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)

Автоматизація створення відео
на основі короткого змісту тексту з веб-сайтів
(тема)

Виконав:
здобувач другого року навчання,
групи СШІм-23-1
Ігор Шатило
(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 доцент Лариса Чала
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Олег ЗОЛОТУХІН
(підпис) (власне ім'я, прізвище)

2025 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 Шатило Ігорю Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Автоматизація створення відео на основі короткого змісту тексту з веб-сайтів

затверджена наказом університету від 21 квітня 20 25 р. № 295Ст

2. Термін подання студентом роботи до екзаменаційної комісії 03 червня 20 25 р.

3. Вихідні дані до роботи науково-технічні публікації з питань обробки природньої мови, реферування текстів, класифікації текстів, дані Інтернет-джерел щодо сучасних підходів до парсингу веб-сторінок, технік семантичного аналізу тексту, генерації коротких змістів та автоматичного створення відеоконтенту, документація до бібліотек для машинного навчання Transformers, Torch, Hugging Face Datasets, а також моделей BERT, RoBERTa, XLM-RoBERTa, BART та MusicGen, документація до API сторонніх сервісів, документація до фреймворку Laravel та Filament, документація до мови програмування Python та бібліотек Flask, Celery, MoviePy, Playwright, Newspaper4k

4. Перелік питань, що потрібно опрацювати в роботі

1) Аналіз предметної галузі та постановка задачі

2) Вибір методів та інструментів для вирішення задачі

3) Проєктування програмного продукту

4) Програмна реалізація та експериментальні дослідження

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	21.04.2025	виконано
2	Аналіз та вибір методів щодо вирішення	21.04.2025-22.04.2025	виконано
	поставленої задачі		
3	Постановка задачі	22.04.2025	виконано
4	Дослідження методів збору текстових даних з	22.04.2025	виконано
	веб-сайтів		
5	Дослідження алгоритмів NLP для текстового	23.04.2025-24.04.2025	виконано
	аналізу та реферування		
6	Реалізація модуля парсингу веб-контенту	25.04.2025-27.04.2025	виконано
7	Реалізація модуля аналізу тексту та реферування	28.04.2025-01.05.2025	виконано
8	Дослідження та розробка алгоритмів підбору	02.05.2025-04.05.2025	виконано
	відео та аудіо контенту		
9	Реалізація модуля синтезу голосу та генерації	05.05.2025-09.05.2025	виконано
	відео		
10	Розробка веб-додатку	10.05.2025-20.05.2025	виконано
11	Написання пояснювальної записки	20.05.2025-25.05.2025	виконано
12	Попередній захист	26.05.2025	виконано
13	Рецензування	29.05.2025	виконано
14	Захист перед ЕК	03.06.2025	

Дата видачі завдання 21 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доцент Лариса Чала
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 157 с., 66 рис., 4 дод., 69 джерел.

ГЕНЕРАЦІЯ ВІДЕО, ГЛИБОКІ НЕЙРОННІ МЕРЕЖІ, ОБРОБКА ПРИРОДНОЇ МОВИ, ОЗВУЧУВАННЯ ТЕКСТУ, ПАРСИНГ ВЕБ-СТОРІНОК, УЗАГАЛЬНЕННЯ ТЕКСТУ, ШТУЧНИЙ ІНТЕЛЕКТ.

Об’єкт дослідження – процеси автоматизованого створення відеофайлів на основі аналізу текстового контенту веб-сайтів.

Предмет дослідження – методи аналізу текстового контенту веб-сайтів та автоматичної генерації відеофайлів на основі їх змісту.

Мета роботи – розроблення, аналіз та програмна реалізація автоматизованої технології, що дозволяє перетворювати текстову інформацію у відеоформат з використанням методів обробки природної мови та нейронних мереж.

Методи дослідження – методи обробки тексту та узагальнення його змісту; методи синтезу мультимедійного контенту та мультимедійного супроводу; методи нейромережевого моделювання.

Практична частина включає експерименти з парсингом текстових даних, NLP-аналізом, підбором відповідного відеоконтенту та створенням відео за допомогою бібліотек Python.

У роботі розглянуто основні етапи автоматизації створення відео: вилучення тексту з веб-сайтів, його аналіз, узагальнення змісту, підбір релевантного аудіо- та відеоконтенту, а також формування фінального відеофайлу. Окремо досліджено можливості сучасних нейронних мереж для обробки тексту. Результати демонструють ефективність підходу та перспективи його використання у різних сферах.

ABSTRACT

Master’s thesis contains: 157 pp., 66 fig., 4 ann., 69 references.

ARTIFICIAL INTELLIGENCE, DEEP NEURAL NETWORKS, NATURAL LANGUAGE PROCESSING, SPEECH SYNTHESIS, TEXT SUMMARIZATION, VIDEO GENERATION, WEB PAGES PARSING.

The object of the research is the processes of automated video generation based on the analysis of textual content from websites.

The subject of the research is the methods for analyzing textual website content and automatically generating video files based on their meaning.

The aim of the work is to develop, analyze, and implement an automated technology that converts textual information into video format using natural language processing techniques and neural networks.

The research methods include methods of text analysis and summarization; methods for generating multimedia content and audiovisual support; and neural network modeling techniques.

The practical part involves experiments with web page parsing, NLP-based text analysis, selection of relevant video content, and the generation of videos using Python libraries.

The thesis covers the main stages of the video automation process: extracting text from websites, analyzing and summarizing its content, selecting appropriate audio and video materials, and generating the final video file. The work also investigates the capabilities of modern neural networks for advanced text processing. The results confirm the effectiveness of the proposed approach and its potential for application in journalism, education, marketing, and other fields.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Абстрактивне реферування – метод автоматичного узагальнення змісту, при якому генерується новий текст, що відрізняється від початкового, але передає його основну суть;

Аналіз сентименту – процес визначення емоційного забарвлення тексту (позитивне, негативне, нейтральне);

БД – база даних;

Веб-додаток – програмний інтерфейс, що працює через веб-браузер;

Веб-парсинг – автоматизований процес отримання текстових або структурованих даних з HTML-коду веб-сторінок;

Екстрактивне реферування – підхід до скорочення тексту шляхом вибору найбільш інформативних речень або фрагментів з наявного тексту без змін у їх структурі;

Промпт (Prompt) – текстовий запит або опис, який подається на вхід генеративній моделі для створення бажаного результату (музики, мовлення, відео тощо);

СУБД – система управління базами даних;

Фреймворк – інфраструктура програмних рішень, що полегшує розробку складних систем;

ШНМ – штучна нейронна мережа;

API – Application Programming Interface – набір правил та протоколів, що дозволяють різним програмам взаємодіяти;

BERT – Bidirectional Encoder Representations from Transformers – модель трансформера для обробки природної мови;

gTTS – Google Text-to-Speech – сервіс синтезу мовлення від Google;

HTML – HyperText Markup Language – мова розмітки гіпертексту;

IDE – Integrated Development Environment – інтегроване середовище розробки;

JS – JavaScript – мова програмування для створення інтерактивних веб-сторінок;

JSON – JavaScript Object Notation – текстовий формат обміну даними;

Laravel – PHP-фреймворк для створення веб-сайтів;

LLaMA – Large Language Model Meta AI – велика мовна модель, розроблена компанією Meta;

LLM – Large Language Model – велика мовна модель;

LSTM – Long Short-Term Memory – архітектура рекурентної нейронної мережі для роботи з послідовностями;

MusicGen – нейромережевий генератор музики від Meta AI;

MVC – Model, View, Controller – архітектурний патерн для структурування ПЗ;

MySQL – система управління реляційними базами даних;

Nginx – веб-сервер;

NER – Named Entity Recognition – виявлення іменованих сутностей;

NLP – Natural Language Processing – обробка природної мови;

ORM – Object-Relational Mapping – спосіб взаємодії з базою даних через об’єкти;

PHP – мова програмування для створення серверної частини (бекенду) сайту;

React – бібліотека JavaScript для створення інтерактивних інтерфейсів користувача;

Redis – система керування базою даних типу ключ-значення, яка зберігає дані у пам’яті;

RNN – Recurrent Neural Network – рекурентна нейронна мережа;

StyleTTS2 – сучасна дифузійна модель синтезу мовлення з керуванням стилем;

TTS – Text-to-Speech – синтез мовлення з тексту.

ВСТУП

Сучасний етап розвитку інформаційного суспільства характеризується стрімким зростанням обсягів цифрових даних, серед яких особливе місце займає текстовий контент. Щодня в мережі Інтернет публікуються тисячі новин, статей, блогів та інших матеріалів, що потребують швидкого аналізу, узагальнення та ефективної подачі для широкої аудиторії. У такому середовищі традиційні способи сприйняття текстової інформації часто виявляються недостатньо зручними або повільними, що викликає потребу в нових підходах до перетворення текстів у більш динамічні й зрозумілі формати.

Однією з найбільш зручних форм подачі інформації на сьогоднішній день є відео [1]. Саме відео забезпечує емоційну залученість, швидке сприйняття, а також адаптацію контенту до різноманітної аудиторії – незалежно від рівня підготовки або доступності інформації. Перетворення тексту у відеоформат дозволяє не тільки покращити донесення змісту, а й підвищити охоплення, збільшити тривалість перегляду та посилити вплив на кінцевого користувача.

Традиційне створення відео вимагає участі команди фахівців – сценариста, диктора, відеомонтажера, дизайнера, спеціаліста зі звуку – що робить процес затратним у часі, складним в реалізації та дорогим. У зв’язку з цим дедалі більшої популярності набувають автоматизовані підходи до створення відеоконтенту на основі текстових матеріалів. Сучасні досягнення у сфері штучного інтелекту, зокрема обробки природної мови, генеративних моделей, синтезу мовлення та мультимедійного пошуку, відкривають можливість реалізації таких систем, здатних створювати повноцінне відео з озвученням, візуальними елементами та фоновою музикою.

Актуальність теми цієї роботи зумовлена зростаючою потребою в інструментах автоматичного створення відео на основі текстового

контенту з веб-сторінок. Такі інструменти можуть застосовуватися в журналістиці, освіті, маркетингу, соціальних мережах, електронній комерції та багатьох інших сферах. Вони дозволяють спростити процес створення медіаконтенту, зменшити витрати та підвищити ефективність комунікації з кінцевим користувачем або споживачем контенту.

Об’єктом дослідження є процес автоматичного перетворення тексту у відеоформат на основі аналізу текстового контенту веб-сайтів. Предметом дослідження є методи обробки, аналізу, класифікації текстів, генерації короткого змісту, озвучення, підбору мультимедійного контенту та інтеграції всіх компонентів у єдину систему.

Метою роботи є розробка та програмна реалізація технології, яка дозволяє автоматично створювати відеофайли на основі короткого змісту тексту, отриманого з веб-джерел, із використанням сучасних підходів штучного інтелекту, мовної обробки, підбору та генерації мультимедійного супроводу.

