної розробки?",
        "Де можна скачати останній патч для M.E.Doc?", "Як вивантажити податкові накладні в XML?",
        "Чи працює ваша підтримка у вечірній час?", "Який термін виготовлення ключів для печатки?",
        "Як додати нового користувача в базу даних?", "Чи є у вас демонстраційна версія програми?",
        "Які модулі входять в стандартний пакет обслуговування?", "Чи можна налаштувати автоматичне копіювання бази?",
        "Підкажіть, як змінити реквізити підприємства в звіті.", "Як підключити торговий еквайринг до BAS?",
        "Яка вартість виклику спеціаліста в офіс?", "Чи підтримує софт роботу з електронними чеками?",
        "Як оновити класифікатор професій в програмі?", "Чи є мобільний додаток для вашої CRM-системи?",
        "Як переглянути історію змін у документі?", "Хочу уточнити статус моєї поточної заявки."
    ],
    'label': [1]*40 + [0]*40 + [2]*40
}
```

```python
# 2. ТРИРІВНЕВЕ РОЗДІЛЕННЯ ДАНИХ (Train: 70%, Val: 15%, Test: 15%)
# Спочатку відокремлюємо 70% для навчання
train_texts, temp_texts, train_labels, temp_labels = train_test_split(
    data['text'], data['label'], test_size=0.3, random_state=42, stratify=data['label']
)

# Решту 30% ділимо навпіл на валідацію та тест
val_texts, test_texts, val_labels, test_labels = train_test_split(
    temp_texts, temp_labels, test_size=0.5, random_state=42, stratify=temp_labels
)

class SoftSvitDataset(Dataset):
    def __init__(self, texts, labels, tokenizer, max_len=64):
        self.texts = texts
        self.labels = labels
        self.tokenizer = tokenizer
        self.max_len = max_len

    def __len__(self):
        return len(self.texts)

    def __getitem__(self, item):
        encoding = self.tokenizer(
            self.texts[item],
            add_special_tokens=True,
            max_length=self.max_len,
            padding='max_length',
            truncation=True,
            return_tensors='pt'
        )
        return {
            'input_ids': encoding['input_ids'].flatten(),
            'attention_mask': encoding['attention_mask'].flatten(),
            'labels': torch.tensor(self.labels[item], dtype=torch.long)
        }
```

```python
# Ініціалізація моделі
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model_name = 'bert-base-multilingual-uncased'
tokenizer = BertTokenizer.from_pretrained(model_name)
model = BertForSequenceClassification.from_pretrained(model_name, num_labels=3).to(device)

train_loader = DataLoader(SoftSvitDataset(train_texts, train_labels, tokenizer), batch_size=4, shuffle=True)
val_loader = DataLoader(SoftSvitDataset(val_texts, val_labels, tokenizer), batch_size=4)
test_loader = DataLoader(SoftSvitDataset(test_texts, test_labels, tokenizer), batch_size=4)

# 3. ЦИКЛ НАВЧАННЯ
optimizer = AdamW(model.parameters(), lr=2e-5)

print(f"Запуск навчання на {device}...")
for epoch in range(10):
    model.train()
    total_loss = 0
    for batch in tqdm(train_loader, desc=f"Епоха {epoch+1}"):
        optimizer.zero_grad()
        input_ids = batch['input_ids'].to(device)
        attention_mask = batch['attention_mask'].to(device)
        labels = batch['labels'].to(device)

        outputs = model(input_ids, attention_mask=attention_mask, labels=labels)
        loss = outputs.loss
        total_loss += loss.item()
        loss.backward()
        optimizer.step()

    # Проміжна валідація
    model.eval()
    val_preds, val_true = [], []
    with torch.no_grad():
        for batch in val_loader:
            outputs = model(batch['input_ids'].to(device), attention_mask=batch['attention_mask'].to(device))
            preds = torch.argmax(outputs.logits, dim=1).cpu().numpy()
            val_preds.extend(preds)
            val_true.extend(batch['labels'].numpy())

    val_acc = accuracy_score(val_true, val_preds)
    print(f"Loss: {total_loss/len(train_loader):.4f} | Val Accuracy: {val_acc*100:.2f}%")
```

