Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

NLP-пайплайн для обробки новинних потоків із генерацією аудіо-контенту

(тема)

Виконав:
здобувач другого року навчання,
групи СШІм-24-2
Савелій НЕЄЛОВ
(власне ім'я, прізвище)
Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)
Керівник проф. Ваган ТЕРЗІЯН
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Лариса ЧАЛА
(підпис) (власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Неєлову Савелію Костянтиновичу
(прізвище, ім'я, по батькові)

1. Тема роботи NLP-пайплайн для обробки новинних потоків із генерацією аудіо-контенту

затверджена наказом університету від 6 квітня 20 26 р. № 269Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 21 травня 20 26 р.

3. Вихідні дані до роботи Відомості щодо існуючих систем медіа-моніторингу та новинних агрегаторів, текстові дані та історія повідомлень з відкритих Telegram-каналів, трансформерні моделі машинного навчання для обробки природної мови, реляційна база даних, консольний клієнтський додаток та дашборди у середовищі Power BI.

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної галузі
2) Постановка задачі дослідження
3) Проектування та програмна реалізація системи

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	06.04.2026	виконано
2	Аналіз предметної галузі	06.04.2026 – 08.04.2026	виконано
3	Постановка задачі дослідження	08.04.2026 – 09.04.2026	виконано
4	Розробка архітектури програмного забезпечення	09.04.2026 – 18.04.2026	виконано
5	Оформлення пояснювальної записки	18.04.2026 – 20.04.2026	виконано
6	Надання роботи на відгук та рецензування	25.04.2026	виконано
7	Перевірка пояснювальної записки на плагіат	03.05.2026	виконано
8	Захист перед ЕК	21.05.2026	

Дата видачі завдання 6 квітня 20 26 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Ваган ТЕРЗІЯН
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 66 с., 28 рис., 1 дод., 17 джерел.

АНАЛІЗ НОВИН, ВЕЛИКІ МОВНІ МОДЕЛІ, ВІЗУАЛІЗАЦІЯ ДАНИХ, ОБРОБКА ПРИРОДНОЇ МОВИ, СИНТЕЗ МОВЛЕННЯ, BI-СИСТЕМИ.

Об’єкт дослідження – процес автоматизованого збору, інтелектуальної обробки та візуалізації неструктурованих текстових даних із новинних джерел.

Предмет дослідження – методи обробки природної мови, моделі машинного навчання для сумаризації та синтезу мовлення, а також алгоритми нормалізації даних для побудови аналітичних дашбордів.

Мета роботи – розробка комплексної автоматизованої системи для парсингу Telegram-каналів, глибокого NLP-аналізу новинних потоків, генерації аудіо-дайджестів та підготовки структурованих баз даних для інтерактивних систем бізнес-аналітики.

Методи дослідження – вивчення методів обробки природної мови, включаючи NER, Zero-Shot Classification, тестування архітектур великих мовних моделей (LLM) та систем синтезу мовлення (TTS), аналіз документації до бібліотек обраних мов програмування (Python, C#), застосування методів проектування баз даних та налаштування BI-систем у середовищі Power BI.

У процесі дослідження були використані наступні технології: Python, FastAPI, C#, Entity Framework, mitmproxy, Microsoft Power BI, а також нейромережеві моделі mDeBERTa, GLiNER, Qwen та Silero. В рамках роботи був інтегрований комплексний NLP-пайплайн для семантичного аналізу новин у мікросервісний застосунок із функціоналом генерації аудіо-дайджестів, створення та візуалізації аналітичних звітів.

ABSTRACT

Master’s thesis contains: 66 pp., 28 fig., 1 ann., 17 references.

BI SYSTEMS, DATA VISUALIZATION, LARGE LANGUAGE MODELS, NATURAL LANGUAGE PROCESSING, NEWS ANALYSIS, SPEECH SYNTHESIS.

The object of study is the process of automated collection, intelligent processing, and visualization of unstructured text data from news sources.

The subject of study includes natural language processing methods, machine learning models for summarization and speech synthesis, as well as data normalization algorithms for building analytical dashboards.

The aim of the work is the development of a comprehensive automated system for parsing Telegram channels, deep NLP analysis of news streams, generation of audio digests, and preparation of structured databases for interactive business intelligence systems.

Research methods include the study of natural language processing methods including NER, Zero-Shot Classification, testing the architectures of large language models (LLMs) and text-to-speech (TTS) systems, analyzing the documentation of libraries for the chosen programming languages (Python, C#), applying database design methods, and configuring BI systems (Power BI).

During the research, the following technologies were used: Python, FastAPI, C#, Entity Framework, mitmproxy, Microsoft Power BI, as well as the neural network models mDeBERTa, GLiNER, Qwen, and Silero. As part of the project, a comprehensive NLP pipeline for semantic news analysis was integrated into a microservice application featuring functionality for generating audio digests, as well as creating and visualizing analytical reports.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

BI – Business Intelligence – бізнес-аналітика, програмне забезпечення та методи для перетворення вхідних даних на значущу інформацію та інтерактивні дашборди (наприклад, Microsoft Power BI);

FastAPI – сучасний високопродуктивний веб-фреймворк мовою Python для створення серверних застосунків та API;

LLM – Large Language Model – велика мовна модель, тип нейромережі з мільярдами параметрів, здатний генерувати, сумаризувати та розуміти тексти природною мовою;

mitmproxy – програмний інструмент для перехоплення та аналізу HTTP/HTTPS-трафіку, використаний для реверс-інжинірингу мережевих запитів;

NER – Named Entity Recognition – розпізнавання іменованих сутностей, метод вилучення з тексту власних назв, геолокацій, організацій та імен;

NLP – Natural Language Processing – обробка природної мови, напрямок штучного інтелекту, що вивчає методи взаємодії між комп'ютерами та людською мовою;

TTS – Text-to-Speech – синтез мовлення, технологія штучної генерації людського голосу з тексту;

Zero-Shot Classification – парадигма машинного навчання, що дозволяє моделі класифікувати текстові дані за категоріями, яких вона не бачила під час етапу тренування.

ВСТУП

Протягом останнього десятиліття демонструється стрімке зростання обсягів цифрової інформації, що є наслідком масової цифровізації суспільства та глобального переходу медіа-ресурсів на децентралізовані цифрові платформи. Сучасний комунікаційний простір відіграє роль не лише інструменту для спілкування, але й повноцінного інформаційного хабу з безмежними можливостями для миттєвого поширення новин, суспільно-політичної аналітики та різноманітного медіаконтенту.

Оперативність доставки інформації та незалежність від традиційних форматів засобів масової інформації привертають увагу як індивідуальних споживачів контенту, так і корпоративних аналітиків, дослідників та бізнес-структур. Підвищення інтересу до автоматизованого моніторингу інформаційного простору пов’язано з необхідністю швидкого реагування на події в економічній, політичній та безпековій сферах. Проте стан постійного розвитку, в якому перебуває цифровий медіа-простір, сприяє появі тисяч нових джерел інформації та, як наслідок, призводить до критичного інформаційного перенавантаження і безперервного накопичення масивів неструктурованих текстових даних.

Велика кількість та розрізненість існуючих алгоритмів обробки природної мови відокремила необхідність у визначенні та створенні вичерпних методів та інструментів у вигляді єдиної наскрізної системи. Розробка комплексного рішення, що об'єднує інтелектуальний аналіз, машинну сумаризацію, трансформацію форматів та візуалізацію даних, є необхідною для максимізації ефективності в прийнятті аналітичних рішень, враховуючи як загальні інформаційні тенденції, так і індивідуальні потреби користувача у якісному та структурованому контенті.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Огляд методів обробки природної мови (NLP) для аналізу новинних потоків

1.1.1 Загальна проблематика обробки неструктурованих текстових даних

Традиційні новинні веб-ресурси та класичні інформаційні агрегатори поступово втрачають монополію на оперативну доставку контенту. Їхнє місце стрімко займають платформи соціальних медіа та месенджери, серед яких безперечним лідером є Telegram. На відміну від звичайних новинних порталів, які генерують узагальнений та стандартизований контент, Telegram функціонує як універсальне середовище, що об'єднує сотні тисяч вузькоспеціалізованих каналів. Така децентралізована архітектура дозволяє цілеспрямовано екстрагувати інформацію за специфічними критеріями та тематиками: від миттєвих сповіщень про повітряні тривоги чи перебіг бойових дій до глибокої криптоаналітики та фінансових інсайтів. Саме ця здатність до детальної сегментації робить Telegram найбільш релевантним джерелом для глибокого моніторингу конкретних інформаційних ніш.

Проте, зазначена перевага породжує серйозну технологічну проблему. Інформаційний простір Telegram характеризується безперервним генеруванням величезних обсягів текстової інформації, які здебільшого являють собою суцільний потік неструктурованих даних. У такому вигляді інформація є придатною виключно для ручного сприйняття людиною, що робить її вкрай складною для швидкої машинної обробки, автоматичної агрегації та подальшого використання при побудові аналітичних дашбордів або географічних мап.

Для того, щоб інформаційна система могла ідентифікувати фактичну суть тексту та перетворити його на структурований набір даних, придатний

для обчислень, застосовуються методи обробки природної мови (Natural Language Processing, NLP). Використання NLP-методів дозволяє автоматизувати процес моніторингу новинних потоків, суттєво знизити когнітивне навантаження на кінцевого користувача та відфільтрувати значні обсяги інформаційного «шуму», залишаючи лише цільові дані.

У контексті розробки комплексного рішення для автоматизованого парсингу, аналізу та візуалізації новин виникає низка послідовних технологічних викликів, пов'язаних з обробкою неструктурованого тексту. Перш за все, існує потреба в динамічній агрегації та категоризації інформації. Оскільки інформаційне середовище месенджерів постійно змінюється, система повинна вміти гнучко та автоматично визначати тематику кожного новинного повідомлення, адаптуючись до нових рубрик без необхідності створення масивних розмічених баз даних чи ручного оновлення правил.

Наступним важливим етапом є глибоке структурування контексту. Із суцільного текстового потоку необхідно з високою точністю виявляти та вилучати ключові об'єкти, такі як імена конкретних осіб, географічні локації та назви організацій. Проте просте вилучення цих елементів є недостатнім для побудови якісних звітів. Для забезпечення коректної роботи засобів бізнес-аналітики (BI) та усунення семантичних дублікатів у базі даних, усі видобуті текстові сутності вимагають алгоритмічного зведення до єдиної стандартизованої словникової форми.

Зрештою, підготовлені та структуровані дані потребують подальшої трансформації для забезпечення зручного сприйняття кінцевим споживачем контенту. Невід'ємним кроком є здатність системи генерувати зв'язний текст на основі десятків розрізнених новинних повідомлень. Виникає необхідність у редукції загального обсягу вхідної інформації таким чином, щоб зберегти ключові факти та сформувати логічно вибудований, семантично цілісний дайджест. Створення такого адаптованого сценарію є

критично важливою передумовою для його подальшого автоматичного озвучення та перетворення на повноцінний аудіо-контент.

Вирішення описаного комплексу підзадач вимагає проєктування наскрізного конвеєра обробки даних із залученням сучасних інтелектуальних алгоритмів.

1.1.2 Розпізнавання іменованих сутностей на основі архітектури GLiNER

Розпізнавання іменованих сутностей є одним із фундаментальних методів обробки природної мови, який відповідає за автоматичне виявлення та класифікацію ключових інформаційних об'єктів у тексті. У своєму класичному та найбільш стандартизованому вигляді задача NER зводиться до сканування неструктурованого тексту та віднесення виділених слів або словосполучень (токенів) до однієї з попередньо визначених категорій із жорстко обмеженого набору (рисунок 1.1).

```
Вхідний текст:Віткофф заявив про значний прогрес після першого дня переговорів у Женеві між Україною США і Росією віримо ваші ставки чого чекати від цих зустрічей.
Віткофф => Особа (Впевненість: 0.94)
Женеві => Місто (Впевненість: 0.99)
Україною => Країна (Впевненість: 0.76)
США => Країна (Впевненість: 0.82)
Росією => Країна (Впевненість: 0.89)
```

Рисунок 1.1 – Приклад результатів NER-класифікації до вхідного тексту

У контексті розроблюваної інформаційної системи моніторингу новин застосування підходу з обмеженим набором сутностей є найбільш доцільним та архітектурно обґрунтованим. Оскільки кінцевою метою є структурування даних для їх збереження у спеціалізованій базі даних та подальшої візуалізації у BI-системах, зокрема, на географічних мапах та діаграмах частотності, неконтрольоване видобування довільних або абстрактних сутностей призвело б до значного засмічення бази даних. Тому

для роботи конвеєра обробки даних було визначено строгу номенклатуру з чотирьох основних цільових класів: «Особа» (Person), «Місто» (City), «Країна» (Country) та «Компанія» (Company).

Процес розпізнавання відбувається на базі нейромережевих моделей архітектури Transformer, які дозволяють враховувати глибокий двонаправлений контекст кожного слова у реченні. Для вирішення цього завдання в системі використано підхід на базі моделі GLiNER (Generalist Model for Named Entity Recognition), яка завдяки двонаправленій трансформерній архітектурі здатна ефективно видобувати сутності за динамічно заданими промптами [1].

З математичної точки зору, згідно з офіційною специфікацією архітектури GLiNER, процес розпізнавання докорінно відрізняється від класичних NER-моделей. Замість вбудови класів у вихідний шар нейромережі, модель приймає назви цільових сутностей безпосередньо як частину вхідної послідовності.

Формат подання вхідного повідомлення I має вигляд конкатенації списку цільових класів та токенів повідомлення та формується наступним чином:

$$I = [c_0, \dots, c_K\ [SEP]\ x_1, \dots, x_N], \tag{1.1}$$

де c_K – токени цільових класів;

$[SEP]\ x_N$ – токени вхідного тексту повідомлення із попереднім токеном-розділювачем.

Після проходження цієї послідовності через приховані шари трансформера, алгоритм переходить до виділення текстових фрагментів, що також іменуються спанами. Замість класифікації кожного окремого слова, модель оцінює неперервні послідовності слів. Векторне подання конкретного спану S_{ij} який починається з токена i та закінчується токеном

j, формується шляхом об'єднання прихованих станів його меж, що визначається як:

$$S_{ij} = FFN(h_i \oplus h_j), \quad (1.2)$$

де h_i, h_j – приховані векторні стани (hidden states) початкового та кінцевого токенів виділеного текстового фрагмента, де максимальна різниця $i - j$ не більше ніж 12;

$\oplus$ – операція конкатенації векторів;

FFN (Feed-Forward Network) – повнозв'язна нейронна мережа прямого поширення, яка стискає об'єднаний вектор до необхідної розмірності для подальшого порівняння [1].