Наукова новизна роботи полягає в комплексному підході до автоматизованого створення відео за текстом з веб-джерел із використанням широкого спектра інтелектуальних технологій. Практична цінність розробки полягає в можливості використання створеної системи для автоматичного перетворення текстового контенту у відео в реальних умовах. Це дозволяє зменшити витрати на виробництво відео, підвищити ефективність створення контенту, а також зробити цифрову інформацію доступнішою для ширшого кола користувачів.

Запропонована в даній роботі система поєднує актуальні наукові розробки з практичними завданнями сучасного цифрового простору, що визначає її важливість як з теоретичного, так і з прикладного погляду. Застосування комплексного підходу дозволяє вирішити одразу кілька важливих завдань: підвищити швидкість створення відеоконтенту, забезпечити його персоналізацію, адаптувати його до інформаційних запитів аудиторії та зменшити потребу в ручному втручанні.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис розв’язуваної проблеми

У сучасному цифровому середовищі обсяг текстової інформації, яка щоденно генерується та публікується в інтернеті, зростає у геометричній прогресії. Новинні портали, блоги, наукові публікації, огляди, аналітичні матеріали – усе це створює значне інформаційне навантаження на користувача. При цьому, споживачі контенту дедалі частіше надають перевагу візуальним та мультимедійним форматам подачі інформації, зокрема відео, яке значно зручніше сприймати, особливо в умовах обмеженого часу [2]. Проте створення відеоконтенту вимагає значних людських і часових ресурсів, що не завжди є доцільним або можливим, особливо для невеликих організацій, приватних авторів або автоматизованих систем публікацій.

Таким чином, виникає потреба в автоматизації процесу створення відео на основі текстового контенту, який уже наявний у відкритому доступі, зокрема на веб-сайтах. Основна проблема полягає в тому, щоб із великого обсягу неструктурованого тексту виокремити ключову інформацію, перетворити її на короткий зміст, доповнити релевантним відео- та аудіоконтентом і автоматично сформувати повноцінне відео. При цьому важливо зберегти смислову цілісність оригінального тексту, дотриматись логіки викладу, а також створити відео з привабливою подачею, що задовольнить вимоги кінцевого користувача.

Складність проблеми полягає в багатоступеневому характері задачі. Насамперед, потрібно коректно отримувати текстовий вміст з веб-сторінок, які часто мають динамічну структуру та велику кількість додаткових елементів, таких як реклама, меню, посилання, кнопки, скрипти тощо. Потім необхідно виконати попередню обробку тексту, видалити зайві фрагменти, провести аналіз мови, структури та смислового

наповнення тексту. Після цього слідує етап автоматичного реферування, що включає виокремлення ключових тез або побудову узагальненого тексту, придатного для сценарію відео.

Наступною складовою є пошук або генерація мультимедійного контенту, який відповідає змісту тексту. Це можуть бути відеофрагменти, зображення або фонові мелодії, які візуалізують зміст кожної сцени. До цього додається ще й етап синтезу голосу, де текст повинен бути озвучений природно звучним голосом, наближеним до людського. І, нарешті, усі ці компоненти повинні бути об’єднані в єдине відео, яке не лише містить інформацію з оригінального тексту, але й подає її у зручній та привабливій формі.

Окремо варто зазначити складність забезпечення інтеграції всіх модулів у єдину програмну систему. Таке рішення має включати модулі парсингу, обробки тексту, генерації відео, синтезу мовлення, а також інтерфейс взаємодії з користувачем – веб-додаток. Складність полягає не лише у розробці кожного з окремих модулів, а й у їхньому злагодженому функціонуванні, обміні даними та масштабованості всієї системи.

Ще одним важливим аспектом проблеми є потреба в універсальності розробленої системи. Вона повинна працювати з різноманітними джерелами контенту (новинні сайти, блоги, тематичні статті), підтримувати багатомовність, адаптуватися до різних форматів текстів і при цьому залишатися стабільною та ефективною. Така система може мати широкий спектр застосувань: від автоматичного створення новинного контенту, освітніх відео, маркетингових роликів до допоміжних сервісів для людей з обмеженими можливостями.

Загалом, проблема, яку вирішує дана кваліфікаційна робота, є комплексною, міждисциплінарною і вимагає залучення сучасних методів з галузей штучного інтелекту, обробки природної мови, генерації мультимедійного контенту та веб-розробки. Її успішне розв’язання дозволить створити ефективну технологію для перетворення текстового

контенту у відеоформат без залучення людини на кожному з етапів, що є надзвичайно актуальним у сучасних умовах цифрової трансформації.

1.2 Актуальність дослідження

Сучасні цифрові технології суттєво змінюють способи створення, обробки та споживання інформації. Особливо це стосується форм представлення контенту: якщо ще донедавна основним форматом залишався текст, то сьогодні все більше платформ переходять до відеоформатів, орієнтуючись на тренди споживання інформації через візуальні та аудіо канали [3]. Такі зміни зумовлені не лише зручністю для користувачів, а й вищою залученістю, яку забезпечує відео. За даними аналітики, відеоконтент значно краще утримує увагу аудиторії, сприяє швидшому запам’ятовуванню інформації та підвищує ефективність комунікації.

На тлі зростання популярності відеоформатів виникає актуальне завдання – створення таких засобів, які дозволяють автоматизувати перетворення текстових матеріалів у відео без необхідності залучення дизайнерів, дикторів та монтажерів. Таке рішення є особливо важливим для організацій з великим потоком текстової інформації, а саме новинних агенцій, освітніх платформ, технічної підтримки, блогів і навіть електронної комерції, де текст є основним джерелом первинного змісту.

Ще одним підтвердженням актуальності є зростаюча кількість користувачів з обмеженнями в читанні – зокрема, через зорові порушення або дислексію. Для таких груп населення автоматично згенерований відеоконтент із озвученням стає способом доступу до інформації, яка інакше була б для них недоступною або складною у сприйнятті. Таким чином, створення таких технологій відповідає концепції інклюзивного дизайну та цифрової доступності.

У науково-технічному плані важливість теми зумовлена також зростаючим рівнем зрілості інструментів для обробки природної мови, синтезу мовлення, генерації зображень і відео, що базуються на сучасних нейромережевих моделях. Інструменти, які раніше вважалися експериментальними, нині демонструють стабільну продуктивність у промислових додатках. Це відкриває можливість створювати комплексні, модульні системи, що здатні обробляти текст, створювати озвучення та компонувати відеоряд автоматично, без участі людини.

Крім того, у світовій IT-індустрії спостерігається активний попит на генеративні системи (такі як GPT, DALL·E, Synthesia, Pictory тощо), які дозволяють автоматизувати процеси створення контенту [4]. Тема, що досліджується в межах цієї кваліфікаційної роботи, органічно вписується в ці тенденції, але має свою унікальність: поєднання відкритих бібліотек та розробку цілісного веб-орієнтованого рішення робить її перспективною як у науковому, так і в прикладному плані.

1.3 Аналіз джерел інформації та вимог до контенту

У процесі автоматизованої генерації відео ключову роль відіграє якість вхідного текстового матеріалу. Від того, наскільки структурованим, зрозумілим, логічно послідовним та змістовно повним є вихідний текст, залежить не лише якість узагальнення, а й доречність підібраного відеоряду, правильність озвучення та загальне враження від готового відеопродукту. Тому важливо ще на початковому етапі системно проаналізувати джерела інформації, з якими буде працювати система, а також сформулювати вимоги до таких текстів.

У межах дослідження передбачається використання відкритих текстів з веб-сайтів – це можуть бути новинні портали, тематичні блоги, науково-популярні статті, огляди технологій, інструкції, аналітичні матеріали тощо. Основна мета – мати доступ до неструктурованих або

слабо структурованих статей, які містять природну мову, а не шаблонні фрагменти чи рекламний контент. Наприклад, у новинній статті про запуск супутника можуть бути чітко виділені заголовок, вступ, основна частина й заключення. У таких текстах легше виділити ключові тези, зрозуміти емоційне навантаження й створити узагальнення. На рисунку 1.1 зображено приклад новини, яка є придатною для подальшої обробки та аналізу [5].

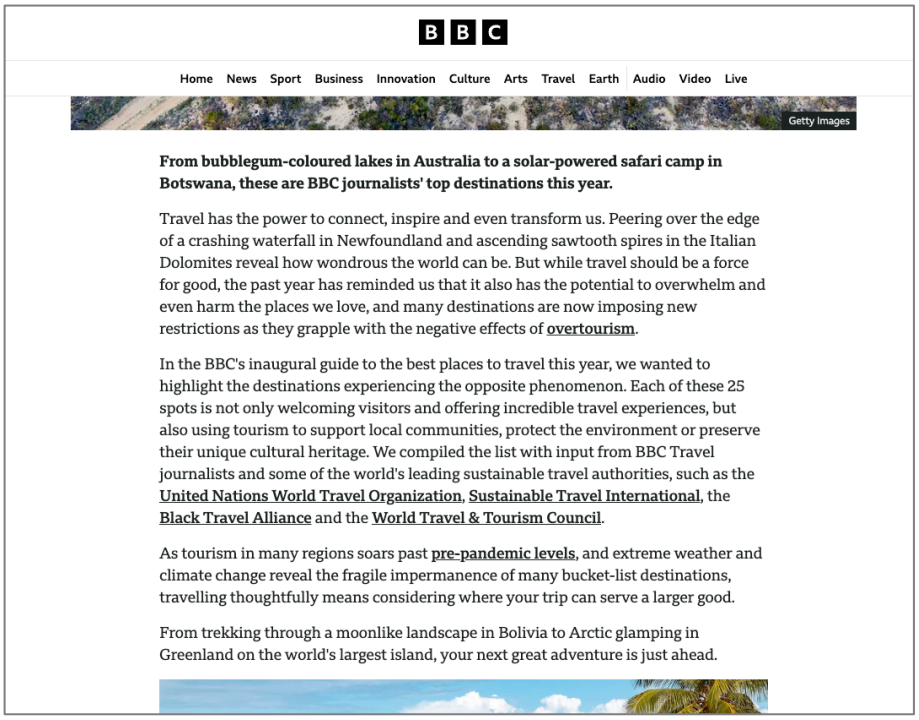


Рисунок 1.1 – Приклад новинної статті, придатної для обробки та узагальнення

Водночас слід враховувати, що не кожна веб-сторінка є придатною для автоматизованого перетворення на відео. Значна частина сторінок має низьку інформативну щільність, наприклад, каталоги товарів, головні сторінки, FAQ-секції або сторінки з великою кількістю рекламних елементів. У таких випадках система може витягти лише часткові фрагменти тексту або навіть хибно визначити головний зміст. Тому важливою умовою для майбутнього модуля парсингу є наявність

механізму фільтрації, який дозволяє визначити тип сторінки та оцінити релевантність її змісту. Наприклад, у блозі, що містить переважно цитати, список товарів або описи послуг, інформація буде непридатною для якісного узагальнення. На рисунку 1.2 зображено приклад Інтернет-магазину, який має нерелевантну інформацію для подальшої генерації відео [6].