```python
# 4. ФІНАЛЬНЕ ТЕСТУВАННЯ (Test Set)
print("\n--- ФІНАЛЬНИЙ ТЕСТ НА НЕЗАЛЕЖНИХ ДАНИХ ---")
model.eval()
test_preds, test_true = [], []
with torch.no_grad():
    for batch in test_loader:
        outputs = model(batch['input_ids'].to(device), attention_mask=batch['attention_mask'].to(device))
        preds = torch.argmax(outputs.logits, dim=1).cpu().numpy()
        test_preds.extend(preds)
        test_true.extend(batch['labels'].numpy())

print(f"Фінальна точність (Test Accuracy): {accuracy_score(test_true, test_preds)*100:.2f}%")
print("\nДетальний звіт:")
print(classification_report(test_true, test_preds, target_names=['Негатив', 'Позитив', 'Нейтрально']))

# Збереження
model.save_pretrained("softsvit_bert_model")
tokenizer.save_pretrained("softsvit_bert_model")
print("Модель успішно навчена та збережена!")
```

Додаток Б

Копія публікації

Міністерство освіти і науки України
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

Матеріали

І Міжнародної науково-практичної конференції студентів і молодих вчених

«ICSPM 2026: Intelligent Computing Systems and Project Management / Інтелектуальні обчислювальні системи та управління проєктами»

21–22 травня 2026 р.

м. Тернопіль - 2026

УДК 004.8:005.8](063)

Присвячено 60-річчю Західноукраїнського національного університету

ICSPM 2026: Інтелектуальні обчислювальні системи та управління проєктами : збірник тез доповідей І Міжнародної науково-практичної конференції студентів і молодих вчених (Тернопіль, 21–22 травня 2026 року). – Тернопіль : ЗУНУ, 2026. – 190 с.

До збірника увійшли тези доповідей учасників І Міжнародної науково-практичної конференції студентів і молодих вчених «ICSPM 2026: Intelligent Computing Systems and Project Management / Інтелектуальні обчислювальні системи та управління проєктами», що відбулася на базі кафедри інформаційно-обчислювальних систем і управління факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету.

Матеріали відображають результати досліджень за такими напрямами:

1. Інтелектуальні обчислення та програмні системи.
2. Проєктно-орієнтоване управління та процеси прийняття рішень.
3. Штучний інтелект, аналітика даних і кібернетика.
4. Прикладні технології та інформаційно-комунікаційні системи.
5. Сучасні інформаційні технології в економіці, праві та освіті.

Рекомендовано до друку на засіданні кафедри інформаційно-обчислювальних систем і управління Західноукраїнського національного університету (протокол №12 від 26.05.2026 р.).

За зміст наукових праць, точність наведених цитувань, перекладу та достовірність фактологічних матеріалів відповідальність несуть автори публікацій. У збірнику збережено авторську стилістику, пунктуацію та орфографію.