Фінальним етапом екстракції є обчислення функції відповідності $f(i, j, t)$, яка визначає ступінь належності виділеного спану S_{ij} до одного із заданих цільових класів c_k де $k \in \{1, \dots, K\}$. Ця функція реалізується як скалярний добуток між векторним поданням текстового фрагмента та векторним поданням самого класу v_k:

$$f(i, j, t) = S_{ij} * v_k. \quad (1.3)$$

Отримане логіт-значення пропускається через активаційну функцію Sigmoid, яка повертає кінцеву ймовірність. Якщо ймовірність перевищує встановлений поріг (threshold), система фіксує знаходження іменованої сутності (наприклад, фрагмент «Київ» розпізнається та маркується виключно як «Місто», а «США» – як «Країна»). Застосування такого стандартизованого, але водночас гнучкого підходу забезпечує високу точність екстракції цільових даних. Сформований моделлю масив типізованих об'єктів легко піддається алгоритмічному сортуванню, що гарантує коректне відображення географічних даних та персоналій у подальших аналітичних звітах та дашбордах.

1.1.3 Класифікація текстів без попереднього навчання на основі архітектури mDeBERTa

Традиційні підходи до машинного навчання у задачах текстової класифікації вимагають наявності значних обсягів попередньо розмічених навчальних даних для кожної окремої категорії. В умовах високодинамічного інформаційного середовища, яким є мережа новинних Telegram-каналів, тематика повідомлень може стрімко змінюватися, а формування нових масштабних датасетів для донавчання моделей є ресурсомістким і хронологічно неефективним процесом.

Для розв’язання цієї проблеми у розроблюваній системі застосовано парадигму класифікації текстів без попереднього навчання (Zero-Shot Classification). Цей підхід дозволяє динамічно розподіляти неструктуровані тексти за рубриками (наприклад, «Війна», «Політика», «Відключення світла») виключно на основі семантичного розуміння контексту, без необхідності оновлення ваг нейронної мережі під конкретний набір класів, як показано на рисунку 1.2.

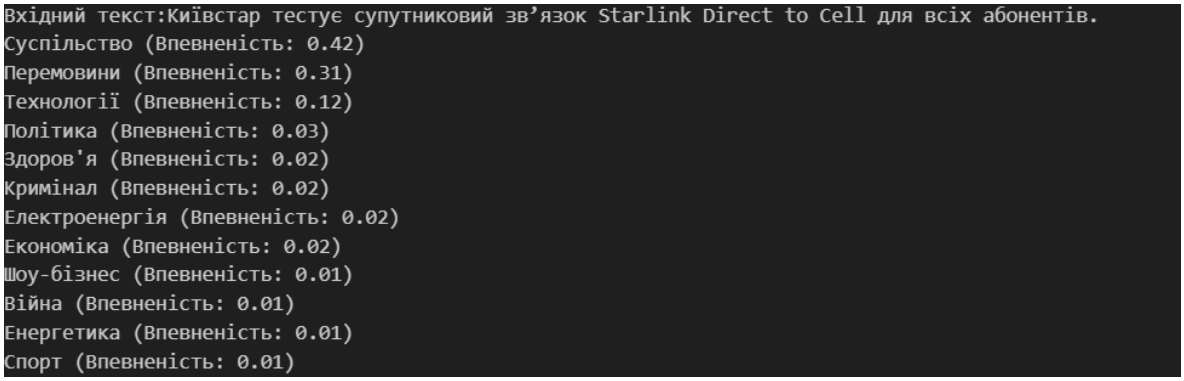


Рисунок 1.2 – Приклад результатів Zero-Shot Classification до вхідного тексту

В основі механізму семантичної класифікації текстів лежить підхід зведення задачі категоризації до задачі визначення логічного текстового

слідування (Natural Language Inference, NLI) [13]. Цей механізм кардинально змінює класичний алгоритм машинного навчання: замість прямого передбачення мітки класу, модель розглядає вхідний текст новинного повідомлення як «Передумову» (p – premise), а назву цільової рубрики певного класу c штучно інтегрує у текстовий шаблон, утворюючи «Гіпотезу» (h_c – hypothesis), наприклад: «Цей текст стосується рубрики c». Демонстрація такого розв'язання зображена на рисунку 1.3.

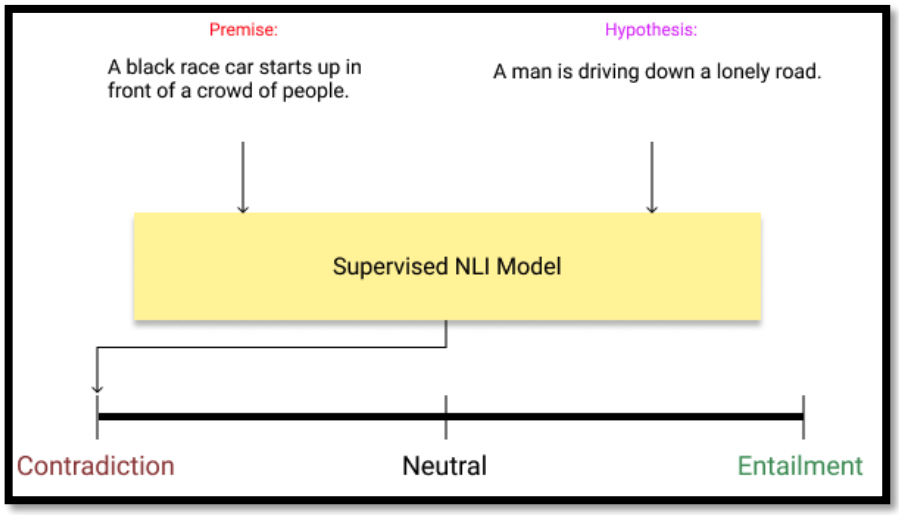


Рисунок 1.3 – Демонстрація NLI-розв'язання

Під час обробки запиту нейромережа обчислює логічну спорідненість між передумовою та набором гіпотез.

Формально, для множини цільових класів C, модель видає логіт-значення $s_{entail}(p, h_c)$, яке відображає ступінь того, наскільки гіпотеза логічно випливає з тексту (entailment).

Кінцева ймовірність $P(c|p)$ належності тексту p до рубрики c обчислюється шляхом нормалізації цих значень через функцію Softmax [13], яка набуває вигляду як у формулі 1.4.

$$P(c|p) = \frac{\exp(s_{entail}(p,h_c))}{\sum_{k=1}^{|C|} \exp\left(s_{entail}(p,h_k)\right)}. \quad (1.4)$$

Для виконання цих трудомістких обчислень у розробленій системі застосовано переднавчену багатомовну трансформерну архітектуру mDeBERTa-v3 [2]. Фундаментальною математичною відмінністю цієї моделі від класичного BERT є використання механізму розщепленої уваги, Disentangled Attention. Замість об'єднання вектора змісту слова та вектора його позиції в єдиний ембеддинг, mDeBERTa обробляє їх окремо. Матриця уваги A_{ij} між токеном i та токеном j обчислюється як сума трьох компонентів:

$$A_{i,j} = H_i H_j^T + H_i P_{(j|i)}^T + P_{(i|j)} H_j^T, \tag{1.5}$$

де H_i, H_j – приховані вектори контенту токенів;

$P_{(i|j)}, P_{(j|i)}$ – вектори їхніх сумісних позицій [2].

Здатність обраної моделі працювати з українською мовою та виконувати Zero-Shot класифікацію без додаткового тренування забезпечується специфікою її донавчання (fine-tuning). Модель була оптимізована на масивному крос-лінгвальному датасеті XNLI (Cross-lingual NLI) [17] за методологією перенесення знань. Фрагмент структури даного набору навчальних даних наведено на рисунку 1.4.

Language	Premise / Hypothesis	Genre	Label
English	You don't have to stay there. You can leave.	Face-To-Face	Entailment
French	La figure 4 montre la courbe d'offre des services de partage de travaux. Les services de partage de travaux ont une offre variable.	Government	Entailment
Spanish	Y se estremeció con el recuerdo. El pensamiento sobre el acontecimiento hizo su estremecimiento.	Fiction	Entailment
German	Während der Depression war es die ärmste Gegend, kurz vor dem Hungertod. Die Weltwirtschaftskrise dauerte mehr als zehn Jahre an.	Travel	Neutral
Swahili	Ni silaha ya plastiki ya moja kwa moja inayopiga risasi. Inadumu zaidi kuliko silaha ya chuma.	Telephone	Neutral
Russian	И мы занимаемся этим уже на протяжении 85 лет. Мы только начали этим заниматься.	Letters	Contradiction
Chinese	让我告诉你，美国人最终如何看待你作为独立顾问的表现。 美国人完全不知道您是独立律师。	Slate	Contradiction
Arabic	تحتاج الوكالات لأن تكون قادرة على قياس مستويات النجاح. لا يمكن للوكالات أن تعرف ما إذا كانت ناجحة أم لا	Nine-Eleven	Contradiction

Рисунок 1.4 – Приклади пар «передумова-гіпотеза» з крос-лінгвального датасету XNLI [17]

Як видно з рисунку 1.4, на етапі тренування нейромережа опрацьовує пари текстових послідовностей (Premise / Hypothesis) різними мовами у поєднанні з відповідними мітками логічного зв'язку (Entailment – слідування, Neutral – нейтральність, Contradiction – протиріччя). Завдяки обробці сотень тисяч таких пар 15-ма різними мовами, модель формує універсальний багатомовний семантичний простір.

Механізм формування цього простору базується на процесі міжмовного перенесення знань (cross-lingual transfer learning) шляхом вирівнювання реченнєвих ембеддингів (sentence embedding alignment). Графічну ілюстрацію цієї архітектурної адаптації наведено на рисунку 1.5.

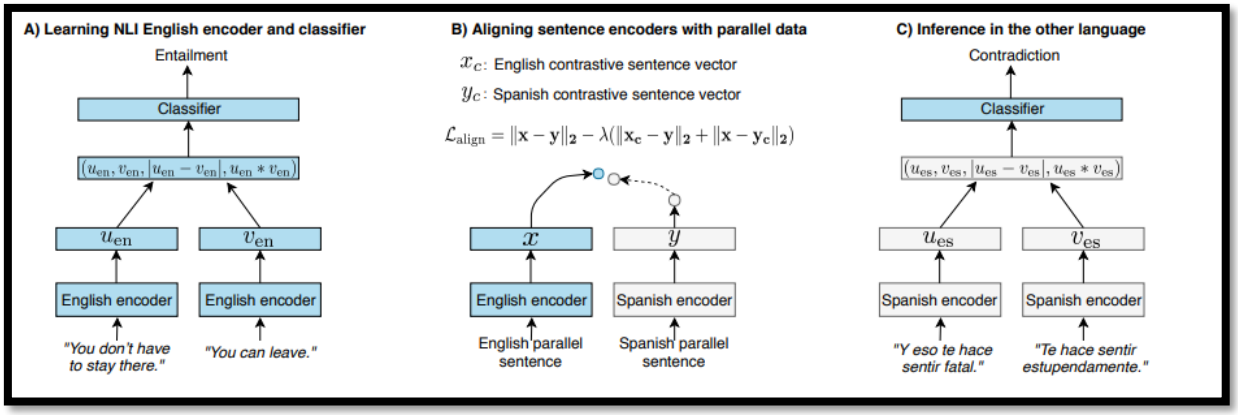


Рисунок 1.5 – Ілюстрація мовної адаптації шляхом вирівнювання векторних подань речень [17]

Як показано на схемі, процес адаптації складається з трьох ключових етапів. На першому етапі (А) базовий кодувальник (encoder) та класифікатор проходять повноцінне навчання на великому масиві англомовних NLI-даних, формуючи первинне семантичне ядро.

Ключовим є етап (Б), на якому відбувається математичне вирівнювання векторних просторів. Використовуючи паралельні дані, а саме ті, що ідентичні за змістом речення англійською та цільовою мовами, алгоритм тренує іншомовний кодувальник таким чином, щоб він генерував

вектори, максимально наближені до векторів англомовного кодувальника. Спеціальна функція втрат $\mathcal{L}_{align}$ мінімізує евклідову відстань між цими ембеддингами. Таким чином, українські та англійські слова з однаковим значенням починають картографуватися в одні й ті самі координати єдиного семантичного простору.

В результаті, на етапі логічного виведення (B), україномовний текст, складовими якого є новина та рубрика, проходить через адаптований кодувальник і перетворюється на вектори, які оригінальний класифікатор здатен обробити так само ефективно, як і англійські. Саме ця архітектурна особливість дозволяє системі з високою точністю обчислювати ймовірність $P(c|p)$ для україномовних новин, навіть якщо конкретні назви цільових рубрик ніколи не зустрічалися їй у тренувальному корпусі.

Окрім мовної універсальності, підхід на базі логічного виведення забезпечує багатовимірну класифікацію текстів. Замість жорсткої прив'язки повідомлення до одного заздалегідь визначеного дерева класів, система здатна аналізувати вхідний текст під різними семантичними кутами одночасно, використовуючи різні набори гіпотез. Демонстрацію такої багатоаспектної класифікації наведено на рисунку 1.5.

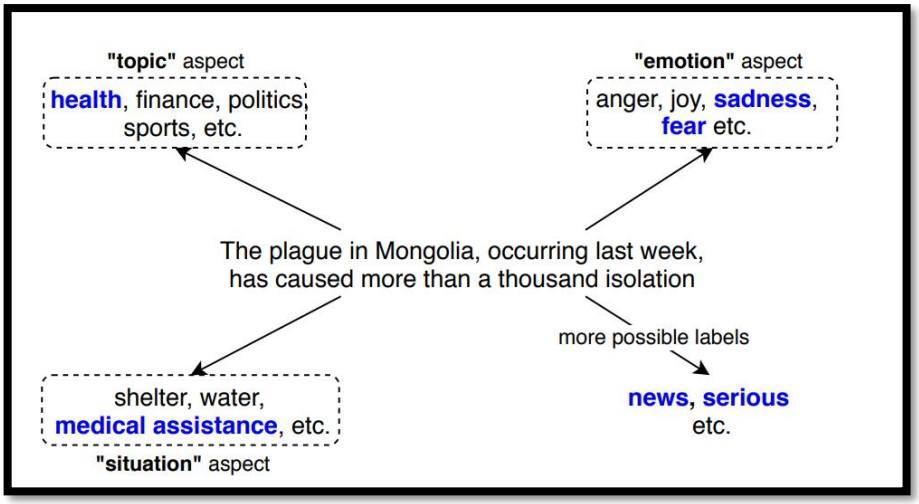


Рисунок 1.6 – Приклад багатоаспектної Zero-Shot класифікації вхідного тексту [13]

Як видно з рисунку 1.6, одне й те саме вхідне повідомлення може бути паралельно класифіковане за загальною тематикою (topic: здоров'я, політика), за емоційним забарвленням (emotion: радість, страх, сум), або за ситуативним контекстом (situation: потреба у медичній допомозі, укриття тощо). Модель гнучко реагує на зміну тексту цільової мітки.

Застосування методу Zero-Shot Classification у конвеєрі обробки даних забезпечує виняткову архітектурну гнучкість розробленої системи. Вона дозволяє розробнику або кінцевому користувачу миттєво додавати нові аналітичні категорії чи аспекти без переписування програмного коду та без ресурсомісткого перенавчання моделі. Крім того, категоризовані у такий спосіб дані формують надійний базис для роботи BI-систем, дозволяючи реалізовувати наскрізну крос-фільтрацію аналітичних дашбордів за обраними тематичними рубриками.