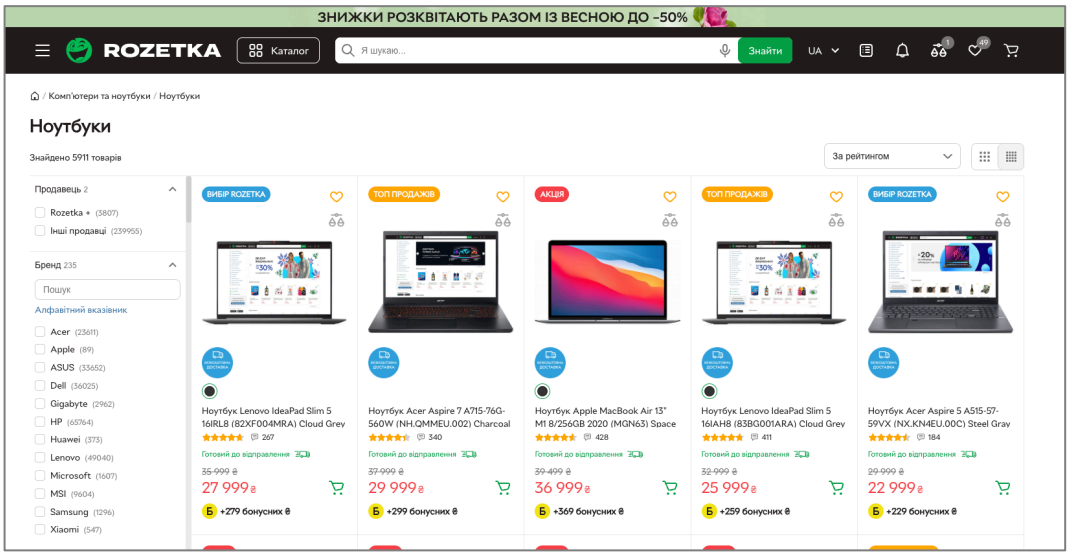


Рисунок 1.2 – Приклад веб-сторінки з низькою інформативністю для генерації відео

До основних вимог до контенту, який має бути оброблений системою, можна віднести такі критерії: наявність зв’язного основного тексту (мінімум 200–400 слів), чітка тематика, логічна структура, відсутність надмірної кількості розгалужених посилань, HTML-розмітки або технічного шуму. Тексти мають бути написані літературною мовою, без надмірної кількості спеціалізованих термінів, абревіатур чи сленгу. Це дозволить системі коректно проводити синтаксичний аналіз та узагальнення змісту. У разі, якщо джерело містить декілька мов або частини тексту розміщені у форматі зображень, система повинна або їх ігнорувати, або повідомляти користувача про неможливість обробки.

Ще одним важливим фактором є врахування семантичного контексту. Наприклад, статті з новинних сайтів, присвячені подіям або суспільним явищам, часто містять імена, дати, географічні назви та числові показники. Ці дані мають бути збережені під час реферування та озвучення, оскільки саме вони несуть основне смислове навантаження. На рисунку 1.3 показано приклад фрагмента тексту з сайту університету [7], який містить важливі факти, але не надто довгий, що робить його ідеальним кандидатом для генерації короткого відеоролику.

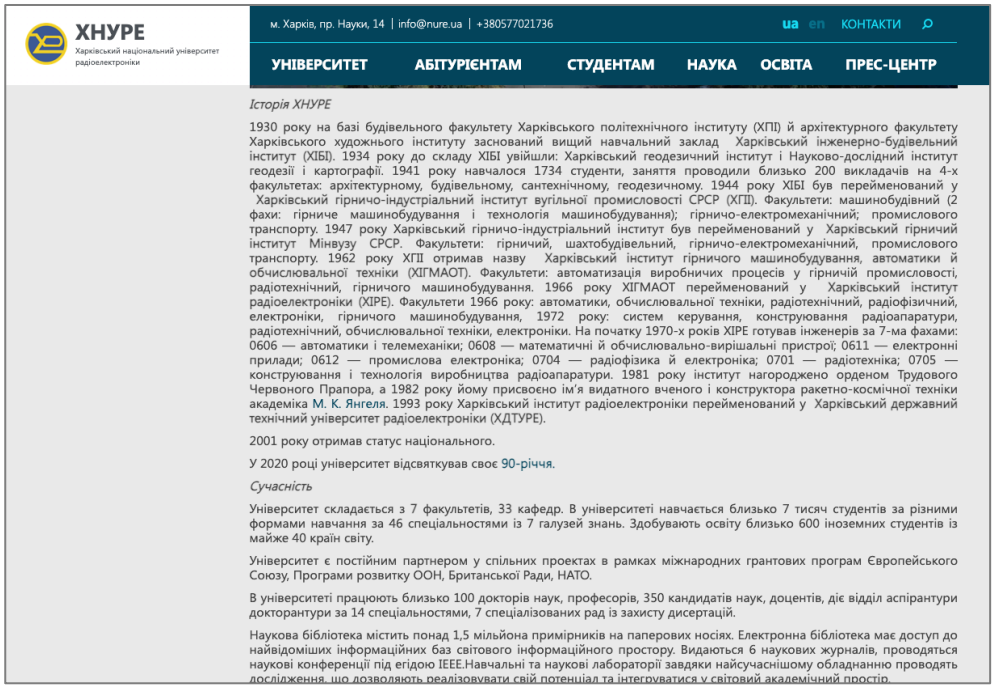


Рисунок 1.3 – Фрагмент веб-сторінки з насиченим фактажем, придатний для перетворення у відео

У загальному випадку система повинна бути достатньо універсальною, щоб працювати з різними форматами текстів – від оглядів техніки до кулінарних рецептів. Однак на етапі дослідження буде обрано кілька типових класів контенту, на яких відпрацьовується основна логіка: новини, блоги, інструкції. Це дає змогу зосередитися на оптимізації обробки найпоширеніших сценаріїв використання. У майбутньому

можлива адаптація до інших типів контенту шляхом донавчання моделей або введення користувацьких шаблонів обробки.

Таким чином, на основі аналізу джерел та їх особливостей сформульовані вимоги до контенту, який буде оброблятись у системі. Це дозволяє ще на етапі постановки задачі передбачити типові труднощі, пов’язані з якістю текстів, та врахувати їх при розробці архітектури системи.

1.4 Оцінка автоматизованості процесу створення відео

Процес створення відео на основі текстового контенту включає в себе низку послідовних етапів, кожен з яких має свій рівень складності та ступінь придатності до автоматизації. В умовах стрімкого розвитку технологій штучного інтелекту дедалі більше кроків, які раніше виконувалися вручну, можуть бути делеговані інтелектуальним системам. Водночас є й ті складові, які залишаються технологічно складними або ресурсомісткими для повної автоматизації.

Першим етапом є вилучення тексту з веб-сторінок. Сучасні парсери здатні з високою точністю знаходити та виокремлювати основний контент сторінки, і навіть деякі сервіси, такі як Pictory, вже підтримують введення URL-адрес для автоматичної обробки. Однак якість цього етапу залишається залежною від структури HTML-документу та складності динамічного контенту. Веб-сторінки, що формуються за допомогою JavaScript або містять великі обсяги реклами й непотрібних елементів, можуть бути неправильно оброблені. Крім того, відсутність явних структурних маркерів (наприклад, класів або тегів статті) ускладнює автоматичне виділення змістовного ядра тексту. Отже, хоча парсинг на перший погляд є добре автоматизованим, на практиці він потребує складної логіки очищення та семантичної перевірки.

Наступним етапом є аналіз та узагальнення тексту. Саме тут відбувається перехід від «сирого» тексту до стиснутого, зв’язного та змістовного викладу основних ідей. Цей етап є одним із найважливіших у контексті створення сценарію відео. Завдяки сучасним трансформерним моделям (таким як BART, T5, PEGASUS, GPT) абстрактивне реферування може бути виконано повністю автоматично з високою точністю. Однак ефективність моделей залежить від якості вхідного тексту, мовної специфіки, обсягу даних і довжини контексту. Окрім того, навіть найсучасніші LLM не завжди формують логічно ідеальний переказ – іноді трапляються повторення, пропущені деталі або надто узагальнені формулювання. Водночас цей етап має високий потенціал до автоматизації, особливо при використанні моделей, попередньо налаштованих на відповідну предметну галузь.

Окремим важливим блоком є підбір візуального супроводу. На сьогодні цей процес є лише частково автоматизованим. Більшість систем використовують ключові слова або фрази з тексту для пошуку відповідних відеофрагментів або зображень у стокових бібліотеках. Такий підхід дозволяє формально поєднати зміст і картинку, однак він не враховує контекст, емоційний настрій або стиль сценарію. Крім того, результати пошуку часто потребують ручної перевірки, оскільки вибрані зображення можуть бути неінформативними або не відповідати культурному чи соціальному фону тексту. Таким чином, хоча базовий рівень автоматизації існує, цей етап все ще має низьку якість повної автономності.

Також варто розглянути озвучення узагальненого тексту. Завдяки розвитку TTS (Text-to-Speech) технологій цей процес сьогодні можна вважати добре автоматизованим. Системи, побудовані на основі Tacotron, FastSpeech або WaveNet, здатні генерувати синтетичне мовлення з високим рівнем природності, інтонації та тембру. Сервіси на зразок Google Cloud TTS, Amazon Polly або ElevenLabs дозволяють генерувати голоси різної статі, мови, акценту, з регулюванням швидкості, пауз і емоційної

виразності. Це дозволяє максимально персоналізувати подачу. Основним обмеженням тут є доступність певних голосів лише в платних тарифах або обмеження на кількість запитів у безкоштовних версіях. Проте з технологічного боку цей етап є майже повністю автоматизованим.

Завершальним етапом є генерація фінального відео – поєднання усіх попередніх компонентів у єдиний відеофайл. Технічно це завдання реалізується через програмні засоби типу ffmpeg, moviepy або спеціалізовані редактори, які можуть з'єднувати відеоряд, звук, титри та фонову музику. Рівень автоматизації тут залежить від того, наскільки гнучко налаштовано логіку компонування сцен. Наприклад, якщо система вміє самостійно визначати тривалість сцени, позицію тексту, фонові ефекти – це значно підвищує якість фінального відео. У протилежному випадку користувач змушений вносити правки вручну, що знижує ефективність автоматизації.

Таким чином, загальна оцінка автоматизованості процесу створення відео дозволяє зробити висновок, що повна автономія є можливою, але потребує глибокої інтеграції різних технологічних компонентів. Найбільш автоматизованими є етапи озвучення та узагальнення тексту, тоді як найменш автоматизованими залишаються підбір візуального контенту та попередній аналіз сторінки. У даній роботі основна увага буде зосереджена на реалізації замкненого ланцюга обробки – від посилання на статтю до повноцінного відео – з використанням гнучких модулів і сучасних моделей ШІ, що дозволить максимально скоротити участь людини в процесі.

1.5 Огляд існуючих аналогічних систем та сервісів для автоматичної генерації відео

Стрімкий розвиток технологій у сфері штучного інтелекту, комп'ютерного зору та обробки природної мови відкрив нові можливості для автоматизації творчих процесів. Одним із яскравих прикладів такого

прогресу є поява систем, що дозволяють генерувати відео з текстового контенту без участі відеоредакторів, операторів або дикторів. Раніше створення відеоролика було тривалим процесом, що вимагав як технічних, так і креативних навичок, а також використання спеціалізованих інструментів. Сьогодні ж цей процес дедалі частіше делегується інтелектуальним системам, які здатні автоматично аналізувати текст, структурувати його зміст, озвучити, підібрати візуальні матеріали, і навіть змонтувати відео з анімаціями, музикою та субтитрами [8].