УДК 004.89:004.912

ЗАСТОСУНОК ДЛЯ ВИЯВЛЕННЯ ЕМОЦІЙНОГО ТОНУ ТЕКСТУ ПОВІДОМЛЕНЬ НА ОСНОВІ ТРАНСФОРМЕРНИХ МОДЕЛЕЙ

Дмитерко Віктор, студент,
vityadm@gmail.com
Домбровський Михайло,
к.т.н., доцент кафедри інформаційно-обчислювальних систем та управління,
Західноукраїнський національний університет
m.dombrovskyi@wunu.edu.ua,
ORCID: 0000-0002-5582-5793

Стрімке зростання обсягів текстових даних у CRM-системах обумовлює необхідність впровадження інтелектуальних методів автоматизованого аналізу користувацьких звернень. Для підприємств, що здійснюють супровід програмного забезпечення, зокрема BAS та M.E.Doc [1, 2], оперативне виявлення негативних повідомлень є важливим фактором забезпечення якості технічної підтримки та підвищення рівня задоволеності клієнтів.

У CRM-системах менеджери підтримки щоденно опрацьовують значну кількість текстових повідомлень різного ступеня критичності. Ручний аналіз звернень характеризується високою трудомісткістю, часовими витратами та залежністю від суб'єктивного сприйняття оператора. Зростання кількості звернень підвищує когнітивне навантаження на персонал і ускладнює своєчасне виявлення критичних повідомлень [3–6]. Традиційні статистичні та словниково-орієнтовані методи аналізу тональності демонструють обмежену ефективність під час обробки українськомовних технічних текстів через складність контекстних залежностей, морфологічну варіативність та наявність професійної термінології [6, 8].

Одним із найбільш ефективних напрямів аналізу природної мови є використання трансформерних архітектур глибокого навчання [3-5]. Модель BERT [3] забезпечує формування контекстно-залежних векторних представлень тексту та врахування семантичних зв'язків між словами незалежно від їхнього розташування у реченні. На відміну від рекурентних нейронних мереж, трансформерна архітектура реалізує паралельне опрацювання послідовностей за допомогою механізму багатоголової уваги (Multi-Head Self-Attention) [4]. Аналіз тональності текстів є одним із ключових напрямів розвитку інтелектуальних інформаційних систем [8], а перспективність інтеграції NLP-технологій підтверджується дослідженнями у сфері інтелектуальної взаємодії людини та комп'ютера [5].

Програмний застосунок реалізовано на основі клієнт-серверної архітектури із використанням мови Python та фреймворку Streamlit [7]. Архітектура системи забезпечує розділення рівнів взаємодії користувача, попередньої обробки тексту та інференсу трансформерної моделі. Для оптимізації використання ресурсів

застосовано механізм кешування за допомогою декоратора @st.cache_resource, що дозволяє уникнути повторного завантаження ваг моделі BERT і підвищити швидкодію застосунку [3, 7].