1.1.4 Нормалізація та лемматизація видобутих текстових сутностей

Після розпізнавання та екстракції цільових іменованих сутностей, зокрема, географічних назв та імен персоналій, з неструктурованого новинного потоку виникає проблема їхньої високої морфологічної варіативності. У природних мовах одне й те саме слово може зустрічатися у тексті в десятках різних форм. Збереження таких сирих даних у реляційній базі даних призводить до утворення множинних семантичних дублікатів, наприклад, текстові токени «Україна», «Україною» та «в Україні» розпізнаються системою як три різні об'єкти. Це унеможливлює коректну агрегацію кількісних показників та руйнує цілісність аналітичних звітів і засобів візуалізації.

Для забезпечення узгодженості та високої якості даних у розроблюваному конвеєрі застосовуються методи текстової нормалізації та лемматизації. Лемматизація являє собою процес алгоритмічного зведення

видобутої словоформи до її базової, словникової форми (леми), якою для іменників є називний відмінок однини (інфінітивна форма).

Програмна реалізація цього етапу повинна здійснюватися із залученням спеціалізованих бібліотек морфологічного аналізу (наприклад, pymorphy3 [5]), які містять великі лінгвістичні словники та використовують евристичні алгоритми для визначення початкової форми слова. Інтелектуальний аналізатор розбирає кожен отриманий від NER-моделі токен, визначає його частину мови, відмінок і число, після чого генерує його нормалізований еквівалент (наприклад, трансформує «Іраном» у базове «Іран»).

Приклад нормалізації видобутих сутностей за допомогою бібліотеки pymorphy3 наведений на рисунку 1.7.

Рисунок 1.7 – Приклад нормалізації видобутих сутностей із застосуванням спеціалізованої бібліотеки

Окремим, не менш важливим етапом нормалізації є обробка специфічних винятків – загальноприйнятих географічних або політичних абревіатур (наприклад, «США», «КНДР», «ОАЕ») та складних складених назв. Оскільки стандартні алгоритми лемматизації можуть некоректно обробляти акроніми (наприклад, зводячи їх до нижнього регістру з однією великою літерою), конвеєр обробки даних доповнюється програмними фільтрами із застосуванням словників жорстких замін.

Завдяки впровадженню такого комплексного підходу до лемматизації, загальний масив видобутих сутностей приводиться до єдиного стандарту перед збереженням на жорсткий диск чи у базу даних. Це є критично необхідною умовою для їх подальшого експорту у форматах зведених таблиць (CSV) та завантаження до систем бізнес-аналітики (BI), оскільки саме стандартизовані дані дозволяють картографічним сервісам безпомилково ідентифікувати локації, а аналітичним алгоритмам – будувати точні графіки частотності та інтерактивні хмари тегів.

1.2 Характеристика сучасних моделей для генерації дайджестів та синтезу мовлення

В умовах інформаційного перенавантаження просте агрегування та класифікація новинних повідомлень є недостатніми для забезпечення комфортного споживання контенту кінцевим користувачем. Критично важливим етапом у конвеєрі обробки даних, що передує їх безпосередньому перетворенню в аудіо-формат, є підготовка цілісного, логічно зв'язного та стислого тексту остаточного дайджесту.

Спроба прямого синтезу мовлення з вихідних повідомлень Telegram-каналів призводить до низької якості фінального аудіопродукту через наявність обірваних фраз, надмірної деталізації, неформальної лексики та відсутності логічних переходів між різними новинами.

Для зменшення інформаційного шуму можна було б використати стандартні алгоритми екстрактивної сумаризації (Extractive Summarization). Проте, їх застосування до масиву коротких новинних повідомлень виявилося неефективним і концептуально невідповідним завданням системи.

Приклад роботи базового конвеєра сумаризації тексту наведено на рисунку 1.8.

```
article = ''' We describe a convolutional neural network that learns\
 feature representations for short textual posts using hashtags as a\
  supervised signal. The proposed approach is trained on up to 5.5 \
  billion words predicting 100,000 possible hashtags. As well as strong\
   performance on the hashtag prediction task itself, we show that its \
   learned representation of text (ignoring the hashtag labels) is useful\
    for other tasks as well. To that end, we present results on a document\
     recommendation task, where it also outperforms a number of baselines.
'''
summarizer(article)
# [{'summary_text': 'REF proposed a convolutional neural network
# that learns feature representations for short textual posts
# using hashtags as a supervised signal.'}]
```

Рисунок 1.8 – Приклад надмірного стиснення тексту базовим алгоритмом сумаризації

Як видно з наведеного програмного лістингу, стандартна модель здійснює занадто агресивне стиснення вхідного тексту. Алгоритм фактично виокремлює лише одне головне речення, повністю відкидаючи контекст, розгорнуті пояснення та важливі деталі. У випадку обробки агрегованих повідомлень з Telegram, такий підхід призводить до втрати критично важливої фактології, перетворюючи новинний дайджест на сухий набір непов'язаних заголовків. Це є неприйнятним для формату аудіо-подкасту, який вимагає цілісності та легкості сприйняття на слух.

Для розв'язання цієї проблеми у розробленій системі застосовуються генеративні великі мовні моделі (Large Language Models, LLM). На відміну від екстрактивних методів сумаризації, які лише жорстко компілюють найважливіші речення з оригінального тексту, сучасні LLM (зокрема, архітектури Qwen [3] або LLaMA) використовують підхід керованої абстрактивної сумаризації. Вони здатні не просто механічно скорочувати обсяг даних, а повністю переписувати масив текстів, синтезуючи єдине, логічно зв'язне аналітичне зведення зі збереженням контексту та створенням природних дикторських переходів між новинами.

Це означає, що модель здатна глибоко аналізувати семантику декількох розрізнених повідомлень, виділяти головну суть та генерувати абсолютно новий, зв'язний текст у форматі професійного радіо-монологу або подкасту. Застосування методів квантування (наприклад, 4-бітного стиснення вагів моделі) дозволяє розгортати такі потужні генеративні мережі локально, забезпечуючи високу швидкість обробки даних без потреби у надвисоких обчислювальних потужностях.

Після того як LLM формує структурований сценарій дайджесту, ініціюється фінальний етап – перетворення тексту в аудіо за допомогою систем синтезу мовлення (Text-to-Speech, TTS). Сучасні нейромережеві моделі генерації голосу (наприклад, архітектура Silero TTS [4]) суттєво відрізняються від класичних роботизованих синтезаторів. Вони аналізують не лише фонетичний склад слів, але й синтаксичну структуру речення, розставляючи природні інтонаційні акценти, паузи та органічні наголоси, що критично важливо для слухового сприйняття новин.

Оскільки глибокі TTS-моделі мають технологічні обмеження щодо довжини тексту, який може бути оброблений за одну ітерацію (зазвичай до 800–1000 символів), програмна реалізація системи вимагає використання алгоритмів інтелектуального сегментування. Згенерований мовною моделлю текст дайджесту автоматично розбивається на семантично завершені фрагменти (чанки), кожен з яких озвучується окремо. Після цього згенеровані аудіо-тензори об'єднуються у єдиний безперервний аудіо-файл за допомогою матричних операцій.

Таким чином, комбінація абстрактивної сумаризації на базі LLM та сучасного нейромережевого синтезу мовлення дозволяє повністю автоматизувати процес створення якісного аудіо-контенту, перетворюючи розрізнений потік текстових даних на готовий до прослуховування інформаційний продукт.

1.3 Огляд існуючих систем моніторингу новин та засобів візуалізації

1.3.1 Системи моніторингу та агрегації новинного контенту

Сучасний ринок програмного забезпечення пропонує значну кількість інструментів для моніторингу інформаційного простору, зокрема соціальних мереж та месенджерів. Більшість існуючих рішень, серед яких популярні міжнародні RSS-агрегатори новин (наприклад, Feedly [11]) або спеціалізовані сервіси аналітики Telegram-каналів (наприклад, TGStat [12]), функціонують переважно на основі парадигми лексичного пошуку за наперед заданими ключовими словами, тегами або регулярними виразами.

Такі платформи глибоко фокусуються на кількісних та маркетингових метриках, залишаючи поза увагою безпосередній зміст контенту. Наприклад, базова аналітична панель (дашборд) популярного сервісу TGStat пропонує користувачам вичерпну статистичну інформацію щодо життєдіяльності каналу. Інтерфейс та ключові можливості платформи наведено на рисунку 1.9.

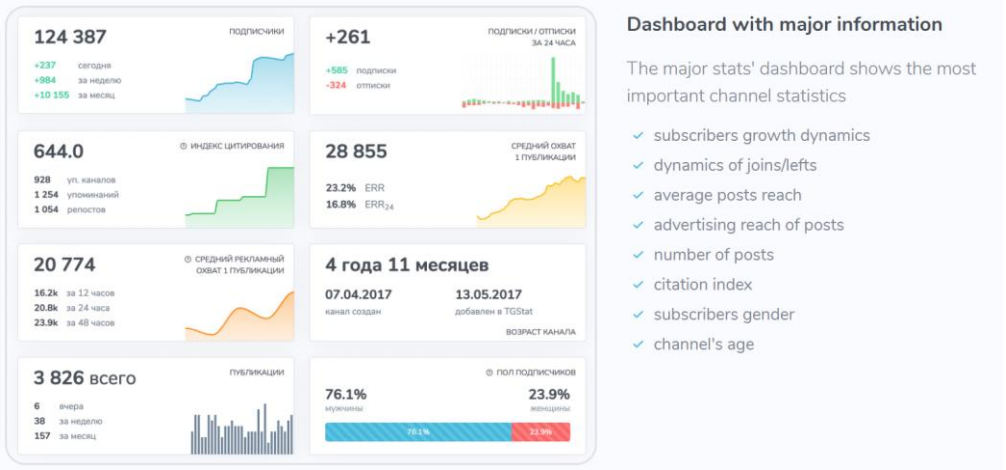


Рисунок 1.9 – Аналітична панель сервісу TGStat із ключовою статистикою Telegram-каналу

Як видно з наведеного рисунка, функціонал TGStat зосереджений на наданні таких кількісних показників, як динаміка приросту та відтоку підписників, середнє охоплення однієї публікації, загальна кількість постів, індекс цитування каналу та гендерний розподіл аудиторії.

Проте, незважаючи на статистичний апарат, подібним платформам бракує інструментів для семантичного аналізу текстового масиву. Вони дозволяють оцінити популярність або релевантність новин у каналі за кількістю переглядів, реакцій або коментарів, але не дають відповіді на питання про змістовну частину каналу. Відсутність функціоналу для автоматичного розпізнавання сутностей, контекстної типізації новин без ключових слів та абстрактивної сумаризації змушує аналітика вручну опрацьовувати великі обсяги повідомлень. Саме ці обмеження існуючих комерційних рішень обумовлюють актуальність розробки інтелектуального NLP-пайплайну, який би змістив фокус із кількісної статистики на глибокий семантичний розбір тексту та зручну генерацію аудіо-дайджестів.

Окрім того, платформи для медіа-моніторингу рідко пропонують вбудовані механізми абстрактивної сумаризації на базі великих мовних моделей та подальшого перетворення тексту в аудіо-формат для зручного прослуховування, обмежуючись лише стандартними текстовими сповіщеннями.

Окрім можливої відсутності інтелектуальних інструментів генерації контенту, фундаментальною проблемою існуючих агрегаторів є їхня нездатність формувати аналітичні звіти, готові для прямої візуалізації у системах Business Intelligence. Інтерактивні платформи візуалізації даних, такі як Power BI, вимагають подачі нормалізованих, типізованих табличних наборів. Натомість класичні агрегатори дозволяють вивантажувати лише непідготовлені тексти повідомлень. Це робить неможливим побудову інтерактивних дашбордів, географічних мап згадуваності або динамічних графіків трендів без розробки та впровадження додаткових проміжних NLP-алгоритмів екстракції та очищення сутностей.

1.3.2 Засоби візуалізації та бізнес-аналітики (BI)

Для аналізу зібраних даних та виявлення прихованих закономірностей у корпоративному та дослідницькому секторах широко використовуються системи класу Business Intelligence (наприклад, Microsoft Power BI [6]). Ці платформи володіють потужним інструментарієм для побудови інтерактивних дашбордів, розрахунку динамічних метрик та відображення географічних даних на інтерактивних мапах. Демонстрацію концептуальної архітектури збору даних та їх подальшої візуалізації у середовищі Power BI наведено на рисунку 1.10.

Рисунок 1.10 – Схема інтеграції різноманітних джерел даних та побудови аналітичних дашбордів у Microsoft Power BI

Проте, фундаментальним обмеженням будь-якої класичної BI-системи є її нездатність самостійно працювати з неструктурованими текстовими масивами. Для того щоб побудувати мапу згадуваності країн або хмару тегів із прізвищами, BI-платформа вимагає подачі вже структурованих, нормалізованих та типізованих даних у табличному форматі.

Жодна з існуючих BI-систем не має вбудованих інструментів для глибокого NLP-аналізу безпосередньо з безперервного потоку повідомлень месенджера. Саме це робить розробку власної проміжної NLP-системи критично необхідним архітектурним кроком.

2 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

2.1 Постановка задачі дослідження

Метою роботи є розробка комплексної автоматизованої інформаційної системи (NLP-пайплайну) для парсингу неструктурованих текстових даних із Telegram-каналів, їх глибокого інтелектуального аналізу, генерації зв'язних аудіо-дайджестів та підготовки нормалізованих баз даних для інтерактивних систем бізнес-аналітики. Загальна структура застосунку зображена на рисунку 2.1.

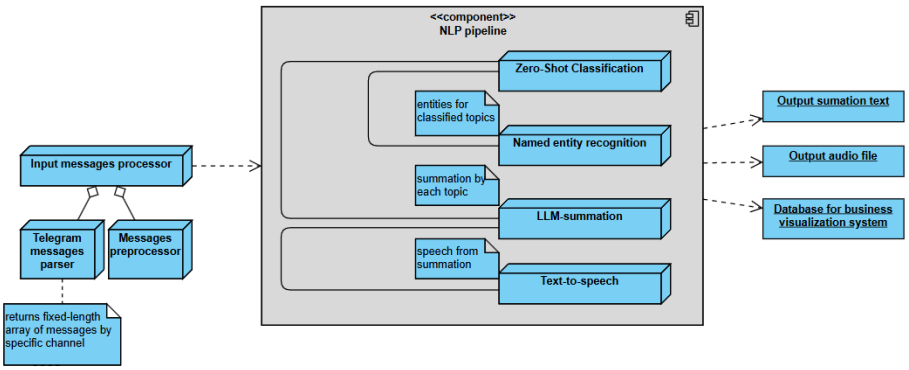


Рисунок 2.1 – Загальний функціонал застосунку

Загальний функціонал системи (рисунок 2.1), який визначає задачу дослідження, повинен включати наступні етапи обробки даних:

– збір неструктурованих текстових даних (повідомлень) з обраних Telegram-каналів за їх унікальними ідентифікаторами;

– попередню програмну обробку та очищення зібраних текстів від інформаційного «шуму» та артефактів форматування;

– динамічну класифікацію повідомлень за довільними тематичними рубриками (топіками) на основі семантичного контексту;

– видобування ключових іменованих сутностей (осіб, географічних назв, організацій) для кожного сформованого топіка;

– генерацію абстрактивного зв'язного тексту дайджесту на основі згрупованих за топіками новин;

– синтез високоякісного аудіо-контенту з отриманого тексту дайджесту для зручного прослуховування користувачем;

– формування та збереження структурованої бази даних видобутих сутностей для подальшої візуалізації на дашбордах BI-систем.