На сучасному ринку вже представлено чимало таких сервісів. Кожен із них має свої переваги, обмеження та вузьку спеціалізацію. Одні орієнтовані на створення коротких маркетингових відео, інші – на освітній контент або корпоративні презентації. Деякі системи вміють працювати з віртуальними ведучими (аватарами), інші – лише з текстом і фоновими відеоматеріалами.

Розглянемо найбільш показові сервіси, які мають різні підходи до створення відео: Pictory, Invideo, Synthesia, та Diffbot – сервіс, який спрямований на аналіз веб-сторінок.

1.5.1 Pictory

Pictory – це хмарна платформа, яка надає користувачам можливість автоматично створювати відео з тексту або статей [9]. Інтерфейс цього сервісу, зображений на рисунку 1.4, орієнтований на максимально просту взаємодію, що дозволяє навіть користувачам без досвіду в дизайні чи відеомонтажі створювати повноцінні відеоролики. Головна ідея, що лежить в основі сервісу, полягає у перетворенні тексту на структуру, придатну для візуалізації.

Однією з особливостей Pictory є підтримка введення як сирого тексту, так і посилань на веб-сторінки. У другому випадку система автоматично здійснює парсинг вмісту сторінки, витягує основний текст та

очищає його від зайвих елементів, таких як реклама або навігація. Далі застосовується алгоритм скорочення тексту, який дозволяє зосередитися на основних тезах. На цьому етапі в роботу включаються моделі обробки природної мови, які виділяють ключові речення та формують структуру для сценарію. Система не просто бере перші абзаци тексту, а здійснює смисловий відбір, що підвищує інформативність відео.

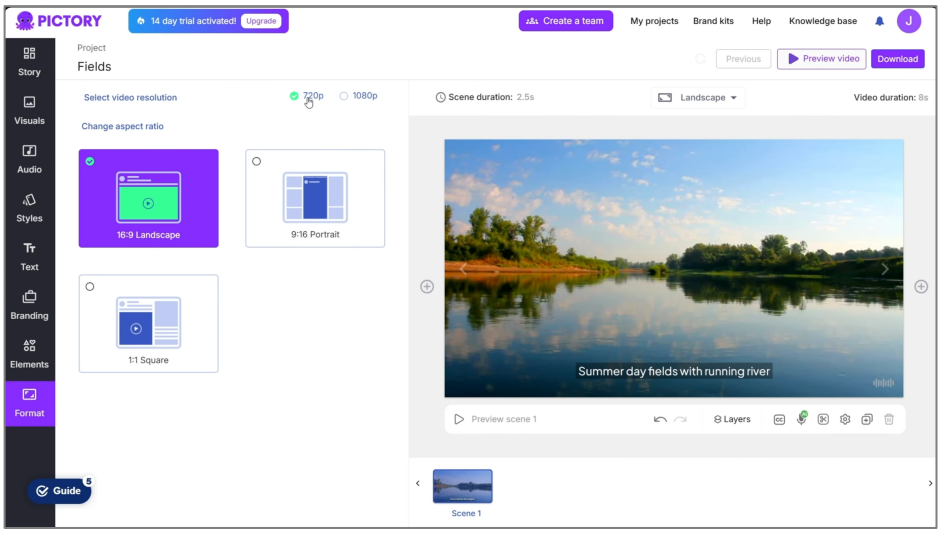


Рисунок 1.4 – Скріншот сервісу Pictory

Після формування сценарію користувач має можливість переглянути та відредагувати його, при необхідності змінити послідовність слайдів, текст, стиль подачі чи навіть додати власні візуальні матеріали. Візуальне оформлення сцени формується автоматично – система підбирає відео або зображення з власної великої бібліотеки, орієнтуючись на ключові слова з кожної частини сценарію. Озвучення також генерується автоматично: користувач обирає голос, стать диктора, мову, акцент. У результаті формується цілісний відеопродукт, який можна завантажити або одразу опублікувати.

Варто зазначити, що хоча Pictory справляється із завданням генерації відео доволі якісно, його алгоритми залишаються закритими, а користувач має обмежені можливості впливу на внутрішню логіку формування змісту.

Сервіс не дозволяє підключення власних моделей, проте має API для інтеграції в сторонні системи та забезпечує підтримку налаштування відео. Загалом, Pictory є одним із лідерів у своєму класі й широко використовується в маркетингу, блогінгу та онлайн-навчанні.

1.5.2 Invideo

Invideo – це онлайн-платформа, яка також дозволяє створювати відео на основі тексту за допомогою штучного інтелекту [10]. Основною метою розробників є спрощення процесу відеопродакшену для малого та середнього бізнесу, маркетологів і творців контенту. Від самого початку Invideo позиціював себе як інструмент «all-in-one» – тобто, такий, що поєднує в собі як генеративні можливості, так і повноцінний редактор для ручного доопрацювання результату. На рисунку 1.5 зображено візуальний інтерфейс сервісу Invideo.

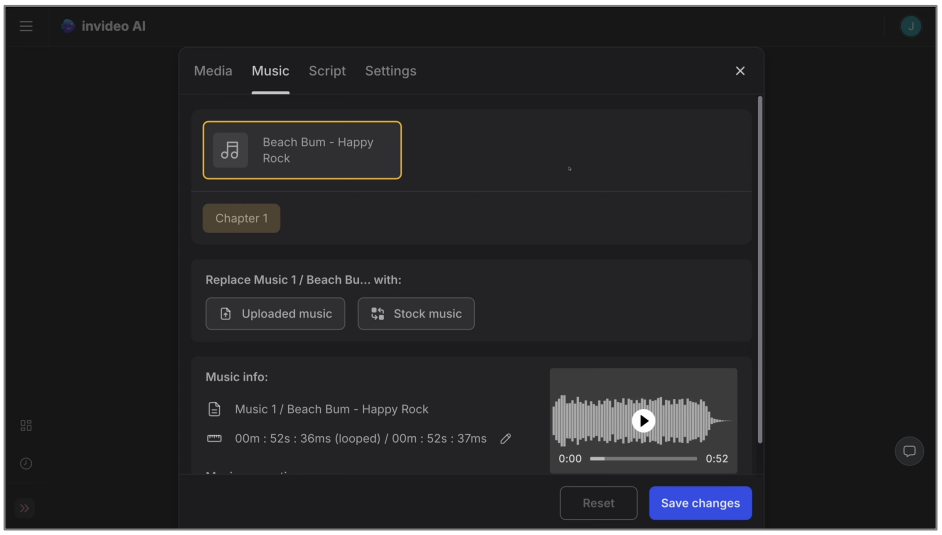


Рисунок 1.5 – Скріншот сервісу Invideo

На відміну від деяких аналогів, Invideo надає користувачу широкий контроль над кожним етапом створення відео. Користувач може ввести

опис відео в довільній формі, після чого система за допомогою генеративних алгоритмів формує сценарій, який потім автоматично трансформується у відео. Система розбиває зміст на окремі слайди або сцени, і для кожної частини обирає відповідний відеофрагмент, графіку або анімацію з вбудованої бібліотеки. Весь процес займає лічені хвилини, проте залишає простір для ручної адаптації – наприклад, зміни кольорової гами, накладання логотипу, заміни тексту або треків.

Особливу увагу в Invideo приділено синтезу мовлення. Користувач має змогу обрати голос із запропонованого переліку – система підтримує як жіночі, так і чоловічі голоси з різними акцентами й тембрами. Озвучення автоматично прив’язується до сцен і відтворюється синхронно з відеорядом. Це дає змогу отримати досить переконливий результат навіть без монтажних навичок. Додатковою перевагою є можливість вбудованої генерації субтитрів на основі синтезованого тексту.

Варто зазначити, що хоча генерація сценарію і є автоматизованою, вона не завжди забезпечує глибоке семантичне узагальнення. Текст розбивається скоріше формально, ніж з урахуванням смислової структури. Тому Invideo краще підходить для відео з чітко визначеною структурою, наприклад: короткі інформаційні ролики, огляди, презентації або рекламні кліпи. Для більш комплексного узагальнення змісту, наприклад із веб-статей або довгих аналітичних текстів, можливостей системи може бути недостатньо.

1.5.3 Synthesia

Synthesia є однією з найтехнологічно просунутих платформ для створення відео, оскільки поєднує генерацію відеоконтенту з використанням віртуальних аватарів [11]. Це означає, що у фінальному відео можна побачити «живого» ведучого – синтетично згенеровану людину, яка вимовляє заданий текст, дивлячись в камеру, з мімікою,

рухами та синхронізацією губ. Такий підхід значно наближує формат автоматично створеного відео до реальних презентацій, лекцій або новинних включень. На рисунку 1.6 зображено процес створення відео з віртуальним аватаром за допомогою сервісу Synthesia.

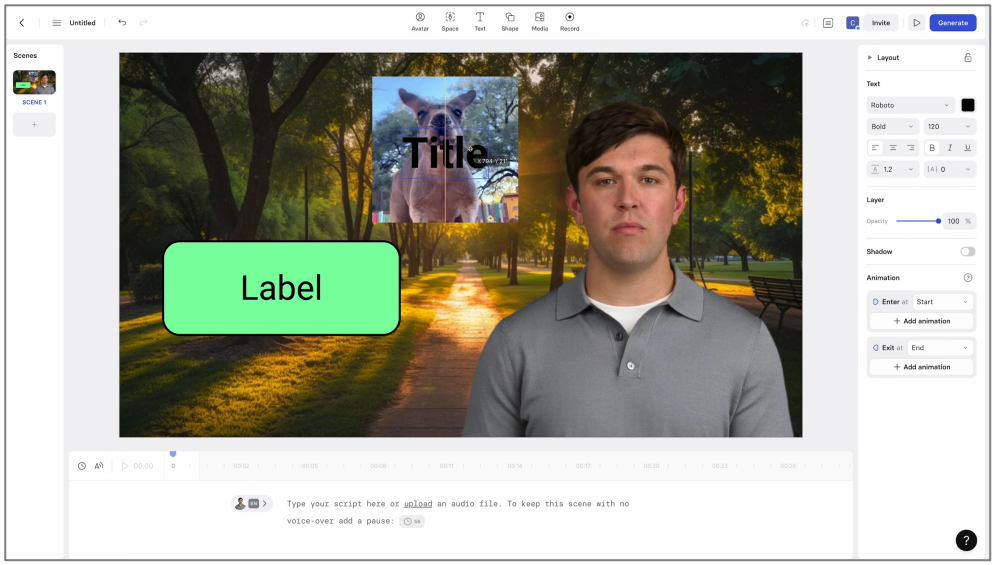


Рисунок 1.6 – Скріншот створення відео
за допомогою сервісу Synthesia

У технічному сенсі Synthesia використовує поєднання технологій комп’ютерного зору, генеративної графіки та синтезу мовлення. Віртуальні аватари генеруються на основі записаних відео реальних людей, які виступають прототипами. Далі система аналізує текст і синхронізує рухи обличчя, губ та очей, формуючи плавну і природну анімацію. Такий підхід робить контент значно більш переконливим у порівнянні зі статичними відео або слайд-шоу з голосовим супроводом.