Процес функціонування застосунку реалізовано у вигляді NLP-конвеєра, наведеного на рисунку 1. Конвеєр включає рівні попередньої обробки тексту, токенізації та інференсу трансформерної моделі. На етапі попередньої обробки здійснюється очищення повідомлень від службових символів і технічних артефактів CRM-системи. На етапі токенізації текст сегментується на субтокени за алгоритмом WordPiece, що забезпечує коректну обробку професійної термінології та мінімізує проблему невідомих лексем [3]. На рівні інференсу формується контекстуальне представлення тексту за допомогою 12-рівневої архітектури Transformer [4], після чого виконується класифікація емоційного тону повідомлення.

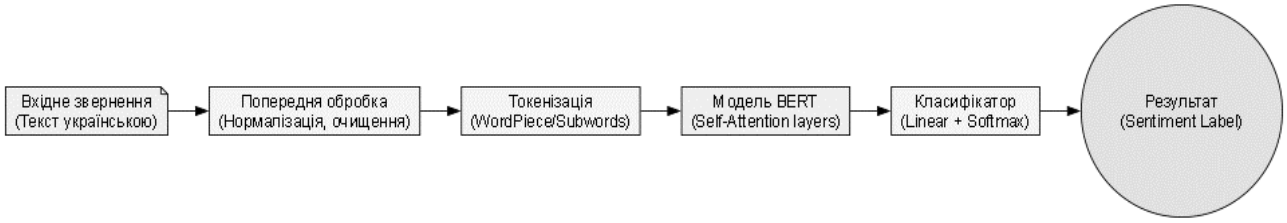

Рисунок 1 – Конвеєр NLP функціонування застосунку

В основі інтелектуального модуля використано модель bert-base-multilingual-uncased, адаптовану до специфіки українськомовних технічних звернень шляхом процедури донавчання (fine-tuning) [3]. Для навчання сформовано спеціалізований датасет на основі звернень клієнтів щодо програмних продуктів BAS, M.E.Doc та суміжних інформаційних сервісів. Повідомлення розподілено на три класи: негативні, позитивні та нейтральні.

Для оптимізації вагових коефіцієнтів трансформерної моделі під час донавчання застосовано функцію втрат перехресної ентропії (Cross-Entropy Loss), яка мінімізує розбіжність між прогнозованими та фактичними класами повідомлень [3, 4]:

$$L = -\sum_{i=1}^{n} y_i \log(\widehat{y_i})$$

Де y_i — істинна мітка класу класифікованих повідомлень, $\widehat{y_i}$ — прогнозована ймовірність класифікованих повідомлень.

Використання механізму Self-Attention дозволило моделі ефективно фокусуватися на ключових маркерах емоцій незалежно від їхнього розташування у реченні [4]. Це забезпечило можливість коректного аналізу технічних звернень зі складною структурою, професійною термінологією та контекстно-залежними конструкціями.

Експериментальна перевірка системи проводилася на тестовій вибірці повідомлень CRM-системи підприємства. Для оцінювання ефективності

класифікації використано метрики Precision, Recall та F1-score. Результати оцінювання наведено у таблиці 1.

Таблиця 1 – Метрики оцінювання класифікаційної моделі BERT

Клас (Емоційний тон)	Precision	Recall	F1-score
Негативний (label 0)	0,75	1,00	0,86
Позитивний (label 1)	1,00	0,67	0,80
Нейтральний (label 2)	1,00	1,00	1,00
Загальна точність (Accuracy)	—	0,88	—