2.2 Обґрунтування вибору програмних засобів реалізації

Для реалізації розробленого конвеєра обробки даних було обрано мікросервісну архітектуру з використанням мови програмування Python як основної платформи для розгортання моделей машинного навчання. Вибір Python зумовлений наявністю потужної екосистеми бібліотек для тензорних обчислень (PyTorch) та роботи з трансформерними архітектурами (Hugging Face Transformers). Для забезпечення максимальної швидкодії системи всі важкі нейромережеві обчислення виконуються з використанням апаратного прискорення на базі графічних процесорів (GPU) за допомогою технології CUDA.

Програмна реалізація аналітичного блоку (NLP-пайплайну) базується на використанні відповідних моделей та інструментів.

Для динамічного розподілу новин за рубриками застосовується багатомовна модель mDeBERTa-v3-base-mnli-xnli. Вона оптимізована для задачі логічного виведення (NLI) і забезпечує високу точність у режимі Zero-Shot. Для підвищення пропускної здатності системи обробка повідомлень виконується пакетами (параметр batch_size), що дозволяє ефективно утилізувати ресурси графічного прискорювача. Модель здатна обробляти довільні міткі класів, що можуть змінюватись за потреби користувача або розробника, що збільшує можливий обсяг застосування моделі та робить процес класифікації новин за рубриками більш гнучким.

Видобування ключових осіб, географічних назв та організацій здійснюється за допомогою мультиязичної моделі gliner_multi-v2.1. Завдяки двонаправленій архітектурі вона демонструє високу адаптивність до різних мов новинного потоку (зокрема української) та дозволяє видобувати сутності за довільними текстовими мітками (промптами). На відміну від класичних NER-моделей мітки класів в обраній можуть бути змінені для інтелектуальної класифікації довільних сутностей.

Для лемматизації видобутих сутностей інтегровано бібліотеку морфологічного аналізу pymorphy3 з підключеним словником української мови. Це гарантує зведення слів до словникової форми перед їхнім збереженням у базу даних.

Для підготовки кінцевого інформаційного продукту застосовуються генеративні нейромережі.

Формування зв'язного тексту дайджесту реалізовано на базі великої мовної моделі Qwen2.5-1.5B-Instruct, яка спеціально донавчена для виконання інструкцій. З метою оптимізації споживання відеопам'яті та прискорення інференсу, тензори моделі завантажуються у форматі напівточної плаваючої коми (bfloat16).

Генерація аудіо-подкастів здійснюється локально за допомогою нейромережевої архітектури Silero TTS. У системі використовується спеціалізована україномовна конфігурація моделі (v3_ua) з акустичним профілем диктора mykyta. Генерація аудіопотоку відбувається з високою частотою дискретизації, що забезпечує студійну якість звучання, придатну для комфортного прослуховування.

Додатковим інструментарієм конвеєра виступають утиліта mitmproxy для інтерактивного перехоплення мережевого трафіку під час вивчення первісної структури даних та проведення процедури їхнього парсингу, а також фреймворки розробки API (FastAPI) та клієнтських інтерфейсів (C#), які забезпечують взаємодію користувача із розробленою AI-системою та експорт нормалізованих даних у середовище Power BI.

2.3 Обґрунтування розробки власного механізму екстракції даних

Одним із ключових етапів роботи розробленого конвеєра обробки природної мови є безперебійне отримання вхідних текстових даних із цільових джерел. Незважаючи на те, що платформа Telegram надає офіційний інструментарій для розробників (Telegram Bot API та Telegram API/MTProto), їх використання для задачі масового моніторингу довільних новинних каналів має суттєві обмеження. Зокрема, стандартні боти не здатні зчитувати історію повідомлень з каналів, де вони не призначені адміністраторами, а клієнтські API мають жорсткі ліміти на кількість запитів (Rate Limits), порушення яких призводить до тимчасового або постійного блокування облікового запису розробника.

З метою створення універсального та незалежного рішення, здатного агрегувати інформацію з будь-якого відкритого Telegram-каналу без прив'язки до офіційних API-обмежень, було прийнято рішення щодо розробки власного програмного парсера. В основі цього підходу лежить метод дослідження та реверс-інжинірингу мережевого трафіку офіційного веб-клієнта месенджера.

Для реалізації цього завдання використовується інструмент інтерактивного перехоплення HTTPS-запитів mitmproxy [9]. Цей засіб дозволяє виконувати перехоплення трафіку в локальному та контрольованому середовищі розробника, дешифрувати захищені канали зв'язку та детально аналізувати структуру запитів (заголовки, параметри) і формат відповідей (JSON/HTML-структури), якими веб-клієнт обмінюється із серверами Telegram.

Результати такого аналізу мережевого трафіку дозволяють точно відтворити необхідні HTTPS-запити програмним шляхом на стороні розроблюваного застосунку для їх подальшої передачі до NLP-модулів системи, не зважаючи при цьому на архітектурні обмеження стандартних API.

3 ПРОЕКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Розробка механізму парсингу та попередньої обробки даних

3.1.1 Реверс-інжиніринг мережевого трафіку Telegram

Першим етапом розробки власного механізму екстракції даних стало дослідження внутрішньої взаємодії між публічним веб-клієнтом Telegram та його серверами. Для цього було застосовано програмний комплекс mitmproxy (зокрема, його графічний веб-інтерфейс mitmweb), який дозволив виконати перехоплення, маршрутизацію та дешифрування HTTPS-трафіку під час сеансу завантаження історії повідомлень з відкритого цільового каналу.

Процес інспектування цільового мережевого запиту та аналізу отриманої відповіді наведено на рисунку 3.1.

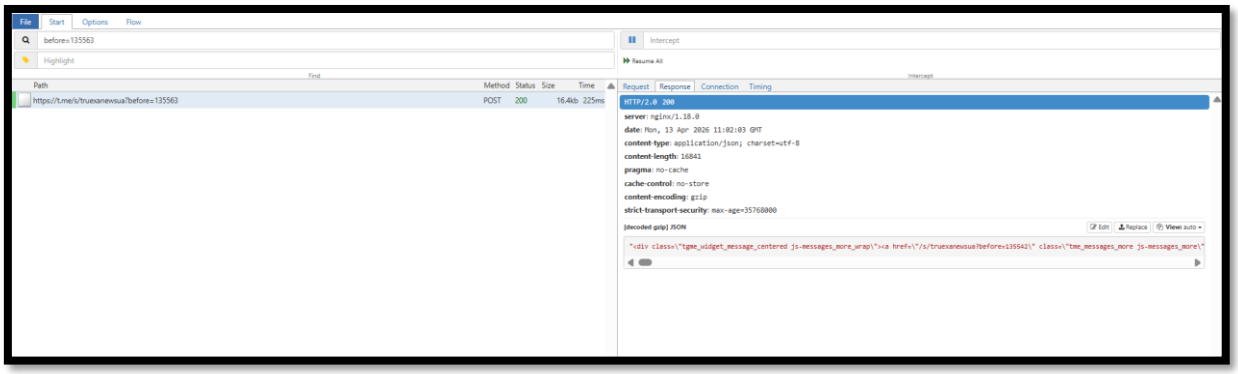


Рисунок 3.1 – Графічний інтерфейс mitmproxy під час перехоплення та аналізу запиту до серверів Telegram

Як видно з наведеної панелі моніторингу трафіку, під час спроби клієнта завантажити попередні повідомлення (скролінг історії каналу), ініціюється POST-запит до внутрішнього ендпоінту формату https://t.me/s/{channel_name}. Ключовою архітектурною особливістю цього

запиту є використання спеціального URL-параметра ?before={message_id}. Цей параметр виконує роль вказівника (курсора) для алгоритму пагінації: він вказує серверу, починаючи з якого ідентифікатора повідомлення необхідно сформувати та повернути попередній блок історичних даних.

Детальний аналіз панелі відповідей демонструє специфіку формату даних, якими оперує Telegram. Сервер повертає успішний статус-код HTTP/2.0 200 OK із типом контенту application/json. З метою оптимізації мережевого навантаження корисне навантаження передається у стисненому вигляді (заголовок content-encoding: gzip), що дозволяє значно збільшити обсяги парсингу без значного підвищення навантаження на мережу виконання запитів.

Після автоматичного декодування gzip-пакета інструментом mitmproxy стає очевидним, що значенням у повернутій JSON-структурі є готова згенерована HTML-розмітка. Цей HTML-код містить DOM-елементи, всередині яких інкапсульовано безпосередній текст повідомлень, метадані, час публікації та посилання на медіа-файли.

Виявлення цього недокументованого механізму динамічного завантаження історії через веб-прев'ю стало фундаментальною базою для розробки програмного парсера. Воно довело практичну можливість алгоритмічної імітації клієнтських запитів: шляхом програмної генерації відповідних POST-запитів із динамічною зміною параметра ?before= можна ітеративно вивантажувати масиви даних з будь-якого відкритого каналу, повністю обходячи архітектурні обмеження та ліміти офіційного Telegram API.

Після успішного отримання, декодування та вилучення HTML-розмітки з JSON-відповіді сервера, наступним кроком є структурний аналіз отриманого документа для побудови алгоритму парсингу.

Детальну структуру DOM-дерева (Document Object Model), яке інкапсулює історію повідомлень цільового каналу, наведено на рисунку 3.2.

Рисунок 3.2 – Структура DOM-дерева веб-прев'ю Telegram-каналу із масивом повідомлень

Як видно з наведеного фрагмента розмітки, архітектура подання даних має чітку ієрархічну та циклічну структуру. Усі повідомлення, що входять до запитаного блоку історії, згруповані всередині єдиного батьківського контейнера – тегу <section> із класами tgme_channel_history js-message_history.

Кожне окреме повідомлення новинного каналу представлене у вигляді незалежного HTML-елемента із класом tgme_widget_message_wrap js-widget_message_wrap. Фактично, ця розмітка утворює ітерований масив вузлів (нодів), кожен з яких виступає контейнером для конкретного поста. Усередині цих блоків міститься повний набір непідготовлених даних: безпосередньо текстовий контент, посилання на медіа-вкладення, мітки часу публікації (timestamps) та статистичні показники переглядів.

Така стандартизована ієрархія DOM-дерева суттєво оптимізує задачу програмної екстракції. Для видобування корисного навантаження системі достатньо застосувати алгоритми синтаксичного аналізу HTML (наприклад,

за допомогою бібліотек класу BeautifulSoup). Програмний парсер знаходить батьківський контейнер, ітерує масив div-елементів повідомлень та вилучає з них внутрішній текстовий контент. Отриманий масив неструктурованих текстів стає первинною інформаційною базою, яка передається на наступний етап конвеєра – до модуля лексичного препроцесингу.

Після ідентифікації батьківського контейнера кожного окремого повідомлення, виникає необхідність глибокого синтаксичного розбору його внутрішньої DOM-структури для екстракції корисного навантаження. Детальний аналіз HTML-розмітки окремого поста демонструє, що текстова та статистична інформація жорстко сегментована за різними вузлами та класами. Внутрішню будову текстового блоку та блоку метаданих наведено на рисунках 3.3 та 3.4 відповідно.

Рисунок 3.3 – Структура текстового вузла повідомлення з елементами форматування

Як видно з рисунка 3.3, безпосередній інформаційний контент інкапсулюється всередині тегу . Проте сирий вміст цього блоку не є суцільним рядковим типом даних (plain text). Збереження авторського форматування Telegram призводить до того, що текст фрагментується на множину дочірніх HTML-елементів. Наприклад, виділення жирним шрифтом формується тегами <b>,

гіперпосилання загортаються в теги <a>, а розриви рядків імітуються тегами
. Окрему складність для алгоритмів обробки природної мови становлять графічні емодзі, які представлені не стандартними Unicode-символами, а складними конструкціями у вигляді тегів <i class="emoji"> із підключеними фоновими зображеннями.

Така гетерогенна структура вимагає обов'язкового етапу лексичного препроцесингу перед передачею даних до NLP-конвеєра. Програмний парсер (наприклад, із застосуванням методів вилучення тексту з дерева у бібліотеці BeautifulSoup) повинен рекурсивно обійти всі дочірні вузли, зчитати виключно текстові значення, не враховувати графічні артефакти та ліквідувати гіперпосилання, формуючи на виході нормалізований рядок тексту.

Рисунок 3.4 – Структура футеру повідомлення

Окрім безпосередньо текстового контенту, архітектура DOM-дерева повідомлення містить цінний масив метаданих, зосереджений у блоках та (рисунок 3.4). З цих вузлів система парсингу здатна алгоритмічно екстрагувати кількісні та часові показники,

які супроводжують кожну новину. Зокрема, кількість унікальних переглядів міститься у текстовому значенні тегу , а точний час публікації у стандартизованому форматі ISO 8601 (з урахуванням часового поясу) фіксується в атрибуті datetime елемента <time>. Крім того, блок реакцій містить кількісні оцінки емоційного відгуку аудиторії на конкретний пост.

Незважаючи на те, що першочерговою задачею розроблюваної системи є NLP-аналіз тексту (сумаризація та розпізнавання сутностей), екстракція цих метаданих має критичне значення для побудови комплексної аналітики. Збереження точного часу публікації та кількості переглядів разом із видобутими текстовими сутностями у реляційній базі даних формує потужний фундамент для систем Business Intelligence.

Це дозволяє майбутнім користувачам інтерактивних дашбордів (у середовищі Power BI) не лише бачити згадуваність об'єктів, але й здійснювати хронологічну фільтрацію новин, відслідковувати динаміку популярності конкретних тем та сортувати згенеровані аудіо-дайджести за рівнем суспільного резонансу, спираючись на кількісті переглядів чи реакцій.

Для забезпечення безперервного збору великих масивів історичних даних критично важливим елементом парсера є механізм автоматичної пагінації.

Оскільки сервер Telegram повертає історію повідомлень лімітованими блоками, системі необхідно динамічно визначати точку відліку для кожного наступного мережевого запиту. Це включає знаходження та зчитування значення для наступного ?before= параметра, формування на основі цього значення подальшого запиту та рекурсивне продовження алгоритму доки не буде досягнута задана максимальна глубина рекурсивних викликів або не буде вичерпана вся історія повідомлень Telegram-каналу.

Алгоритм екстракції цього маркера пагінації наведено на рисунку 3.5.

Рисунок 3.5 – Екстракція токена пагінації з мета-тегів HTML-документа

Детальний аналіз заголовної частини (<head>) отриманого HTML-документа свідчить, що на самому початку кожної завантаженої сторінки сервер передає навігаційні метадані. Зокрема, ідентифікатор для отримання наступної ітерації повідомлень (при русі вглиб історії) міститься у службовому тезі <link rel="prev">. Атрибут href цього елемента містить готове відформатоване посилання із необхідним URL-параметром, наприклад, ?before=131492 (рисунок 3.5).