Synthesia дозволяє створювати відео з ефектом присутності людини, що є надзвичайно актуальним у сфері електронного навчання, HR, тренінгів та клієнтської підтримки. Її можливості можуть доповнювати інші системи, які генерують текстовий сценарій, – утворюючи в такий спосіб комплексне рішення зі створення мультимедійного контенту.

1.5.4 Diffbot

На відміну від інших сервісів, що були розглянуті вище, Diffbot не є системою генерації відео в прямому розумінні. Натомість він виконує іншу, не менш важливу функцію – автоматизоване вилучення структурованого контенту з неструктурованих джерел, зокрема з веб-сторінок, навіть у складних випадках, коли структура HTML-коду є нетиповою або динамічною [12]. Саме ця функціональність є критичною для систем, які генерують відео на основі текстів, розміщених у відкритому доступі в Інтернеті. На рисунку 1.7 зображено структуровані результати парсингу веб-сторінки за допомогою сервісу Diffbot.

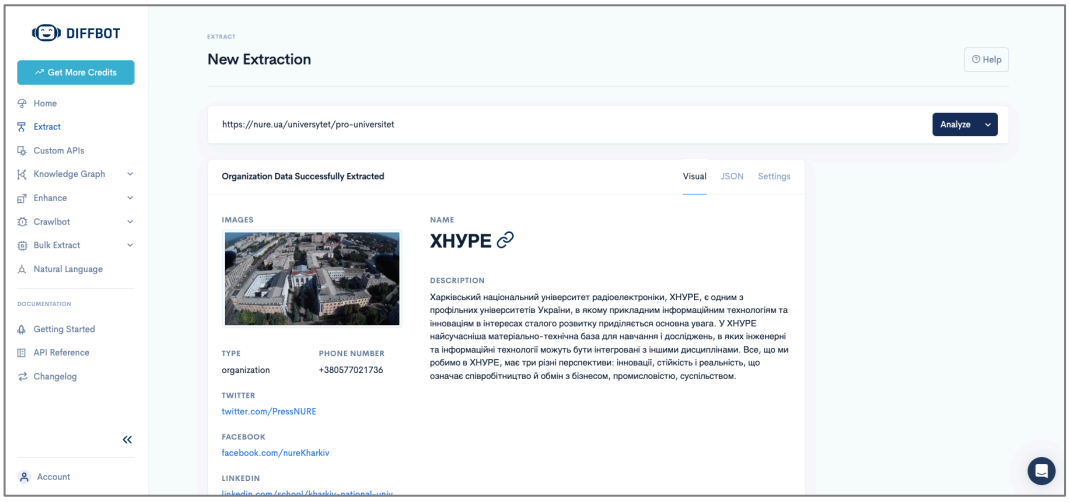


Рисунок 1.7 – Скріншот результатів парсингу веб-сторінки за допомогою сервісу Diffbot

Варто також зазначити, що Diffbot є SaaS-рішенням з API-доступом. Це означає, що його можна інтегрувати безпосередньо у веб-додаток, який розробляється в рамках дипломної роботи. За допомогою простих REST-запитів можна отримати JSON-структуру з уже виділеними ключовими елементами статті – без необхідності власноруч писати код парсера.

1.6 Постановка задачі

Метою даної кваліфікаційної роботи є розроблення, аналіз та програмна реалізація автоматизованої технології, яка дозволяє перетворювати текстовий контент з веб-сайтів у відеоформат із використанням сучасних методів обробки природної мови та нейронних мереж. Запропонована система повинна реалізовуватись у вигляді веб-додатку з окремими модулями для парсингу веб-сторінок, обробки тексту, генерації короткого змісту, пошуку мультимедійного контенту та автоматичного створення відео.

Система має забезпечити автоматизоване отримання текстових даних зі статей або новинних ресурсів, провести попередній аналіз отриманої інформації, класифікувати контент, визначити його емоційний тон та виявити іменовані сутності. На основі згенерованого короткого змісту та ключових слів має здійснюватися пошук відповідних відео та зображень. Отримані дані використовуються для формування сцен відео з додатковими елементами, такими як текстові підписи, ілюстративні зображення та озвучення синтезованим голосом.

Кінцевий користувач повинен мати змогу, не володіючи технічними знаннями, створити відео з тексту, просто надавши посилання на потрібну статтю або текстовий фрагмент. Важливою складовою є реалізація зручного та інтуїтивного веб-інтерфейсу, який забезпечить взаємодію з модулями системи.

1.7 Висновки до розділу

У цьому розділі було проведено детальний аналіз предметної галузі, в межах якої виконується дана кваліфікаційна робота, а також здійснено постановку задачі з урахуванням реальних потреб ринку та сучасних технологічних можливостей.

Було з’ясовано, що автоматичне створення відео на основі тексту є актуальним завданням, оскільки відеоконтент стає основним способом сприйняття інформації в цифрову епоху. Водночас обсяги текстової інформації, які продукуються щодня, створюють попит на засоби, що дозволяють трансформувати її у зручний для споживання формат без значних затрат часу та ресурсів.

У ході огляду існуючих підходів до парсингу веб-сторінок було виявлено, що найефективнішими інструментами для витягу змісту з HTML-документів є бібліотеки, які поєднують динамічний рендеринг із семантичним аналізом вмісту. Такі рішення, як Playwright, дозволяють коректно працювати з більшістю сучасних сайтів, незалежно від складності їх структури, а бібліотека Newspaper4k для Python є гарним і швидким інструментом для виділення основної частини веб-сторінки.

Окрему увагу було приділено огляду методів текстового аналізу та реферування. Абстрактивні моделі, зокрема BART та T5, демонструють високу якість при створенні зв’язного короткого змісту, що є надзвичайно важливим для подальшого формування сцен для відео. Семантична обробка, визначення тональності й ключових слів дозволяють зробити майбутній відеоконтент інформативним, логічним і стислим.

Наявні сервіси автоматизують окремі етапи створення відео, проте зазвичай не дають користувачам можливості змінювати логіку парсингу чи налаштовувати моделі обробки. Запропонована в межах цієї роботи система усуває ці обмеження завдяки відкритій модульній архітектурі, що забезпечує контроль користувача або розробника на всіх етапах – від отримання змісту до генерації фінального відео. Основними перевагами є гнучкість налаштувань, можливість адаптації до різних завдань, використання сучасних моделей для якісного текстового аналізу та повна автоматизація процесу.

2 ВИБІР МЕТОДІВ ТА ІНСТРУМЕНТІВ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ

2.1 Методи отримання та аналізу тексту з веб-сторінок

Одним із ключових етапів у процесі автоматизації створення відео на основі текстового контенту є отримання структурованої текстової інформації з веб-сайтів, тобто парсинг. Веб-сторінки переважно мають складну HTML-структуру, що містить численні допоміжні елементи – банери, меню, скрипти, рекламу, віджети, коментарі тощо. Це створює додаткові труднощі в автоматизованому витягу саме основного контенту, який є джерелом для подальшої обробки [13].

Існує два основні підходи до парсингу веб-сторінок:

– статичний парсинг, при якому HTML-код завантажується напряму і обробляється без врахування динамічного контенту, що генерується JavaScript;

– динамічний парсинг, який імітує завантаження сторінки у браузері, забезпечуючи повну генерацію DOM-структури перед аналізом.

Статичний парсинг є ефективним і швидким, але його можливості обмежені. Він не дозволяє витягувати контент із сайтів, що завантажують інформацію динамічно. У таких випадках застосовується динамічний рендеринг із використанням спеціалізованих бібліотек і браузероподібних середовищ.

Одним із найпопулярніших сучасних інструментів для динамічного парсингу є Playwright – бібліотека від Microsoft, яка дозволяє запускати Chromium, Firefox або WebKit у headless-режимі, емулюючи повноцінну поведінку браузера. Playwright підтримує асинхронну обробку, взаємодію з елементами сторінки, обхід CAPTCHA та авторизацію, що робить його потужним інструментом для збору даних із веб-ресурсів [14].

Для виділення основного тексту з HTML-документа використовуються бібліотеки, такі як:

– Readability.js (від Mozilla) – JavaScript-реалізація алгоритму визначення основного тексту [15];

– Newspaper4k – бібліотека для Python, яка використовує статистичні методи та евристики (співвідношення тексту до коду, щільність контенту тощо) для автоматичного визначення основного текстового блоку статті [16];

– Boilerpipe – бібліотека для Java для фільтрації HTML-документів, орієнтована на видалення шуму. Проте цей інструмент вже є застарілим і давно не оновлюється [17].

У складніших випадках застосовуються комбінації бібліотек: динамічний рендеринг разом з текстовою фільтрацією. Наприклад, спочатку отримується увесь контент веб-сторінки за допомогою браузероподібного середовища, після чого отриманий HTML аналізується, фільтруються відповідні теги, видаляється нерелевантний контент і в кінцевому випадку отримується основний контент сторінки. Такий підхід забезпечує високу точність парсингу навіть на складних або захищених сайтах.

Іншим сучасним підходом до виділення основного контенту з веб-сторінок є використання великих мовних моделей (LLM), таких як GPT, Claude, LLaMA тощо [19]. На відміну від традиційних інструментів, що покладаються на структурний або евристичний аналіз HTML-документів, LLM здатні семантично «розуміти» вміст сторінки на рівні значення та контексту. Це дозволяє їм відрізняти важливу інформацію від допоміжних елементів, навіть у випадках, коли структурні підказки відсутні або хибні.

Такий підхід зазвичай реалізується як постобробка HTML або витягнутого тексту, коли весь контент або його великі фрагменти передаються на вхід LLM з інструкцією на кшталт: «Знайди головний текст статті та сформулюй його короткий зміст». Завдяки високій контекстній чутливості, моделі можуть виявляти заголовки, підзаголовки,

абзаци та їх логічні зв’язки навіть за відсутності правильної HTML-структури.

Деякі проєкти реалізують повністю нейромережеві пайплайни для парсингу, де нейронні моделі класифікують кожен HTML-блок за роллю в структурі сторінки (контент, навігація, реклама, футер тощо). Ці моделі навчаються на великій кількості вручну розмічених сторінок, що дозволяє досягати високої точності навіть на «нестандартних» сайтах.

Використання LLM також відкриває можливість виділення контенту з мультимовних сторінок, де традиційні правила можуть не працювати. Модель може адаптуватися до мови автоматично, що значно спрощує побудову універсального рішення.

Хоча такі підходи з використанням LLM або нейромереж мають вищі обчислювальні вимоги та потребує кращої інфраструктури (GPU, обмеження на довжину контексту), вони суттєво підвищують якість обробки складних або нетипових веб-сторінок. Вони можуть використовуватись як автономно, так і в комбінації з класичними парсерами – наприклад, для фінального уточнення основного тексту або формування семантичного «знімка» сторінки.