Експериментальна перевірка продемонструвала точність класифікації 88,89 %. Найбільш важливим показником для CRM-систем є значення Recall для негативного класу, яке досягло 1,00. Це свідчить про відсутність пропуску критичних звернень і забезпечує своєчасне реагування менеджерів підтримки на проблемні повідомлення користувачів. Високі значення Precision для нейтрального та позитивного класів підтверджують здатність моделі мінімізувати кількість помилкових класифікацій. Приклад інтерфейсу завантаження даних для перевірки працездатності моделі наведено на рисунку 2. У вікні застосунку відображаються повідомлення користувачів та результати автоматичної класифікації емоційного тону.

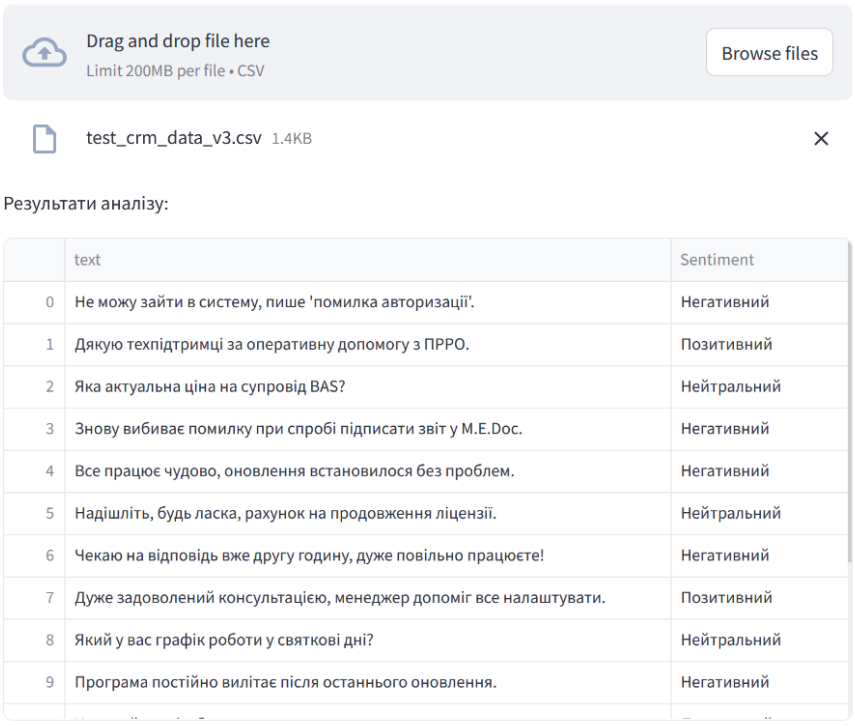

Рисунок 2 – Завантажених даних для перевірки працездатності моделі

Для підвищення ефективності аналізу результатів у застосунку реалізовано модуль інтерактивної візуалізації. Стовпчикова діаграма, наведена на рисунку 3, відображає кількісний розподіл повідомлень за емоційними класами та дозволяє оперативно оцінювати домінуючу тональність звернень.

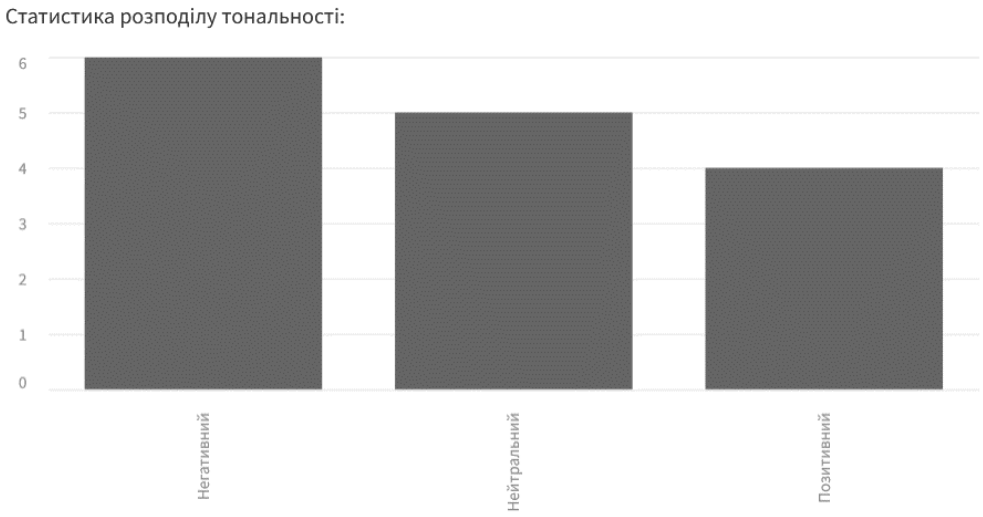

Рисунок 3 –Діаграма розподілу емоційної тональності

Кругова діаграма співвідношення емоційних категорій, наведена на рисунку 4, забезпечує узагальнене представлення емоційного фону текстового масиву повідомлень CRM-системи.

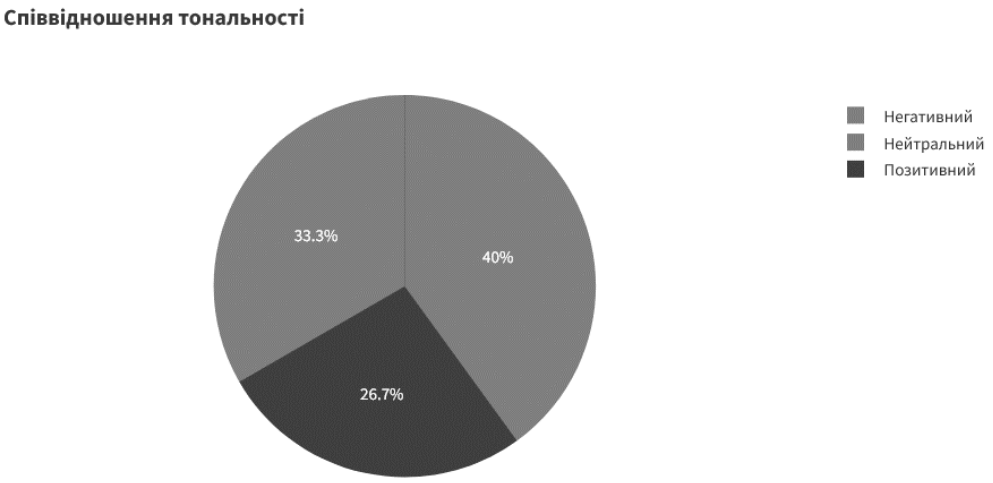

Рисунок 4 –Діаграма співвідношення емоційної тональності

Запропонований підхід передбачає удосконалення методики донавчання мультимовних трансформерних моделей для аналізу українськомовного технічного контенту в CRM-системах IT-підприємств. На відміну від універсальних моделей аналізу тональності, розроблений застосунок враховує специфіку професійної лексики, технічних скорочень і морфологічну складність спеціалізованих звернень користувачів. Це дозволило підвищити ефективність класифікації та забезпечити високу повноту виявлення негативних повідомлень навіть за умов обмеженої навчальної вибірки завдяки використанню механізмів Transfer Learning. У підсумку створено автономний програмний модуль, інтегрований у CRM-системи через стандартизовані формати обміну даними. Використання застосунку автоматизує пріоритезацію звернень клієнтів,

мінімізує ризик пропуску критичних повідомлень, підвищує оперативність роботи служби підтримки та знижує когнітивне навантаження на персонал завдяки інтерактивній візуалізації результатів аналізу.

У результаті дослідження розроблено та практично реалізовано інтелектуальний програмний застосунок для автоматизації аналізу емоційного тону повідомлень у CRM-системах підтримки клієнтів. Використання трансформерної архітектури BERT у поєднанні з методикою донавчання на спеціалізованому корпусі українськомовних технічних звернень забезпечило ефективне контекстне розуміння текстових повідомлень і високу якість класифікації.