Програмне вилучення цього значення (за допомогою парсингу атрибутів тегу <link>) дозволяє розробленому механізму працювати у повністю автономному рекурсивному циклі:

– парсер завантажує поточну сторінку (блок) каналу;
– екстрагує та очищує масив повідомлень для подальшої NLP-обробки;
– зчитує токен пагінації з верхньої частини документа;
– формує новий HTTPS-запит із підставленим токеном для отримання попереднього блоку історії.

Цей ітеративний процес (лістинг 3.1) триває до досягнення заданої глибини парсингу (ліміту кількості повідомлень, встановленого користувачем або досягнення максимальної кількості повідомлень в каналі). Таким чином, забезпечується надійна та масштабована екстракція даних будь-якого обсягу без втручання оператора.

Лістинг 3.1 – Асинхронний рекурсивний метод екстракції повідомлень Telegram-каналу

```
    public async Task<List<TelegramDataRecord>>
GetChannelDataAsyncRecursively(
        string channelName,
        string nextToken = "",
        int currentRequestCount = 0,
        int maxRequestCount = 1)
    {
        var url = string.IsNullOrEmpty(nextToken)
    ? $"{BaseUrl}{channelName}"
    : $"{BaseUrl}{channelName}?before={nextToken}";
        var response = await _httpClient.GetAsync(url);
        response.EnsureSuccessStatusCode();

        var content = await
response.Content.ReadAsStringAsync().ConfigureAwait(false);

        var extractedData = ExtractData(content);
        var nextSearchToken = GetNextToken(content);

        return currentRequestCount == maxRequestCount
            ? extractedData
            :
            [
                .. extractedData,
                .. await
GetChannelDataAsyncRecursively(channelName: channelName,
nextSearchToken, currentRequestCount + 1,
maxRequestCount).ConfigureAwait(false)
            ];
    }
```

Як видно з лістингу 3.1, метод GetChannelDataAsyncRecursively приймає ідентифікатор каналу та поточний токен пагінації. На основі цих

даних динамічно формується цільовий URL. Після виконання асинхронного HTTP-запиту (через об'єкт _httpClient) отриманий HTML-контент передається у допоміжні методи ExtractData (для синтаксичного розбору повідомлень) та GetNextToken (для пошуку наступного маркера пагінації). Ключовою особливістю імплементації є використання рекурсивного виклику із застосуванням синтаксису "spread operator" (..) для конкатенації списків. Це дозволяє ефективно накопичувати екстраговані масиви TelegramDataRecord у єдину колекцію, контролюючи глибину занурення в історію за допомогою лічильника currentRequestCount. Окрім цього, перевірка на успішність виконання кожного запиту при умові імплементації try-catch блоку дозволяє завчасно обірвати рекурсивне виконання методу та дає можливість повернути результат попередніх операцій, що встигли накопитися у цільовому списку.

Таким чином, розроблений модуль парсингу виконує комплексне перетворення сирої HTML-розмітки на структурований інформаційний об'єкт (DTO) із можливістю кількісного встановлення глибини зчитування історичних даних, готовий до подальшого машинного навчання та збереження.

3.1.2 Препроцесинг тексту повідомлень

Отриманий після HTML-парсингу масив повідомлень все ще містить значну кількість лексичного шуму: пунктуаційні артефакти, спеціальні символи, емодзі, а також типові для Telegram-каналів рекламні приписки, посилання. Для підготовки тексту до подальшої обробки трансформерними моделями (які є чутливими до нестандартних токенів), було розроблено алгоритм глибокої лексичної нормалізації.

Програмну імплементацію цього етапу наведено на лістингу 3.2.

Лістинг 3.2 – Метод лексичного препроцесингу та очищення текстових повідомлень

```
public List<string> PreprocessText(List<string>? texts)
{
    var preprocessedTexts = texts
        .Select(t =>
        {
            var splitted = t.Split("\n");
            var splittedLength = splitted.Length;

            return splittedLength > 1
                ? string.Join(". ", splitted[..^1])
                : t;
        })
        .Select(t =>
        {
            var onlyDigitsAndWords = Regex.Replace(t, @"[^\p{L}\p{N}\s]", " ");
            var deletedWhiteSpaces = Regex.Replace(onlyDigitsAndWords, @"\s+", " ");
            return deletedWhiteSpaces.Trim();
        })
        .Where(t => !string.IsNullOrEmpty(t) && t.Split(" ").Length >= 3)
        .ToList();

    return preprocessedTexts;
}
```

Як видно з наведеного коду (лістинг 3.2), процес нормалізації побудовано у вигляді ланцюжка LINQ-перетворень, який включає необхідні етапи препроцесингу.

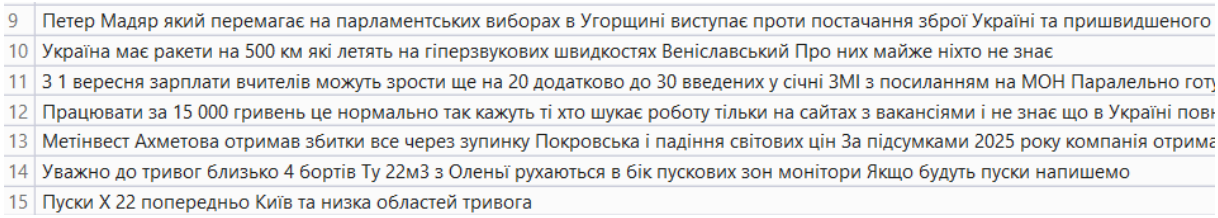
За допомогою методу Split("\n") текст розбивається на абзаци. Якщо повідомлення багаторядкове, алгоритм автоматично відкидає останній

рядок (через використання оператора діапазону [..^1]), замінюючи розриви на крапки. Це дозволяє ефективно видаляти типові заклики на кшталт «Джерело», «Підписатися» або гіперпосилання, які зазвичай розміщуються в кінці публікації.

Наступний крок використовує патерн [^\p{L}\p{N}\s], який залишає виключно літери (\p{L}), цифри (\p{N}) та пробіли, замінюючи всі інші символи (емодзі, хештеги, нестандартну пунктуацію) на пробіли. Після цього послідовності множинних пробілів згортаються в один патерном \s+.

Фінальний фільтр Where відкидає порожні рядки та ультракороткі повідомлення (менше 3 слів), оскільки вони не несуть достатнього семантичного навантаження для подальшої класифікації та сумаризації.

Результат роботи алгоритму препроцесингу на реальних даних Telegram-каналу наведено на рисунку 3.6.



9	Петер Мадяр який перемагає на парламентських виборах в Угорщині виступає проти постачання зброї Україні та пришвидшеного
10	Україна має ракети на 500 км які летять на гіперзвукових швидкостях Веніславський Про них майже ніхто не знає
11	З 1 вересня зарплати вчителів можуть зрости ще на 20 додатково до 30 введених у січні ЗМІ з посиланням на МОН Паралельно гоу
12	Працювати за 15 000 гривень це нормально так кажуть ті хто шукає роботу тільки на сайтах з вакансіями і не знає що в Україні повн
13	Метінвест Ахметова отримав збитки все через зупинку Покровська і падіння світових цін За підсумками 2025 року компанія отрима
14	Уважно до тривог близько 4 бортів Ту 22м3 з Олені рухаються в бік пускових зон монітори Якщо будуть пуски напишемо
15	Пуски Х 22 попередньо Київ та низка областей тривога

Рисунок 3.6 – Фрагмент масиву нормалізованих повідомлень, підготовлених для NLP-обробки

Наведений фрагмент з рисунку 3.6 підтверджує ефективність обраного підходу: на виході система отримує стандартизований формат тексту без графічних та пунктуаційних артефактів. Як можна побачити з рисунку 3.6, цільовий сервіс застосунку формує один великий масив всіх повідомлень, що були вилучені з Telegram-каналу та предоброблені. У такому вигляді дані готові до передачі через API для проходження етапів машинного навчання, зокрема Zero-Shot класифікації та видобування іменованих сутностей.

3.2 Програмна реалізація NLP-пайплайну

Центральним ядром аналітичної системи є серверний додаток, реалізований мовою Python із використанням високопродуктивного веб-фреймворку FastAPI. Архітектурний патерн мікросервісу передбачає, що всі важкі нейромережеві обчислення виконуються на локальному сервері, а клієнтський додаток взаємодіє з ними виключно через REST API запити.

Головною проблемою розгортання локальних Large Language Models та трансформерів є значний час їхнього «холодного старту», оскільки завантаження ваг моделей (тензорів) у пам'ять графічного прискорювача є ресурсомісткою I/O-операцією. Для оптимізації цього процесу та мінімізації часу запуску сервера було застосовано підхід асинхронного багатопотокового завантаження. Програмну реалізацію цього механізму наведено на лістингу 3.3.

Лістинг 3.3 – Багатопотокове завантаження моделей у пам'ять сервера

```
    with concurrent.futures.ThreadPoolExecutor(max_workers=4)
as executor:
        future_ner = executor.submit(load_ner)
        future_topic = executor.submit(load_topic)
        future_llm = executor.submit(load_LLM)
        future_audio = executor.submit(load_audio)

        ner_classifier = future_ner.result()
        topic_classifier = future_topic.result()
        llm_model = future_llm.result()
        audio_model = future_audio.result()
```

Як видно з лістингу 3.3, для паралельної ініціалізації використовується клас ThreadPoolExecutor із пулом у чотири робочі потоки, за що відповідає значення параметру max_workers=4. Кожна модель, що використовується в пайплайні (GLiNER, mDeBERTa, Qwen, Silero)

завантажується у власному потоці (метод submit). Головний потік сервера очікує завершення всіх паралельних завдань за допомогою методу result(). Після успішної ініціалізації об'єкти моделей зберігаються в оперативній пам'яті сервера (як глобальні інстанси) та готові миттєво обробляти вхідні запити через API-ендпоінти без необхідності повторного завантаження ваг з диска.

3.2.1 Програмна реалізація Zero-Shot класифікації

Першим фундаментальним етапом аналізу тексту в рамках розробленого NLP-пайплайну є його семантична категоризація. Оскільки вхідний потік новин з Telegram-каналу є гетерогенним (одночасно містить інформацію про політику, надзвичайні події, економіку тощо), системі необхідно згрупувати повідомлення для забезпечення якісної роботи наступних генеративних моделей. Для цього використовується алгоритм Zero-Shot класифікації на базі багатомовної трансформерної архітектури mDeBERTa. Вона аналізує глибокий контекст кожного нормалізованого повідомлення та привласнює йому найбільш релевантну тематичну мітку із заздалегідь визначеного масиву рубрик.

Проте, під час практичного тестування базового лінійного класифікатора було виявлено архітектурну проблему – неоднозначність міток (Label Ambiguity) та перекриття класів. Наприклад, така рубрика як «Інфраструктура» може фігурувати як у контексті цивільного будівництва, так і в контексті наслідків військових обстрілів. Надання моделі єдиного великого списку міток призводило до зниження точності прогнозування, оскільки функція активації змушувала модель гадати між спорідненими поняттями.

Для вирішення цієї проблеми та підвищення точності розподілу було спроектовано ієрархічну дворівневу модель класифікації, структурну схему якої наведено на рисунку 3.7.

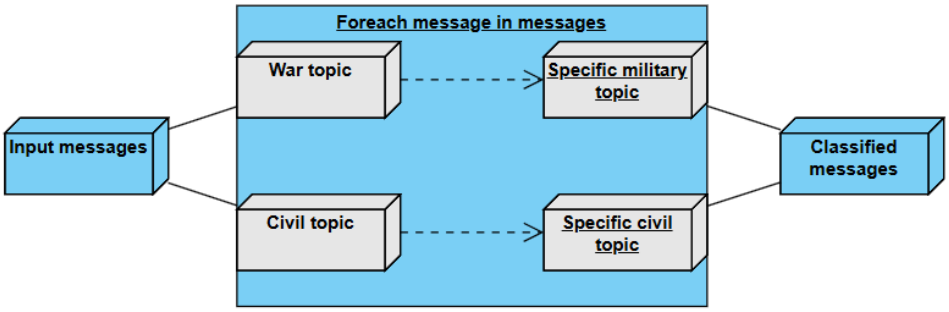
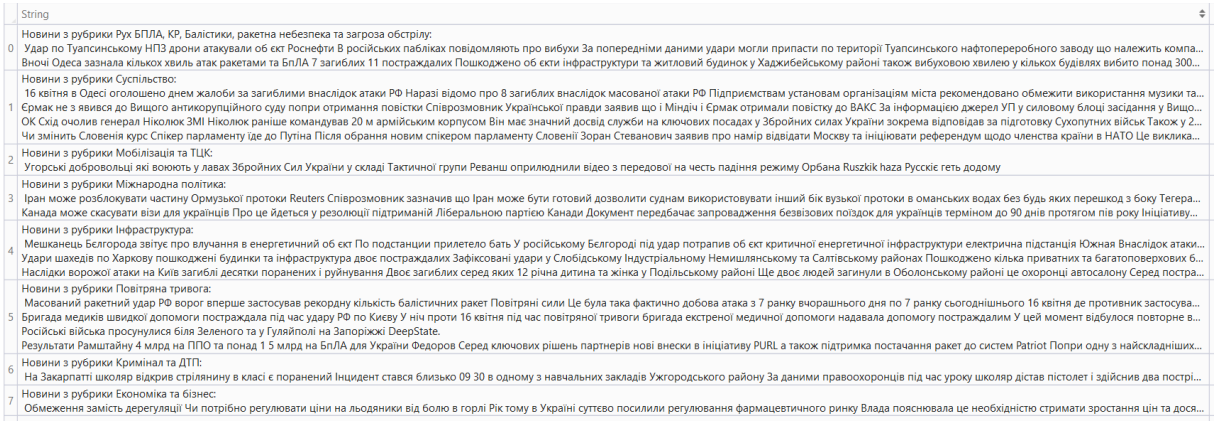


Рисунок 3.7 – Архітектура дворівневої ієрархічної класифікації повідомлень

Як видно з наведеної схеми (рисунок 3.7), процес семантичного розбору алгоритмічно розділено на макро- та мікро-класифікацію. На першому етапі вхідний масив повідомлень (Input messages) проходить глобальну фільтрацію, де модель визначає базовий контекст: військову тематику (War topic) або цивільне життя (Civil topic). Після цього масив текстів програмно розділяється на два відповідні батчі, зберігаючи свої оригінальні індекси.

На другому етапі кожен батч паралельно передається на мікро-класифікацію, де модель обирає вже конкретну вузьку підтему (Specific military topic або Specific civil topic) із двох жорстко ізольованих списків рубрик. Таке проектування дозволяє використовувати однакові мітки, як-от, «Інфраструктура» чи «Енергетика» у різних списках без ризику хибного спрацьовування: якщо новина пройшла через гілку War topic, модель обиратиме виключно між бойовими діями та руйнуваннями, ігноруючи цивільну політику. Такий підхід дозволяє більш якісно формувати звітну інформацію до візуалізації у BI-системах, що критично важливо для проведення аналітики за конкретними типами подій.