Для подальшої обробки тексту важливо враховувати багатомовність веб-сайтів, тому визначення мови тексту є одним з важливих перших етапів після парсингу даних. Для цього використовуються бібліотеки langdetect або textblob, які дозволяють автоматично виявити мову за мовними моделями [19].

Ще один етап – попередня обробка тексту, яка включає очищення від HTML-тегів, зайвих символів, скриптів, а також нормалізацію пробілів, пунктуації, розділення на абзаци або речення. Зазвичай це реалізується з використанням регулярних виразів або засобів бібліотеки BeautifulSoup (для мови програмування Python), яка дозволяє зручно працювати з HTML-деревом [3].

У разі потреби додаткової фільтрації або налаштування правил можна створювати білі списки тегів (наприклад, залишити лише <p>, <h1>, <h2>) або використовувати навчання моделей для розпізнавання «шуму» на основі розмічених прикладів.

Загалом, парсинг є фундаментальним етапом, без якого неможливо забезпечити якісне подальше опрацювання тексту. Вибір конкретного підходу залежить від типу сайту, його структури та динаміки контенту. У рамках цієї роботи було обрано комбінацію бібліотек Playwright (для рендерингу) та Newspaper4k (для виділення основного контенту веб-сторінки), що забезпечує гнучкість і стабільну роботу з більшістю веб-ресурсів.

2.2 Методи перекладу тексту

У системах автоматизованої обробки природної мови важливим завданням є забезпечення міжмовної сумісності – зокрема, коли текстовий контент веб-сторінок представлено різними мовами. У рамках реалізації системи, що генерує відео на основі тексту, переклад відіграє ключову роль: він дозволяє обробляти контент багатьох мов у єдиному мовному середовищі, забезпечуючи узгодженість із моделями реферування, категоризації та озвучення. Для цього необхідно застосовувати ефективні методи машинного перекладу, які здатні зберігати зміст, стиль і контекст оригінального тексту.

Сучасні методи перекладу еволюціонували від простих словникових замін до потужних нейромережевих моделей. Найбільш значущим кроком стало впровадження нейромережевого машинного перекладу (Neural Machine Translation, NMT) [20]. Перші успішні реалізації NMT ґрунтувались на рекурентних нейронних мережах (RNN), які використовувалися для поетапного кодування вхідного речення та його

декодування іншою мовою. Проте ці моделі мали обмеження щодо довжини контексту та паралелізму обчислень.

Прорив у цій галузі був досягнутий завдяки трансформерній архітектурі, запропонованій у статті «Attention is All You Need» [21]. Трансформери використовують механізм самоуваги (self-attention), що дозволяє моделі фокусуватись на різних частинах вхідного речення при генерації кожного слова перекладу. Це забезпечує глибше розуміння контексту та логіки речення.

Суть роботи трансформера для перекладу полягає у двох основних компонентах: енкодері та декодері. Енкодер приймає вхідне речення, перетворює його на контекстні вектори, а декодер – генерує переклад, слово за словом, ґрунтуючись на цих векторах і раніше згенерованих словах. Схематично цей процес зображено на рисунку 2.1.

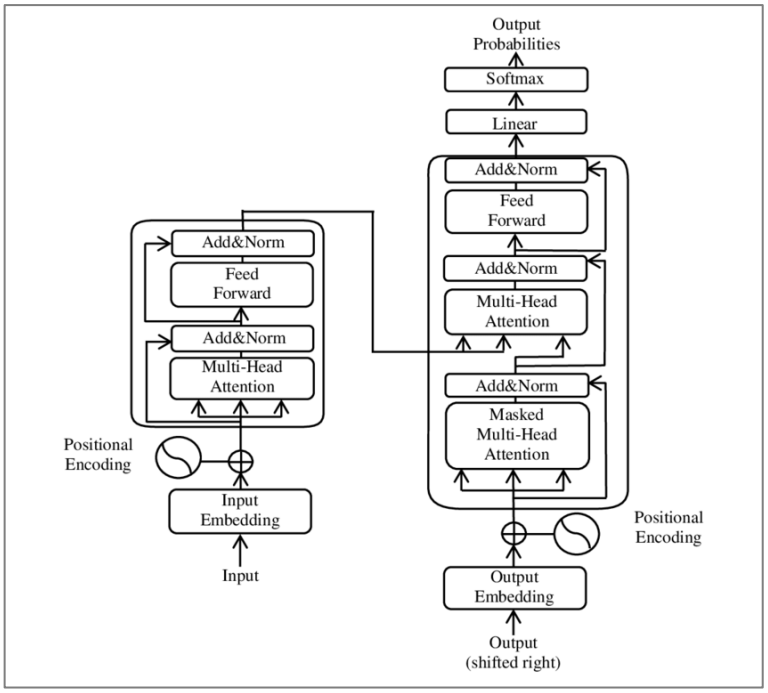


Рисунок 2.1 – Схема трансформерної архітектури

Основою механізму самоуваги є обчислення ваги важливості кожного слова у реченні щодо кожного іншого. Формально, ключова

частина трансформера – Scaled Dot-Product Attention – обчислюється за формулою:

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

де Q – матриця запитів (queries);

K – матриця ключів (keys);

V – матриця значень (values);

d_k – розмірність простору ключів.

Ця формула дозволяє моделі визначити, які слова в реченні є найважливішими для розуміння конкретного слова, що перекладається.

Серед найбільш відомих моделей перекладу на основі трансформерів варто відзначити:

– Google Translate – комерційна реалізація, яка підтримує понад 100 мов, постійно вдосконалюється, і має API-доступ;

– Facebook Fairseq / M2M-100 / NLLB – моделі від Meta, що підтримують багато пар мов і працюють без потреби в англійській як посереднику;

– Helsinki-NLP / MarianMT – відкриті багатомовні моделі, інтегровані у бібліотеку Hugging Face Transformers;

– DeepL Translator – ще один комерційний сервіс, який позиціонується як більш «стилістично правильний» для європейських мов.

У межах розроблюваної системи доцільно використовувати відкриті моделі, зокрема MarianMT [22], оскільки вони забезпечують достатню якість перекладу, не вимагають підключення до зовнішніх API та підтримують донавчання. Це дозволяє масштабувати систему, адаптуючи її до спеціалізованих тем або нестандартних мов.

Таким чином, машинний переклад є ефективно автоматизованим процесом, який легко вбудовується у загальний пайплайн системи

генерації відео. Використання трансформерних моделей дозволяє забезпечити високу якість перекладу навіть для складних текстів, а відкриті бібліотеки – інтегрувати цю функцію у веб-додаток без суттєвих витрат.

2.3 Методи текстового аналізу та узагальнення змісту

Після отримання текстового контенту з веб-сторінки наступним важливим етапом є його аналіз та узагальнення. Це складний багаторівневий процес, який включає низку послідовних етапів: визначення ключових слів, категоризацію, аналіз тональності, а також узагальнення змісту. Всі ці етапи є фундаментом для подальших кроків системи – формування сценарію, озвучення, візуалізації, структурування сцен тощо.

2.3.1 Методи виділення ключових слів з тексту

Виділення ключових слів є базовим етапом у системах обробки природної мови (NLP), особливо коли йдеться про автоматизацію створення мультимедійного контенту. Ключові слова дозволяють узагальнити зміст тексту, виділити основні теми й поняття, а також – використовуються для підбору релевантного візуального супроводу у відео.

Залежно від підходу, методи виділення ключових слів можна умовно поділити на три групи: статистичні, лінгвістичні та нейромережеві.

Статистичні методи базуються на кількісному аналізі частоти появи слів у тексті або корпусі. Найпростішим і найпоширенішим є метод TF-IDF (Term Frequency – Inverse Document Frequency), що оцінює важливість терміну в документі на тлі інших документів [23].

Формула обчислення ваги терміну t у документі d:

$$TF - IDF(t,d) = tf(t,d) \times \log \frac{N}{df(t)}, \quad (2.2)$$

де $tf(t,d)$ – частота терміна t у документі d;

$df(t)$ – кількість документів, у яких зустрічається термін t;

N – загальна кількість документів у корпусі.

TF-IDF дозволяє відсікати слова, які часто зустрічаються в багатьох текстах (наприклад, «це», «який», «даний») і фокусуватися на специфічних термінах. Однак цей метод не враховує контекст, морфологію та не розпізнає синоніми, що є його основними обмеженнями.

Також варто згадати методи TextRank і RAKE (Rapid Automatic Keyword Extraction), які будують графові структури зв'язків між словами на основі їхньої близькості в тексті та обчислюють важливість вузлів [24]. Наприклад, у TextRank ваги обчислюються подібно до алгоритму PageRank. На рисунку 2.2 зображено приклад графу ключовий слів, який було отримано за допомогою методу TextRank.

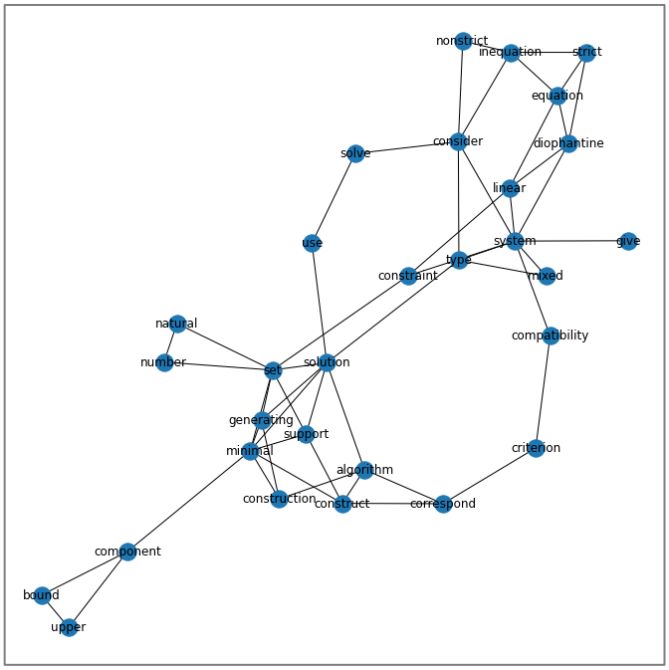


Рисунок 2.2 – Приклад графу ключових слів,
згенерованого методом TextRank

Лінгвістичні методи використовують граматичні та синтаксичні правила для виділення іменників, іменникових фраз, дієслівних конструкцій тощо. Часто застосовується Part-of-Speech (POS) Tagging – маркування частин мови у реченнях [25]. Зазвичай ключові слова – це іменники, прикметники або сталі словосполучення. Наприклад, із речення «Система генерує відео на основі короткого змісту» можуть бути виділені такі ключові фрази, як «генерація відео», «короткий зміст», «система».

Перевага лінгвістичних методів – точність і чистота результату, але вони можуть бути обмеженими для текстів із розмовною лексикою, технічними термінами або елементами іншої мови. Крім того, їх важко масштабувати для багатомовних систем.