Експериментально підтверджено ефективність запропонованого підходу: досягнуто точності класифікації 88,89 % та показника повноти виявлення негативних звернень на рівні 1,00. Практичне впровадження застосунку забезпечує автоматизацію пріоритезації звернень у CRM-системах та підвищує ефективність роботи служб технічної підтримки.

Список використаних джерел

1. Офіційний сайт програмних продуктів BAS [Електронний ресурс]. – Режим доступу: https://www.bas-soft.eu/ (дата звернення: 07.05.2026).
2. Офіційний сайт системи електронного документообігу M.E.Doc [Електронний ресурс]. – Режим доступу: https://medoc.ua/ (дата звернення: 07.05.2026).
3. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of NAACL-HLT. – Minneapolis, USA, 2019. – P. 4171–4186.
4. Vaswani A., Shazeer N., Parmar N. et al. Attention Is All You Need // Advances in Neural Information Processing Systems. – 2017. – Vol. 30. – P. 5998–6008.
5. Pashchuk I., Turchenko I., Dombrovskyi M., Hrushytskyi M. AI-Driven Natural Language Processing to SQL Query Generation for Enhanced Human-Machine Interaction in the Digital Era // 2025 IEEE 13th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). – 2025. – P. 1–6.
6. Субботін С. О., Олійник А. О. Інтелектуальні інформаційні технології аналізу текстових даних // Радіоелектроніка, інформатика, управління. – 2022. – № 3. – С. 97–108.
7. Wolf T., Debut L., Sanh V. et al. Transformers: State-of-the-Art Natural Language Processing // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. – 2020. – P. 38–45.
8. Коваленко А. О., Шевченко І. В. Методи аналізу тональності текстової інформації в інформаційних системах // Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. – 2023. – № 2. – С. 45–52.

ІНТЕЛЕКТУАЛЬНІ ОБЧИСЛЮВАЛЬНІ СИСТЕМИ ТА УПРАВЛІННЯ ПРОЄКТАМИ

МАТЕРІАЛИ
І МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ
СТУДЕНТІВ І МОЛОДИХ ВЧЕНИХ

INTELLIGENT COMPUTING SYSTEMS AND PROJECT MANAGEMENT

CONFERENCE PROCEEDINGS OF THE I INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE OF STUDENTS AND YOUNG SCIENTISTS

Тернопіль, 21–22 травня 2026 року / Ternopil, May 21–22, 2026

Підписано до друку 20.05.2026.
Формат 60x84/16. Гарнітура Times New Roman.
Папір офсетний 80 г/м². Друк електрографічний.
Умов.-друк. арк. 9,12. Обл.-вид. арк. 10,03.
Тираж 300 примірників. Замовлення №26-515.

Віддруковано ФОП Шпак В.Б.
Свідоцтво про внесення суб'єкта видавничої справи до
Державного реєстру видавців, виготовлювачів і розповсюджувачів
видавничої продукції серія ДК№7599 від 10.02.2022
Тел.: 097 299 38 99
E-mail: tooums@ukr.net