Результат роботи програмного комплексу та кінцевого групування повідомлень за топіками наведено на рисунку 3.8.



	String
0	Новини з рубрики Рух БПЛА, КР, Балістики, ракетна небезпека та загроза обстрілу: Удар по Туапсинському НПЗ дрони атакували об єкт Роснефти В російських пабліках повідомляють про вибухи За попередніми даними удари могли припасти по території Туапсинського нафтопереробного заводу що належить компа... Вночі Одеса зазнала кількох хвиль атак ракетами та БпЛА 7 загиблих 11 постраждалих Пошкоджено об єкти інфраструктури та житловий будинок у Хаджибейському районі також вибуховою хвилею у кількох будівлях вибито понад 300...
1	Новини з рубрики Суспільство: 16 квітня в Одесі оголошено днем жалоби за загиблими внаслідок атаки РФ Наразі відомо про 8 загиблих внаслідок масованої атаки РФ Підприємствам установам організаціям міста рекомендовано обмежити використання музики та... Єрмак не з явився до Вищого антикорупційного суду попри отримання повістки Співрозмовник Української правди заявив що і Міндіч і Єрмак отримали повістку до ВАКС За інформацією джерел УП у силовому блоці засідання у Вищо... ОК Схід очолив генерал Ніколюк ЗМІ Ніколюк раніше командував 20 м армійським корпусом Він має значний досвід служби на ключових посадах у Збройних силах України зокрема відповідав за підготовку Сухопутних військ Також у 2... Чи змінить Словенія курс Спікер парламенту їде до Путіна Після обрання новим спікером парламенту Словенії Зоран Стеванович заявив про намір відвідати Москву та ініціювати референдум щодо членства країни в НАТО Це виклика...
2	Новини з рубрики Мобілізація та ТЦК: Угорські добровольці які воюють у лавах Збройних Сил України у складі Тактичної групи Реванш оприлюднили відео з передової на честь падіння режиму Орбана Ruszkik haza Русскіє геть додому
3	Новини з рубрики Міжнародна політика: Іран може розблокувати частину Ормузької протоки Reuters Співрозмовник зазначив що Іран може бути готовий дозволити суднам використовувати інший бік вузької протоки в оманських водах без будь яких перешкод з боку Тегера... Канада може скасувати візи для українців Про це йдеться у резолюції підтриманій Ліберальною партією Канади Документ передбачає запровадження безвізових поїздок для українців терміном до 90 днів протягом пів року Ініціативу...
4	Новини з рубрики Інфраструктура: Мешканець Бєлгорода звітує про влучання в енергетичний об єкт По підстанции прилетело бать У російському Бєлгороді під удар потрапив об єкт критичної енергетичної інфраструктури електрична підстанція Южная Внаслідок атаки... Удари шахедів по Харкову пошкоджені будинки та інфраструктура двоє постраждалих Зафіксовані удари у Слобідському Індустріальному Немишлянському та Салтівському районах Пошкоджено кілька приватних та багатоповерхових б... Наслідки ворожої атаки на Київ загиблі десятки поранених і руйнування Двоє загиблих серед яких 12 річна дитина та жінка у Подільському районі Ще двоє людей загинули в Оболонському районі це охоронці автосалону Серед постра...
5	Новини з рубрики Повітряна тривога: Масований ракетний удар РФ ворог вперше застосував рекордну кількість балістичних ракет Повітряні сили Це була така фактично добова атака з 7 ранку вчорашнього дня по 7 ранку сьогоднішнього 16 квітня де противник застосува... Бригада медиків швидкої допомоги постраждала під час удару РФ по Києву У ніч проти 16 квітня під час повітряної тривоги бригада екстреної медичної допомоги надавала допомогу постраждалим У цей момент відбулося повторне в... Російські війська просунулися біля Зеленого та у Гуляйполі на Запоріжжі DeepState. Результати Рамштайну 4 млрд на ППО та понад 1 5 млрд на БпЛА для України Федоров Серед ключових рішень партнерів нові внески в ініціативу PURL а також підтримка постачання ракет до систем Patriot Попри одну з найскладніших...
6	Новини з рубрики Кримінал та ДТП: На Закарпатті школяр відкрив стрільбу в класі є поранений Інцидент стався близько 09 30 в одному з навчальних закладів Ужгородського району За даними правоохоронців під час уроку школяр дістав пістолет і здійснив два постріл...
7	Новини з рубрики Економіка та бізнес: Обмеження замість дерегуляції Чи потрібно регулювати ціни на льодяники від болю в горлі Рік тому в Україні суттєво посилили регулювання фармацевтичного ринку Влада пояснювала це необхідністю стримати зростання цін та дося...

Рисунок 3.8 – Результат семантичного розподілу новинних повідомлень за цільовими рубриками

Як видно з рисунку 3.8, розроблений алгоритм успішно класифікує суцільний неструктурований текстовий потік на чітко відокремлені логічні блоки. Система демонструє високу адаптивність: серед ідентифікованих рубрик присутні як стандартні загальнонаціональні теми («Суспільство», «Економіка та бізнес», «Міжнародна політика»), так і вузькоспеціалізовані актуальні категорії, що відображають специфіку поточного безпекового середовища («Повітряна тривога», «Мобілізація та ТЦК», «Рух БПЛА, КР, Балістики...»). У кожному рядку згруповано масив текстів, які концептуально належать до однієї сфери.

Такий підхід до попереднього дворівневого сортування відіграє критично важливу роль в архітектурі всього конвеєра. Завдяки тому, що дані передаються на наступні етапи не хаотичним масивом, а структурованими тематичними пулами, вирішуються одразу дві ключові задачі:

– модель екстракції сутностей може точніше прив'язувати знайдені об'єкти (імена, локації) до правильного контексту (наприклад, політик згадується саме в контексті «Міжнародної політики», а локація – у контексті військової «Інфраструктури»);

– під час передачі згрупованих текстів до генеративної моделі повністю усувається проблема розмиття контексту. Модель сумаризує

новини виключно в межах однієї рубрики, що дозволяє формувати максимально точні, глибокі та логічно зв'язні аналітичні дайджести для кожної сфери окремо.

3.2.2 Програмна реалізація NER

Наступним аналітичним кроком після тематичної кластеризації є глибокий інформаційний аналіз тексту за допомогою методів розпізнавання іменованих сутностей (NER). Для кожного сформованого пулу новин ініціюється виклик двонаправленої трансформерної моделі GLiNER. Завдяки попередньому групуванню за рубриками (Zero-Shot), модель здатна більш точно екстрагувати сутності, спираючись на вузький тематичний контекст. Головною задачею цього етапу є виявлення та класифікація ключових об'єктів за визначеними категоріями: політичні та громадські діячі («Особа»), географічні локації («Місто», «Країна») та організації, що класифікуються як («Компанія»).

Результат роботи алгоритму екстракції сутностей, згрупований за відповідними цільовими рубриками, наведено на рисунку 3.9.

	ValueTuple`2.Item1	ValueTuple`2.Item2
0	(Р...	Рух БПЛА, КР, Балістики, ракетна небезпека та загроза обстрілу	[Роснефт-Компанія, Роснефть-Компанія, Одеса-Місто]
1	(С...	Суспільство	[Одеса-Місто, Рф-Країна, Рф-Країна, Єрмак-Особа, Єрмак-Особа, Мінді-Особа, Єрмак-Особа, Словенія-Країна, Словенія-Країна...]
2	(...	Мобілізація та ТЦК	[Україна-Країна]
3	(...	Міжнародна політика	[Іран-Країна, Іран-Країна, Тегеран-Місто, Ізраїль-Країна, Канада-Країна, Канада-Країна, Монреаль-Місто]
4	(І...	Інфраструктура	[Бєлгород-Місто, Бєлгород-Місто, Харків-Місто, Ігор терехов-Особа, Київ-Місто, Київ-Місто, Київ-Місто, Кий-Місто]
5	(...	Повітряна тривога	[Рф-Країна, Юрій ігнат-Особа, Київ-Місто, Рф-Країна, Київ-Місто, Гуляйполе-Місто, Великої британія-Країна, Німеччина-Країна...]
6	(К...	Кримінал та ДТП	[Школяр-Особа, Школяр-Особа]
7	(Е...	Економіка та бізнес	[Україна-Країна]
8	(В...	Війна в Україні	[Донецький-Місто]

Рисунок 3.9 – Екстракція та класифікація іменованих сутностей у розрізі тематичних рубрик

Як видно з наведеної структури даних, система формує зв'язані колекції (наприклад, у форматі ValueTuple), де ключем виступає назва рубрики, а значенням – масив знайдених сутностей із привласненими мітками класу (наприклад, Єрмак-Особа, Одеса-Місто, Роснефть-

Компанія). Детальний аналіз візуалізації демонструє високу контекстну точність розпізнавання алгоритму: так, у рубриці «Міжнародна політика» коректно ідентифіковані релевантні країни та міста (Іран, Канада, Тегеран, Монреаль), тоді як у рубриці «Інфраструктура» виділяються імена посадових осіб та локальні міста (Харків, Київ, Ігор Терехов).

Отримання таких даних відіграє вирішальну роль у підготовці інформаційної бази для кінцевих систем бізнес-аналітики. Знайдені зв'язки «Рубрика – Сутність» дозволяють перетворити вхідний текст на чітку реляційну структуру. Після того як ці видобуті слова пройдуть фінальний етап морфологічної лемматизації (зведення різних відмінків до єдиної називного форми за допомогою pymorphy3), вони будуть збережені у базу даних. Це дозволить клієнтам платформи Power BI будувати інтерактивні географічні мапи згадуваності або графи зв'язків між політиками у розрізі конкретних подій.

3.2.3 Формування дайджесту та вихідного аудіо

Фінальним етапом текстового аналізу є формування загального, логічно зв'язного дайджесту. Згруповані за тематичними рубриками пули повідомлень передаються на вхід генеративній великій мовній моделі (LLM архітектури Qwen) разом із попередньо сконструйованим системним промптом. Завданням моделі є виконання абстрактивної сумаризації: вона як компілює речення з різних новин, так й синтезує єдиний аналітичний текст, виділяючи найважливіші факти та усуваючи дублювання інформації.

Окрім передачу пулу повідомлень з різних рубрик, був протестований формат роботи моделі, що базується на сумаризації тексту в межах одного топіку, такий формат виявився найбільш прийнятним для подальшого формування аудіо-звітів

Приклад згенерованого аналітичного зведення для рубрики «Повітряна тривога» наведено на рисунку 3.10.

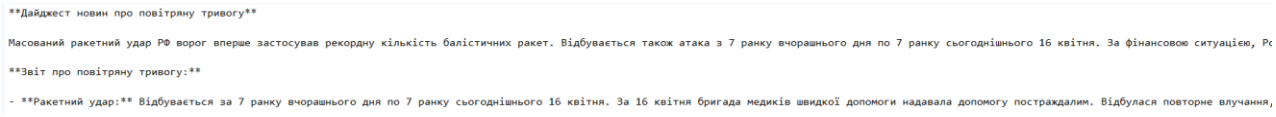


Рисунок 3.10 – Результат абстрактивної сумаризації новин генеративною мовною моделлю

Як видно з наведеного фрагмента, нейромережа успішно синтезувала розрізнені вхідні дані у цілісний звіт. Модель коректно агрегувала ключові події (інформацію про рекордну кількість балістичних ракет, часові рамки атаки з 7 ранку 15 до 16 квітня, а також факти надання допомоги постраждалим бригадами медиків). Структура згенерованого тексту має чіткий розподіл на абзаци та логічні блоки з використанням Markdown-розмітки.

Отриманий текстовий артефакт є передостанньою ланкою NLP-пайплайну. Оскільки кінцевою метою системи є створення аудіо-подкастів, цей текст передається до модуля синтезу мовлення (Text-to-Speech) на базі моделі Silero. Для забезпечення високоякісного та природного дикторського звучання, перед безпосередньою генерацією аудіопотоку (.wav файлу), до тексту застосовується швидкий мікро-препроцесинг: за допомогою регулярних виразів із рядка видаляються всі технічні символи Markdown-розмітки (зірочки **, дефіси списків -), що гарантує безперебійну вокалізацію контенту без артикуляції програмою спеціальних символів.

Проте, фундаментальною архітектурною особливістю більшості нейромереж для синтезу мовлення (зокрема Silero) є жорстке обмеження на максимальну довжину вхідної послідовності символів. Спроба передати весь згенерований дайджест єдиним масивом неминуче призводить до деградації якості звуку або помилок переповнення. Для вирішення цієї проблеми було реалізовано алгоритм семантичного чанкування тексту, програмну імплементацію якого наведено на лістингу 3.4.

Лістинг 3.4 – Алгоритм семантивного чанкування тексту для моделі TTS

```
    def split_text_into_chunks(text: str, max_chars: int = 800)
-> list:
        sentences = re.split(r'(?<=[.!?\n])\s+', text)
        chunks = []
        current_chunk = ""

        for sentence in sentences:
            sentence = sentence.strip()
            if not sentence:
                continue

            if len(current_chunk) + len(sentence) < max_chars:
                current_chunk += sentence + " "
            else:
                if current_chunk:
                    chunks.append(current_chunk.strip())
                current_chunk = sentence + " "

        if current_chunk:
            chunks.append(current_chunk.strip())

        return chunks
```

Як видно з наведеного лістингу, розроблена функція split_text_into_chunks розбиває суцільний текст не сліпим методом по символах, а алгоритмічно виділяє окремі речення, орієнтуючись на розділові знаки (.!?\n). Це критично важливо для збереження правильної інтонації та природних пауз диктора. Далі речення послідовно акумулюються у текстові блоки (чанки), розмір яких контролюється і не перевищує ліміт у 800 символів. Кожен такий блок по черзі передається до

моделі Silero, після чого згенеровані аудіосегменти конкатенуються у єдиний фінальний запис.

Результат роботи цього конвеєра – готові до прослуховування аудіофайли повноцінних новинних подкастів – наведено на рисунку 3.11.

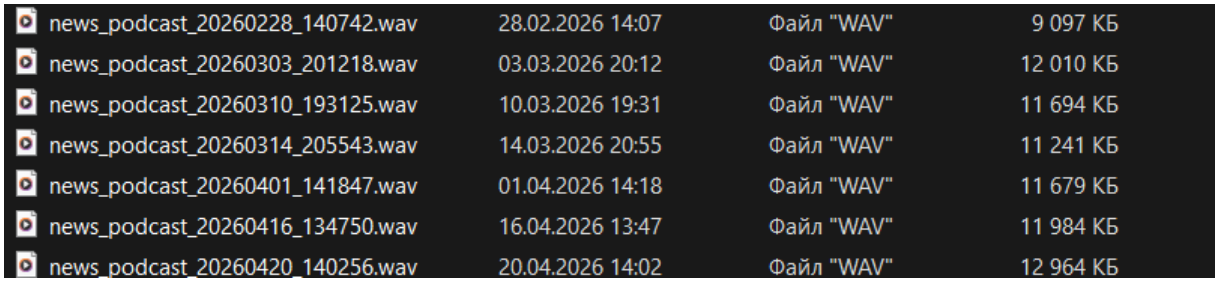


Рисунок 3.11 – Згенеровані аудіофайли новинних подкастів у форматі .wav

Завершується цикл обробки на стороні сервера формуванням комплексного HTTP-відгуку, який містить готовий аудіо-файл дайджесту, що представляється у вигляді байтового масиву, та структурований масив видобутих іменованих сутностей для повернення клієнтському застосунку.