Сучасні моделі на основі трансформерів (наприклад, BERT, DistilBERT, RoBERTa) дозволяють виділяти ключові слова, враховуючи контекст, граматику, семантику і стилістику тексту [27]. Зазвичай використовуються два підходи:

– sequence labeling – визначення, чи є кожне слово частиною ключової фрази;

– semantic similarity – обчислення схожості кожного слова або фрази до загальної тематики тексту (наприклад, через cosine similarity у векторному просторі).

Модель BERT дозволяє подавати речення на вхід, а на виході – отримувати контекстні вектори для кожного слова. Далі за допомогою класифікатора або евристик можна виділити ті вектори, які найбільше наближаються до «ключових».

Перевагою таких підходів є висока якість результатів навіть у складних текстах, а недоліком – потреба у великих ресурсах та іноді – у донавчанні моделей під конкретні задачі.

Однією з найпопулярніших бібліотек для цієї задачі є KeyBERT, яка використовує векторні уявлення слів і речень, отримані за допомогою

моделей BERT, та обчислює схожість між ними на основі косинусної відстані [27]. На рисунку 2.3 зображено принцип роботи цієї моделі.

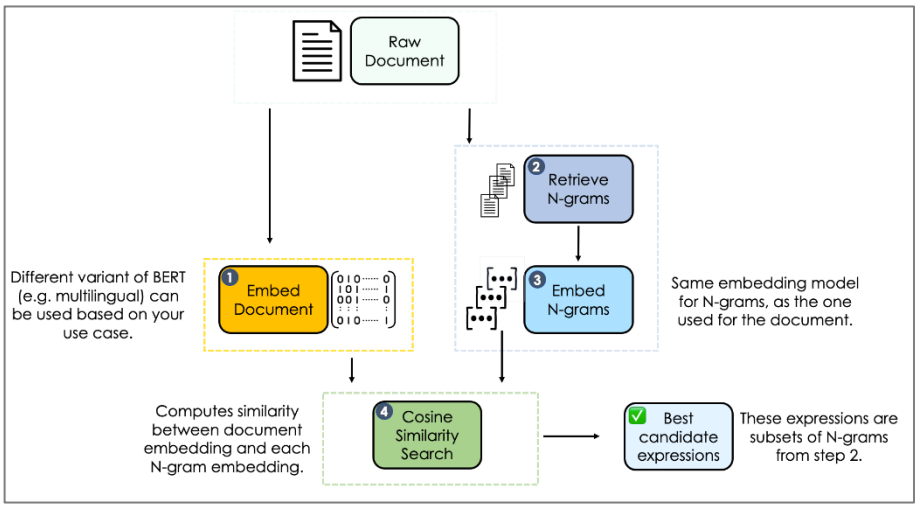


Рисунок 2.3 – Принцип роботи моделі KeyBERT

Виділення ключових слів є важливим етапом для подальшого підбору мультимедійного контенту. У рамках даної роботи було обрано використання моделі KeyBERT, щоб досягти найкращого балансу між швидкістю, точністю й контекстною релевантністю.

2.3.2 Методи класифікації тексту

Класифікація тексту – ще один важливий компонент, який дозволяє визначити належність тексту до певної тематики чи домену. Наприклад, новинна стаття може належати до категорії «політика», «спорт», «технології» тощо. Це дозволяє не лише впорядковувати інформацію, але й покращувати персоналізацію відео або підбір стилістичних рішень. Завдяки точній класифікації система може автоматично адаптувати та персоналізувати контент під інтереси аудиторії.

Класичні методи класифікації тексту ґрунтуються на використанні ручних ознак (features), які витягуються з тексту перед подачею до

алгоритму машинного навчання. Найпоширенішими способами представлення тексту є модель «торба слів» (англ. – Bag of Words) або TF-IDF-вектори, які перетворюють текст у вектори фіксованої довжини, що відображають частоту появи слів або їхню інформаційну вагу [28].

Після векторизації тексту до класифікації можна застосовувати різноманітні алгоритми, зокрема:

– наївний баєсівський класифікатор (Naive Bayes) – простий і швидкий метод, що базується на теоремі Баєса та припущенні незалежності ознак. Добре працює для задач із великою кількістю текстів, але має низьку стійкість до кореляції між словами;

– логістична регресія – ефективний лінійний метод, який добре масштабується на великі набори даних і дозволяє інтерпретувати ваги ознак;

– метод опорних векторів (SVM) – потужний алгоритм, який шукає гіперплощину, що найкраще розділяє простір ознак між класами. Добре працює навіть у випадках з багатьма нерелевантними ознаками.

Класичні методи демонструють хороші результати на коротких текстах, особливо в задачах з обмеженим словником і невеликою кількістю класів. Проте їх головним недоліком є неврахування порядку слів, контексту і граматичних залежностей, що обмежує їхню ефективність у складних або довгих текстах.

Для подолання обмежень класичних моделей у задачах текстової класифікації широкого поширення набули нейромережеві архітектури, зокрема рекурентні нейронні мережі (RNN) та їхні вдосконалені варіанти – LSTM (Long Short-Term Memory) і GRU (Gated Recurrent Unit). Їх головна перевага – здатність працювати з послідовністю слів, враховуючи контекст і порядок [29].

Базова ідея рекурентних моделей полягає у збереженні прихованого стану h_t при обробці кожного елемента послідовності, який оновлюється за допомогою вхідного вектору x_t та попереднього стану h_{t-1}:

$$h_t = \tan(Wx_t + Uh_{t-1} + b)\,, \tag{2.3}$$

де W, U, b – параметри моделі, які навчаються.

Архітектура LSTM дозволяє ефективно запам’ятовувати як короткотермінові, так і довготривалі залежності. Вона складається з трьох «воріт»: забувального, вхідного та вихідного, що контролюють потік інформації через стан пам’яті. На рисунку 2.4 зображено схему одного блоку LSTM з його «воротами».

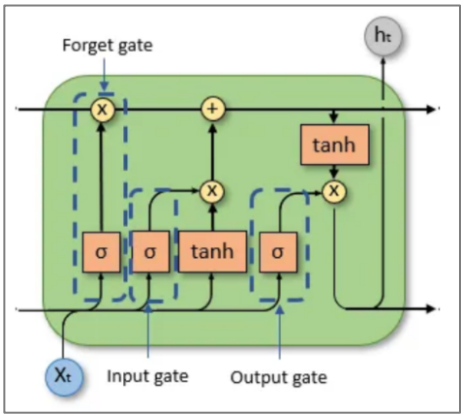


Рисунок 2.4 – Схема одного LSTM-блоку з вхідним, забувальним і вихідним воротами

Основні рівняння одного LSTM-блоку виглядають наступним чином:

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big), \tag{2.4}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{2.5}$$

$$\tilde{C}t = \tanh(W_C \cdot [ht - 1, x_t] + b_C)\,, \tag{2.6}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}t, \tag{2.7}$$

$$o_t = \sigma(W_o \cdot [ht - 1, x_t] + b_o), \quad (2.8)$$

$$h_t = o_t * \tanh(C_t), \quad (2.9)$$

де f_t – забувальне вікно (forget gate);

i_t – вхідне вікно (input gate);

o_t – вихідне вікно (output gate);

C_t – стан пам'яті;

h_t – прихований стан.

GRU – спрощена версія LSTM, яка має лише двоє воріт: ворота оновлення та ворота скидання, що дозволяє зменшити обчислювальні витрати без суттєвої втрати якості. На рисунку 2.5 зображено архітектуру GRU-блоку.

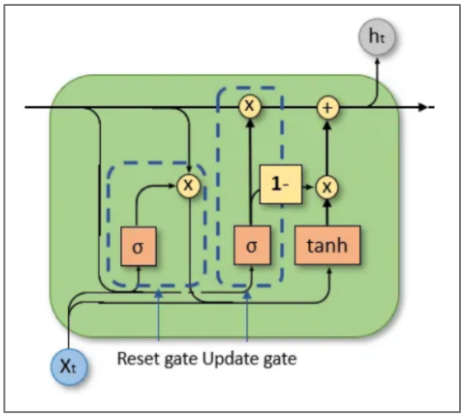


Рисунок 2.5 – Архітектура одного GRU-блоку

До основних рівнянь GRU відносять:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]), \quad (2.10)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]), \quad (2.11)$$

$$\tilde{h}t = \tanh(W \cdot [r_t * ht - 1, x_t]), \quad (2.12)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t}, \quad (2.13)$$

де z_t – оновлення (update gate);

r_t – скидання (reset gate);

h_t – новий прихований стан.

Хоча LSTM і GRU залишаються актуальними, сьогодні найкращі результати у класифікації демонструють трансформерні моделі, зокрема BERT, DistilBERT, RoBERTa. Вони здатні враховувати контекст у всій послідовності тексту одночасно (через self-attention) і підходять навіть для задач із малою кількістю прикладів (через transfer learning). Для багатомовного аналізу застосовуються моделі типу XLM-RoBERTa, що дозволяє виконувати сентимент-аналіз без попереднього перекладу тексту.

У даній роботі планується використати модель на базі LSTM або GRU, навченої на реальних прикладах тематично розмічених текстів. Такий підхід дозволить створити компактну й ефективну модель категоризації, яку легко розгорнути в межах веб-сервісу та використовувати автономно.

2.3.3 Узагальнення змісту (реферування)

Генерація короткого змісту тексту є ключовим компонентом системи автоматичного створення відео на основі тексту, оскільки саме узагальнений зміст виступає базою для формування сценарію відео. Його завдання полягає у створенні стислого, змістовного, логічного та зв’язного тексту, який зберігає основні ідеї вихідного матеріалу. Цей текст має бути достатньо інформативним, щоб передати зміст, але водночас – коротким і лаконічним, придатним для озвучення та структурування візуального ряду.

Існують два принципово різні підходи до реферування: екстрактивне та абстрактивне [30]. Екстрактивне реферування передбачає виділення найбільш релевантних речень з оригінального тексту без зміни їх формулювання. Такий підхід є технічно простішим, має невеликі вимоги до обчислювальних ресурсів і може бути реалізований за допомогою статистичних алгоритмів, таких як TF-IDF, TextRank або LexRank. Проте основним недоліком екстрактивного методу є відсутність зв'язності між вибраними реченнями, дублювання інформації, неструктурованість і занадто формальний стиль. Цей метод працює задовільно для коротких текстів (наприклад, новин), але втрачає ефективність на довших, складно структурованих або емоційно забарвлених матеріалах.

Абстрактивне реферування, натомість, працює за принципом генерації нового тексту, який переказує зміст вихідного, комбінуючи речення, перефразовуючи, узагальнюючи та логічно структуруючи інформацію. Такий підхід є значно ближчим до людського стилю викладу. Він дозволяє формувати стислий переказ тексту, зберігаючи смислові акценти, логіку розвитку думки та адаптуючи стиль під формат майбутнього відео. Абстрактивне узагальнення вимагає глибокого розуміння контексту та структури тексту, що робить використання класичних моделей малоефективним. Саме тому для цієї задачі застосовуються архітектури нейронних мереж, зокрема трансформери.