3.3 Проектування бази даних та реалізація клієнтського інтерфейсу

3.3.1 Формування структурованих наборів даних для систем бізнес-аналітики

Після завершення всіх етапів обробки тексту на стороні Python-сервера, клієнтський додаток отримує комплексний відгук, який містить згенерований аудіофайл та масив видобутих метаданих (класифіковані топіки та іменовані сутності). Для того щоб ці результати нейромережевих обчислень стали придатними для візуалізації в системах класу Business Intelligence (зокрема, Microsoft Power BI), клієнтська частина системи

виконує їх агрегацію та трансформацію у табличний формат (реляційну структуру).

Процес трансформації полягає у лемматизації, групуванні та підрахунку частоти згадувань кожної нормалізованої сутності в межах проаналізованого масиву новин за певну дату. Фрагмент згенерованого аналітичного звіту (датасету), підготовленого клієнтським застосунком для експорту, наведено на рисунку 3.12.

	Date	Category	ItemName	Count
2	03.03.2026	Topic	Реклама	4
3	03.03.2026	Topic	Повітряна тривога	15
4	03.03.2026	Topic	Мобілізація та ТЦК	8
5	03.03.2026	Topic	Міжнародна допомога т	17
6	03.03.2026	Topic	Війна в Україні	9
7	03.03.2026	Topic	Рух БПЛА, КР, Балістики,	7
8	03.03.2026	Topic	Міжнародні конфлікти	21
9	03.03.2026	Topic	Суспільство	10
10	03.03.2026	Topic	Благодійність та збори	6
11	03.03.2026	Topic	Корупція та скандали	1
12	03.03.2026	Topic	Інфраструктура	8
13	03.03.2026	Topic	Міжнародна політика	10
14	03.03.2026	Topic	Економіка та бізнес	3
15	03.03.2026	Topic	Спорт	5
16	03.03.2026	Country	Іспанія	3
17	03.03.2026	Country	Іран	19
18	03.03.2026	Country	Саудівській аравія	1
19	03.03.2026	Country	Сша	13
20	03.03.2026	Country	Україна	7
21	03.03.2026	Country	Катар	1
22	03.03.2026	Country	Політика україна	2
23	03.03.2026	Country	Литва	1
24	03.03.2026	Country	Ізраїль	6
25	03.03.2026	Country	Індія	1
26	03.03.2026	Country	Рф	1
27	03.03.2026	City	Вінниця	1

Рисунок 3.12 – Структура агрегованого набору даних для експорту в системи візуалізації

Як видно з наведеного фрагмента, сформований набір даних має строго типізовану структуру, що складається з чотирьох ключових атрибутів. Поле Date фіксує хронологічний маркер проаналізованого пулу новин, що є необхідною умовою для побудови часових рядів та відслідковування динаміки трендів. Атрибут Category виконує роль

універсального класифікатора: він об'єднує як результати роботи алгоритму Zero-Shot (значення Topic), так і класи іменованих сутностей, видобутих моделлю NER (значення Country, City тощо). Колонка ItemName містить безпосередньо лемматизовані назви рубрик чи об'єктів (наприклад, «Міжнародні конфлікти», «Іран», «Вінниця»), а поле Count відображає абсолютний кількісний показник їхньої згадуваності.

Приведення складних ієрархічних JSON-відповідей від NLP-пайплайну до такого уніфікованого табличного формату ефективно розв'язує фундаментальну проблему інтеграції з BI-платформами. Наявність агрегованого поля Count у поєднанні зі стандартизованими категоріями дозволяє аналітикам без додаткового програмування будувати в Power BI складні візуалізації: теплові географічні мапи (відфільтрувавши набір за категоріями Country та City), хмари тегів, а також діаграми розподілу уваги медіа-простору за конкретними темами (Topic). Таким чином, клієнтський додаток виконує роль надійного ETL-модуля (Extract, Transform, Load) між нейромережами та кінцевими аналітичними дашбордами.

3.3.2 Візуалізація результатів аналізу в середовищі Power BI

Експортований на попередньому етапі структурований набір даних слугує надійним фундаментом для побудови інтерактивних аналітичних дашбордів у середовищі Microsoft Power BI. Завдяки попередній лемматизації та нормалізації даних системою, процес візуалізації не потребує застосування складних DAX-запитів для очищення тексту, а зводиться до прямого мапінгу підготовлених атрибутів на графічні віджети.

Першим і базовим елементом візуальної аналітики є оцінка загальної структури інформаційного потоку. Для розуміння того, які саме теми домінували у медіа-просторі за обраний період, на основі агрегованого поля Count (із застосуванням фільтрації за атрибутом Category = Topic) було побудовано кругову діаграму (Pie Chart). Вона відображає питому вагу

кожної класифікованої рубрики у загальному масиві проаналізованих новин, що зображено на рисунку 3.13.

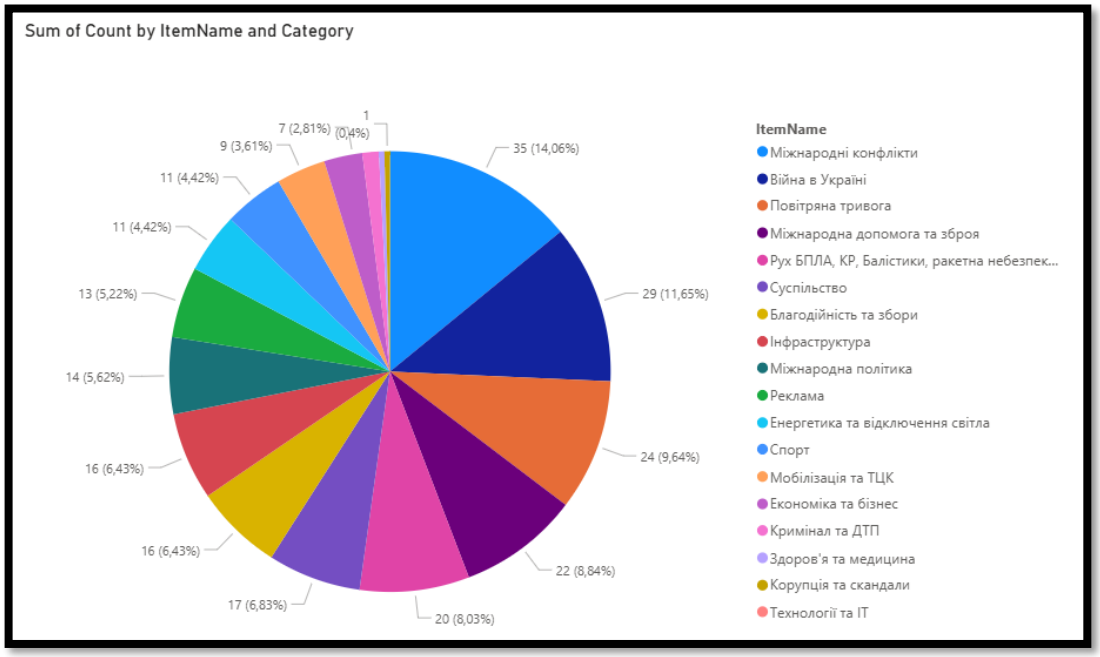


Рисунок 3.13 – Кругова діаграма розподілу новинного потоку за тематичними рубриками

Як видно з рисунку 3.13, побудований дашборд дозволяє наочно оцінити медіа-фокус досліджуваного джерела. Згідно з отриманими результатами роботи Zero-Shot класифікатора, домінуючими темами в інформаційному просторі є «Міжнародні конфлікти» (14,06% від загального обсягу новин), «Війна в Україні» (11,65%) та оперативні повідомлення про безпекову ситуацію, зокрема «Повітряна тривога» (9,64%) і «Рух БПЛА, КР, Балістики» (8,03%). Водночас, меншу, але статистично значущу частку займають рубрики соціально-економічного спрямування («Суспільство», «Інфраструктура»).

Таке графічне представлення має високу практичну цінність для бізнес-аналітики. Воно дозволяє кінцевому споживачеві інформації, наприклад, аналітику, журналісту чи дослідникові медіа-ринку, миттєво

ідентифікувати головні інформаційні тренди та аномалії інформаційного поля без необхідності ручного перегляду та прочитання сотень текстових повідомлень. Крім того, завдяки інтерактивності Power BI, цей графік виступає в ролі активного фільтра: при натисканні на будь-який сектор, як-от, «Міжнародні конфлікти», усі інші візуалізації на дашборді можуть автоматично перерахуватись, відображаючи сутності та локації, пов'язані виключно з цією рубрикою.

Наступним рівнем аналітики, який розкриває потенціал системи розпізнавання іменованих сутностей, є просторовий аналіз інформаційного поля. Для візуалізації географічного фокусу новинного каналу в Power BI було використано віджет інтерактивної бульбашкової мапи (Azure Maps). Ця візуалізація будується шляхом застосування фільтрації датасету за умовою Category = Country. При цьому лемматизовані назви держав з колонки ItemName автоматично геокодуються платформою, а показник Count визначає радіус відповідного маркера на мапі.

Результат просторового аналізу згадуваності країн наведено на рисунку 3.14.

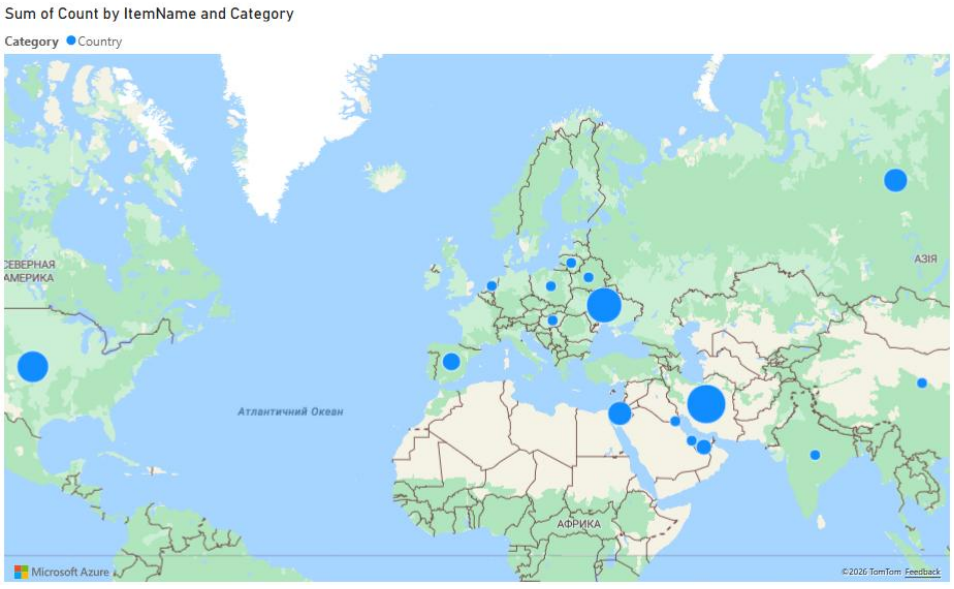


Рисунок 3.14 – Географічна мапа щільності згадуваності країн у новинному потоці

Як видно з наведеної мапи, система дозволяє чітко локалізувати епіцентри медійної уваги. Найбільші візуальні кластери (бульбашки) сконцентровані над територією України (внутрішній порядок денний), а також над США, Іраном, рф та країнами Близького Сходу. Такий географічний розподіл добре корелює з результатами попереднього топік-моделювання (рисунок 3.11), де рубрика «Міжнародні конфлікти» займала перше місце. Дрібніші маркери над країнами Європи (Іспанія, країни Балтії) та Азії (Індія) відображають фонові дипломатичні або економічні новини.

Технічна та аналітична цінність цієї візуалізації полягає у її здатності перетворювати сухий текстовий потік на інструмент геополітичного моніторингу. Завдяки тому, що на стороні C#-клієнта дані з моделі GLiNER були приведені до єдиного називного відмінка (лемматизовані), Power BI не створює дублікатів для маркерів (наприклад, слова «США», «у США», «із США» зводяться до єдиної точки).

Для кінцевого користувача (наприклад, аналітика безпекового сектору) цей інструмент відкриває можливості для глибокого крос-фільтрування. Обравши на мапі конкретну країну (наприклад, Іран), користувач ініціює автоматичну перебудову всіх інших віджетів на дашборді: кругова діаграма може показувати, в яких саме рубриках фігурував Іран, а інструмент дайджестів може миттєво відфільтрувати лише ті зведення, де згадується ця держава. Таким чином, розроблений конвеєр від перехоплення трафіку (mitmproxy) до кінцевої візуалізації доводить свою усебічну функціональну придатність.

Для поглиблення просторового аналізу та переходу від макрорівня держав до мікрорівня конкретних населених пунктів, у системі реалізовано деталізовану візуалізацію згадуваності міст. Цей рівень аналітики активується шляхом перемикання атрибута фільтрації на Category = City. Алгоритми геокодування Power BI обробляють лемматизовані назви міст, видобуті моделлю NER, і проектують їх на карту з точністю до конкретних координат.

Результат геопросторового розподілу інформаційного потоку на рівні міст наведено на рисунку 3.15.

Рисунок 3.15 – Деталізована бульбашкова мапа згадуваності міст у медіа-просторі

Як видно з наведеної мапи, зміна рівня деталізації дозволяє виявити конкретні локальні епіцентри подій всередині раніше ідентифікованих країн-лідерів. Абсолютна більшість найбільших маркерів (бульбашок з найвищим показником Count) прогнозовано сконцентрована на території України. Це об'єктивно відображає специфіку досліджуваного новинного джерела, де висока щільність локальних новин зумовлена висвітленням внутрішнього порядку денного (наприклад, згадки міст у контексті рубрик «Повітряна тривога», «Інфраструктура» або «Суспільство»). Також чітко прослідковуються точкові інформаційні кластери на Близькому Сході та в

країнах Південної Європи, які деталізують загальний міжнародний контекст.

Практична значущість цієї візуалізації полягає у забезпеченні можливостей для тактичного моніторингу. На відміну від згадки цілої країни (яка часто фігурує у загальних дипломатичних новинах), згадка конкретного міста найчастіше є індикатором конкретної фізичної події на місці (надзвичайна ситуація, візит посадовців, інфраструктурні зміни). Це може бути найбільш ефективно використано у хронологічному перегляду локальних подій, наприклад, потерпання міст від обстрілів за певний проміжок часу, де частота згадування міста може корелювати з серйозністю наслідків його руйнування.

Завдяки безшовній інтеграції всіх модулів у середовищі Power BI, цей віджет працює як потужний просторовий фільтр. Аналітик може обрати конкретне місто на мапі (наприклад, Харків чи Одесу) і миттєво відфільтрувати загальний датасет. У результаті інші візуалізації покажуть, у яких саме рубриках фігурувало це місто, а користувач зможе звернутися до бази даних, щоб прослухати згенерований LLM та озвучений TTS аудіо-дайджест подій, що стосуються обраної локації. Це перетворює розроблену систему з новинного агрегатора та класифікатора на повноцінний інструмент для проведення глибокої аналітики.

Для аналізу персоналізованого виміру інформаційного потоку та ідентифікації головних дійових осіб (акторів) медіа-простору, у системі застосовано візуалізацію типу «Хмара слів» (Word Cloud). Цей аналітичний віджет генерується на основі масиву даних, відфільтрованого за ознакою Category = Person. Як вхідні дані використовуються лемматизовані імена та прізвища публічних діячів, попередньо екстраговані з текстів за допомогою моделі GLiNER. Застосований віджет дає змогу оцінити точність класифікації моделлю персоналій та ступінь зведення імен та прізвищ до однієї інфінітивної форми. Результат візуалізації частоти згадувань конкретних осіб наведено на рисунку 3.16.

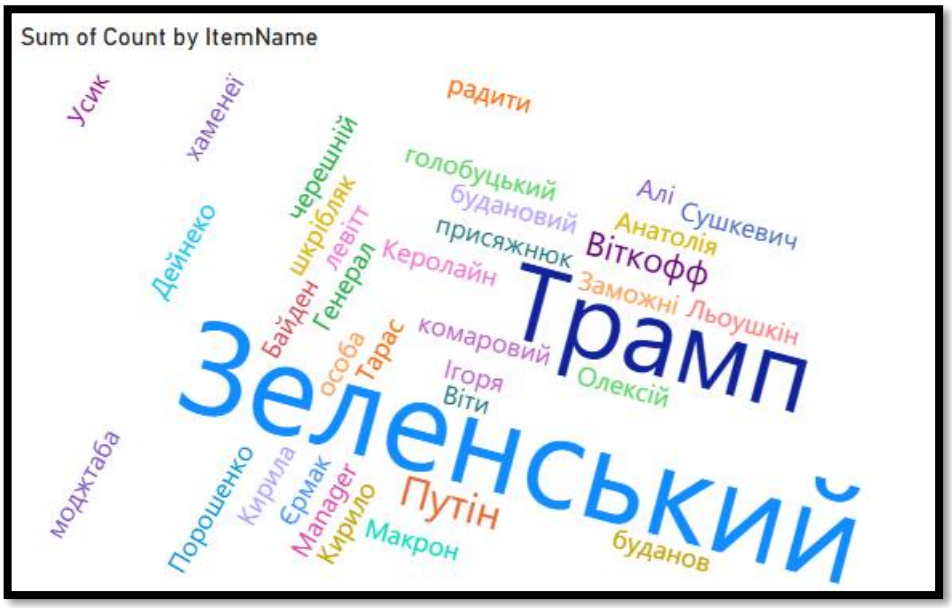


Рисунок 3.16 – Хмара тегів згадуваності політичних та громадських діячів у новинному потоці

Як видно з наведеної візуалізації, розмір кожного текстового елемента є прямо пропорційним до абсолютного показника його згадуваності (Count) у проаналізованому масиві новин. Такий підхід, відомий у обробці природної мови як «Bag of Words» (мішок слів), дозволяє миттєво зчитати ієрархію інформаційної присутності.

На графіку абсолютно домінують прізвища «Зеленський» та «Трамп», що слугує прямим індикатором найвищого суспільного та медійного інтересу до цих фігур у визначений хронологічний період. Другий ешелон за частотою згадувань формують прізвища світових лідерів та ключових посадовців («путін», «Макрон», «Байден», «Єрмак», «Буданов»), що корелює з раніше виявленим домінуванням рубрик «Міжнародні конфлікти» та «Війна в Україні» (рисунок 3.13).

Головна технічна перевага цієї візуалізації полягає в ефективності попереднього препроцесингу на клієнтській C#-частині. Оскільки всі прізвища були програмно зведені до називного відмінка (наприклад, словоформи «Зеленського», «Зеленському» об'єднані в єдиний токен

«Зеленський»), хмара слів відображає абсолютно точну статистику, уникаючи фрагментації даних.

Для фахівців з інформаційної безпеки або PR-аналітиків цей інструмент у середовищі Power BI виконує роль швидкого детектора фокусу уваги. Взаємодіючи з дашбордом, користувач може натиснути на будь-яке прізвище (наприклад, «Макрон») у хмарі слів, після чого система може підсвітити на дашборді топіки, в контексті яких він згадувався (наприклад, «Міжнародна підтримка», а також може відфільтрувати згенеровані LLM аудіо-дайджести, залишаючи лише події, де фігурує обраний політик.

Для комплексного охоплення всіх аспектів медіа-простору, окрім геополітичного та персоналізованого вимірів, розроблена система забезпечує глибокий аналіз корпоративного сектору. З цією метою у середовищі Power BI було створено окрему візуалізацію формату «Хмара слів», яка базується на масиві даних, відфільтрованому за цільовим стовпцем Category = Company. Цей віджет акумулює лемматизовані назви комерційних брендів, підприємств та організацій, видобутих трансформерною моделлю GLiNER.

Результат візуалізації медійної присутності компаній наведено на рисунку 3.17.

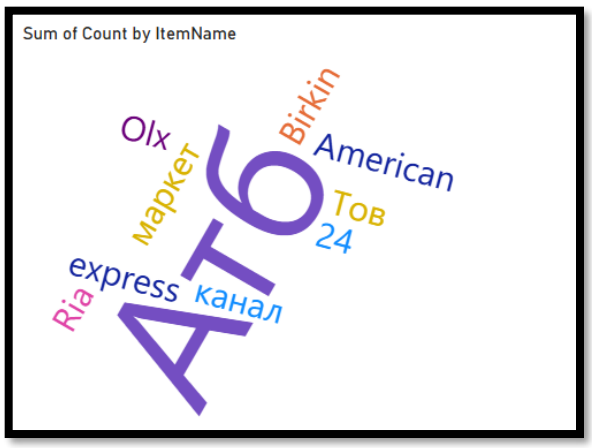


Рисунок 3.17 – Хмара тегів згадуваності комерційних компаній та брендів у інформаційному потоці

Аналіз наведеної візуалізації демонструє специфіку корпоративного інформаційного поля досліджуваного Telegram-каналу та дозволяють зробити низку висновків щодо структури медіа-простору. Абсолютним лідером за частотою згадувань є національна мережа супермаркетів «АТБ». Домінування компаній сектору товарів повсякденного попиту у неспеціалізованих новинах є закономірним, оскільки такі підприємства часто виступають не лише комерційними суб'єктами, але й маркерами соціально-економічної ситуації, гуманітарних ініціатив або інфраструктурних змін у регіонах. Також у досліджуваному текстовому масиві чітко виділяються популярні цифрові маркетплейси та сервіси, наприклад, «Olx», «Ria», що відображає високий рівень інтеграції електронної комерції у повсякденний інформаційний потік. Поява окремих міжнародних брендів чи компаній, наприклад, «American», «Birkin» свідчить про наявність новин міжнародного економічного сектору, повідомлень про імпортні постачання або специфічні ринкові інциденти.

Окремої уваги з інженерної та алгоритмічної точок зору заслуговує наявність у хмарі слів таких лексем, як «маркет», «Тов», «express» та «канал». Це яскраво ілюструє практичні аспекти та складнощі роботи алгоритмів розпізнавання іменованих сутностей (NER) на непідготовлених, текстах із соціальних мереж. На відміну від класичних літературних чи офіційно-ділових текстів, публікації у месенджерах характеризуються високим рівнем неформальної лексики, скорочень та специфічного форматування.

Поява таких сутностей у результатах роботи трансформерної моделі свідчить про те, що нейромережа коректно ідентифікує організаційно-правові форми (ТОВ) або розпізнає поширені частини складених назв компаній (наприклад, «Еко-маркет» може бути розбито на окремі токени через специфіку написання у вихідному пості). Крім того, наявність слова «канал» як іменованої сутності класу «Компанія» підкреслює контекстну залежність моделі: у середовищі Telegram адміністратори часто

використовують фрази на кшталт «спонсор нашої публікації – канал Х», через що штучний інтелект логічно класифікує згаданий канал як комерційну організацію або бренд. Це підтверджує ефективність виявлення рекламних інтеграцій та спонсорських публікацій, які є невід'ємною частиною монетизації популярних медіа-ресурсів. У подальшому ці артефакти можуть бути усунуті шляхом налаштування додаткових фільтрів стоп-слів (post-processing) на рівні бекенд-сервера перед записом до бази даних.

Практична цінність цього аналітичного зрізу може бути особливо високою для фахівців галузі маркетингу, PR-менеджменту, конкурентної розвідки та економічної аналітики. Представлена візуалізація дозволяє автоматизовано вимірювати показник Share of Voice (SoV) – частку голосу конкретних брендів в органічному неспеціалізованому медіа-просторі. Замість ручного моніторингу тисяч повідомлень, аналітик отримує миттєвий зріз інформаційної присутності компаній.

Завдяки наскрізній інтерактивності розробленого дашборду в середовищі Power BI, ця хмара тегів не є статичною картинкою, а слугує повноцінним інструментом глибокого дослідження даних. Бізнес-аналітик має змогу клікнути на конкретну компанію, як-от, «АТБ» і за допомогою автоматичної крос-фільтрації інших візуальних елементів з'ясувати точний контекст її згадування. Зокрема, у поєднанні з результатами роботи Zero-Shot класифікатора, система миттєво покаже, чи фігурував бренд у рубриці «Економіка та бізнес», якщо йдеться про фінансові показники чи відкриття нових філій, у рубриці «Інфраструктура», що може містити інформацію про руйнування чи відновлення об'єктів внаслідок бойових дій, або ж у рубриці «Благодійність та збори», якщо компанія виступила донором волонтерських ініціатив. Таким чином, розроблений NLP-пайплайн у синергії з BI-платформою перетворює хаотичний потік новин на структуровану систему підтримки аналізу та прийняття рішень.

ВИСНОВКИ

У ході дослідження було проведено ґрунтовне проектування та розробку комплексної автоматизованої системи медіа-моніторингу з інтегрованим NLP-пайплайном для аналізу новинних потоків з публічних Telegram-каналів.

Значна увага була приділена порівнянню методів обробки природної мови, аналізу архітектур трансформерних моделей та великих мовних моделей (LLM), аналізу існуючих аналогів агрегаторів новин, а також визначенню оптимальних підходів для реверс-інжинірингу мережевого трафіку та екстракції даних.

Було виявлено, що ефективна обробка гетерогенного неструктурованого потоку новин в умовах автоматизованих систем вимагає використання сучасних моделей штучного інтелекту. Було ідентифіковано та імплементовано ключові алгоритми, які забезпечують семантичну категоризацію, екстракцію фактів та сумаризацію тексту на основі моделей mDeBERTa, GLiNER, Qwen та Silero.

Був спроектований та програмно реалізований підхід, який дозволяє системі автоматично збирати масиви повідомлень, очищувати їх від лексичного шуму, вирішувати проблему неоднозначності контексту через ієрархічне групування за рубриками та синтезувати на їх основі логічно зв'язні аудіо-дайджести.

Були сформовані основні вимоги до аналітичного представлення результатів, що дозволило повноцінно врахувати специфіку агрегації даних, спроектувати нормалізовану реляційну базу даних та забезпечити безшовну інтеграцію видобутих сутностей із середовищем Business Intelligence.

У підсумку, розроблена система перетворює вхідний потік текстових даних у структуровану аналітичну форму, враховуючи вимоги кінцевого користувача до отримання оперативних зведень у форматі дайджестів та візуальної аналітики.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1) GLiNER: Generalist model for named entity recognition using bidirectional transformer / U. Zaratina et al. *arXiv*. 2023. URL: https://arxiv.org/abs/2311.08526 (date of access: 01.02.2026).

2) He P., Gao J., Chen W. DeBERTaV3: Improving DeBERTa using ELECTRA-Style pre-training with gradient-disentangled embedding sharing. *arXiv*. 2023. URL: https://arxiv.org/abs/2111.09543 (date of access: 01.02.2026).

3) Qwen technical report / J. Bai et al. *arXiv*. 2023. URL: https://arxiv.org/abs/2309.16609 (date of access: 02.02.2026).

4) Silero TTS: High-quality text-to-speech models. *Github*. 2026. URL: https://github.com/snakers4/silero-models (date of access: 02.02.2026).

5) Korobov M. Morphological analyzer for russian and ukrainian languages (pymorphy). *Github*. 2026. URL: https://github.com/pymorphy2/pymorphy2 (date of access: 04.02.2026).

6) Power BI documentation. *Microsoft*. 2026. URL: https://learn.microsoft.com/en-us/power-bi/ (date of access: 02.02.2026).

7) FastAPI framework documentation. *Tiangolo*. 2026. URL: https://fastapi.tiangolo.com/ (date of access: 09.02.2026).

8) Jurafsky D., Martin H. J. Speech and language processing. 3rd ed. draft. *Stanford University*. 2023. URL: https://web.stanford.edu/~jurafsky/slp3/ (date of access: 29.01.2026).

9) mitmproxy - an interactive HTTPS proxy. *Official documentation*. 2026. URL: https://docs.mitmproxy.org/stable/ (date of access: 01.02.2026).

10) Історія кафедри - Кафедра штучного інтелекту ХНУРЕ. *Кафедра штучного інтелекту ХНУРЕ*. URL: https://ai.nure.ua/istoriya-kafedri (дата звернення: 02.03.2026).

11) Feedly – the world's most popular RSS and REST based news aggregator. *Official documentation*. 2026. URL: https://docs.feedly.com/ (date of access: 26.01.2026).

12) TGStat – Telegram analytics and channel search. *API documentation*. 2026. URL: https://tgstat.com/ (date of access: 26.01.2026).

13) Yin W., Hay J., Roth D. Benchmarking zero-shot text classification: datasets, evaluation and entailment approach. *arXiv*. 2019. URL: https://arxiv.org/abs/1909.00161 (date of access: 06.04.2026).

14) HuggingFace's transformers: State-of-the-art natural language processing / T. Wolf et al. *arXiv*. 2019. URL: https://arxiv.org/abs/1910.03771 (date of access: 11.04.2026).

15) Hugging Face documentation: Token classification. *Hugging Face*. 2026. URL: https://huggingface.co/docs/transformers/tasks/token_classification (date of access: 12.04.2026).

16) Less annotating, more classifying: Addressing the data scarcity bottleneck of text classification with NLI, zero-shot learning, and LLMs / M. Laurer et al. *arXiv*. 2024. URL: https://arxiv.org/abs/2212.10471 (date of access: 20.04.2026).

17) XNLI: Evaluating cross-lingual sentence representations / A. Conneau et al. *arXiv*. 2018. URL: https://arxiv.org/abs/1809.05053 (date of access: 21.04.2026).