Однією з найвідоміших моделей для абстрактивного реферування є BART (Bidirectional and Auto-Regressive Transformers) [31]. Вона поєднує дві архітектури: двонапрямлений енкодер, подібний до BERT, і авторегресивний декодер, подібний до GPT. Під час тренування BART навчається відновлювати текст, пошкоджений випадковими змінами – видаленням, перестановкою або маскуванням фрагментів. Такий підхід дозволяє моделі краще розуміти загальну структуру тексту, логіку викладу і створювати цілісні перекази.

Інша потужна модель – T5 (Text-to-Text Transfer Transformer), яка була створена з метою уніфікації всіх NLP-завдань у формат «вхідний текст → вихідний текст» [32]. Такий підхід дозволяє використовувати одну й ту саму архітектуру як для перекладу, так і для класифікації, генерації, або ж реферування. Для задач узагальнення текстів дана модель демонструє результати, близькі до BART, але з дещо вищими вимогами до обчислювальних ресурсів. Варіанти цієї моделі, зменшені за розміром (наприклад, t5-small або t5-base), можуть застосовуватись у реальному часі на менш потужних системах.

Важливою перевагою трансформерних моделей є їхня здатність працювати з контекстом великої довжини, що дозволяє застосовувати їх до повноформатних статей, блогів або аналітичних звітів. Крім того, ці моделі легко адаптуються до конкретної предметної галузі – для цього достатньо провести fine-tuning (доналаштування) на спеціалізованому корпусі текстів, зокрема для певної мови, жанру або стилістики.

Узагальнення змісту є критично важливою ланкою між неструктурованим текстом і структурованим мультимедійним продуктом. Використання трансформерних моделей у цій задачі відкриває нові горизонти як у точності передавання сенсу, так і в автоматизації творчих процесів. Абстрактивне реферування перетворює класичне завдання NLP на потужний інструмент створення нових форматів контенту – зокрема, відео, що створюється автоматично на основі тексту з веб-сайтів.

2.3.4 Методи виявлення іменованих сутностей

Виявлення іменованих сутностей (Named Entity Recognition, NER) є однією з базових задач обробки природної мови, що дозволяє визначати та класифікувати конкретні елементи тексту, такі як імена людей, організацій, місць, дати, грошові суми та інші значущі категорії [33]. Використання NER суттєво підвищує рівень розуміння тексту системами штучного

інтелекту та дозволяє покращити точність подальших етапів обробки. На рисунку 2.6 зображено приклад виділення іменованих сутностей.

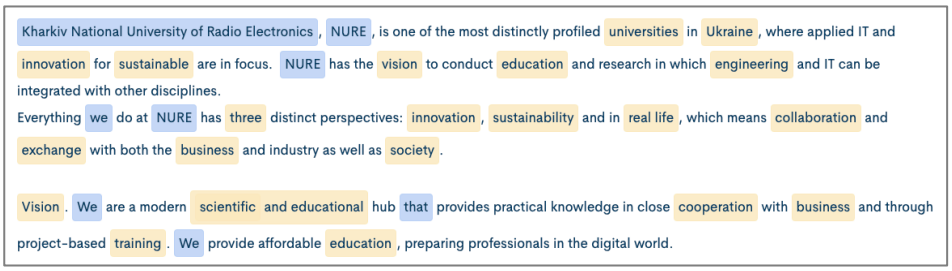


Рисунок 2.6 – Приклад виділення іменованих сутностей у тексті

У межах даної кваліфікаційної виявлення іменованих сутностей використовується не лише для аналізу тексту, але й для створення додаткового візуального контексту у відеосценах.

Визначені сутності можуть бути представлені на екрані як зображення з підписами, що робить відеоконтент більш інформативним, структурованим і привабливим для користувачів. Наприклад, якщо у тексті згадується певна особа чи організація, у відповідній сцені відео може бути відображено фотографію цієї особи або логотип організації з підписом. Це дозволяє глядачам краще сприймати зміст та швидше орієнтуватися у контенті.

Найперші методи ґрунтувалися на використанні ручних правил і шаблонів, створених експертами [34]. Вони дозволяли знаходити певні структури в тексті за допомогою регулярних виразів або заздалегідь визначених списків слів. Такі рішення були відносно простими у впровадженні, швидкими у роботі та забезпечували достатню якість для обмежених і стабільних задач. Проте масштабування таких систем на великі й різнорідні текстові потоки призводило до зниження точності та збільшення кількості помилок через неможливість враховувати складні контекстуальні залежності.

Подальшим кроком стало застосування статистичних методів, серед яких найбільш відомими є моделі прихованих марковських процесів (Hidden Markov Models, HMM) та умовних випадкових полів (Conditional Random Fields, CRF). Ці методи дозволяли автоматично враховувати контекст і навчатися на анотованих даних, що дало змогу досягти кращих результатів порівняно з rule-based підходами. Проте вони все ще мали обмеження щодо складності контексту та потребували великих обсягів розмічених даних для досягнення високої точності.

Суттєвий прорив у цій сфері відбувся з появою нейромережевих моделей. Спочатку почали використовуватися рекурентні нейронні мережі (RNN), а згодом – їх удосконалені версії, такі як двонаправлені довгі короткострокові пам’яті (Bi-LSTM), які здатні враховувати як попередній, так і наступний контекст у послідовності слів. Такі моделі демонструють значне підвищення точності, особливо при використанні word embeddings – векторних представлень слів, які дозволяють моделі розуміти семантичні подібності між словами та контекстами.

Останнім і найбільш ефективним на сьогодні етапом розвитку стали трансформерні моделі, серед яких найбільшу популярність здобули BERT та його варіації, такі як RoBERTa і XLM-RoBERTa. Вони забезпечили ще кращу здатність розпізнавати складні семантичні та синтаксичні залежності завдяки механізмам самоуваги (self-attention) і можливості враховувати контекст слів одночасно в обох напрямках. Використання таких моделей дозволяє досягти надзвичайно високої точності навіть у багатомовних і тематично різноманітних корпусах даних.

Для завдань цієї роботи було обрано використання моделі XLM-RoBERTa, яка добре зарекомендувала себе у багатьох дослідженнях з обробки природної мови та підтримує багатомовну обробку текстів. Ця модель дозволяє якісно розпізнавати сутності в текстах різними мовами, що є важливим для аналізу контенту веб-сайтів із різних джерел.

2.4 Методи вибору відповідного мультимедійного контенту

Підбір мультимедійного контенту може бути реалізований як за допомогою вже існуючих сервісів, так і з використанням нейромережевих генераторів. У рамках даної роботи доцільно комбінувати обидва підходи: використовувати стокові відео/аудіо для базового рівня, а також додати можливість інтеграції з генеративними моделями – для створення унікального, індивідуалізованого контенту.

Такий підхід дозволяє не лише підвищити якість фінального відео, а й забезпечити гнучкість системи при роботі з різноманітними тематиками та мовами. Завдяки використанню ключових слів, виділених із тексту, система може автоматично підбирати відповідні фонові відео та аудіо, які найкраще передають зміст та емоційне навантаження кожної сцени. Це значно покращує користувацький досвід і забезпечує більшу адаптивність створюваного контенту.

2.4.1 Підходи до підбору відео-контенту

Автоматизований підбір відеоряду є надзвичайно важливим етапом у системі генерації відео, адже саме він забезпечує візуальну складову, яка супроводжує озвучений текст. Правильно підібране відео не лише покращує сприйняття інформації, а й додає глядачу емоційного контексту, візуального фрейму та візуальної естетики. Існує кілька підходів до реалізації цієї задачі, які можна умовно поділити на пошукові (на основі наявних баз даних) та нейромережеві (на основі генерації або інтелектуального підбору контенту).

Найбільш практичним і популярним рішенням у сучасних системах є інтеграція зі стоковими платформами, що надають відкритий доступ до великої кількості відео. Такими сервісами є, наприклад, Pexels, Pixabay, Coverr та Videvo.

Ці сервіси мають публічні API, через які можна автоматично надсилати запити для пошуку відео за ключовими словами. У контексті даної роботи ключові слова попередньо виділяються з тексту і використовуються як основа для формування запитів.

Наприклад, якщо система виявила, що стаття стосується запуску ракети, то з неї можуть бути виділені ключові слова: «rocket», «space», «launch». Ці слова формують запит до API. У відповідь система отримує JSON-об’єкт із відео, кожне з яких має метадані: тривалість, роздільна здатність, опис, URL для завантаження. За допомогою фільтрації за релевантністю, тривалістю або співвідношенням сторін система може обрати найбільш відповідне відео.

Даний підхід простий в реалізації та ефективний при роботі з узагальненими текстами. Його недоліками є обмеженість контенту (особливо в нішевих тематиках), обмеження API (кількість запитів, роздільна здатність), а також – можливе повторне використання однакових відео в різних проєктах.

Коли використання стокових відео виявляється недостатнім або не дозволяє точно передати зміст тексту, доцільним стає застосування нейромережевих сервісів для генерації відео на основі текстових описів або промптів. Такі сервіси використовують генеративні моделі штучного інтелекту, здатні перетворити текст у відео, створене повністю штучним чином або з використанням наперед заданих шаблонів, сцен чи персонажів.

Серед найпоширеніших рішень у цьому напрямі можна відзначити сервіси Synthesia, Runway ML, Sora та інші [35]. Вони дозволяють користувачам створювати відео, де віртуальні аватари озвучують текст, або повністю згенеровані сцени, які відповідають вхідному опису. Наприклад, системі можна подати фразу на кшталт «сонячне місто біля моря з чайками у небі», і вона згенерує відео з відповідним візуальним наповненням.

Такі сервіси значно полегшують створення унікального контенту, дають змогу створювати відео на складні, вузькоспеціалізовані або емоційно забарвлені теми, де неможливо підібрати відповідний матеріал із відкритих бібліотек. Вони забезпечують гнучкість, індивідуальність та швидкість у процесі створення візуального ряду.

Разом із тим, подібні сервіси мають і низку обмежень. По-перше, більшість із них є платними або працюють за моделлю підписки, що створює залежність від зовнішніх ресурсів. По-друге, генерація відео часто займає значний час та потребує потужних обчислювальних ресурсів, особливо для моделей високої якості. Крім того, не всі сервіси дозволяють гнучку інтеграцію у власні системи – деякі не мають публічного API або знаходяться у фазі тестування, як, наприклад, Sora від OpenAI.

У контексті даної роботи використання таких сервісів розглядається як перспективний напрям для подальшого розвитку системи – як альтернатива до стокових відео у випадках, коли потрібно створити унікальний або складний за змістом візуальний супровід. Проте для реалізації системи використовується підхід зі стоковими відео, оскільки він забезпечує достатню якість при значно меншій складності реалізації, не потребує великих обчислювальних ресурсів і дозволяє швидко інтегрувати функціональність за допомогою доступних API, зокрема від сервісу Pexels [36]. Завдяки цьому система може оперативно формувати візуальний супровід для більшості типових тем без необхідності у генерації нових сцен.

Такий підхід є технічно стабільним і легко масштабованим. Водночас він залишає можливість подальшого розширення: при потребі або за наявності необхідних ресурсів система може бути доповнена модулем генерації відео на основі нейромереж. Це дозволить у майбутньому створювати повністю унікальний відеоконтент для нішевих тем або сценаріїв, де використання шаблонних відео є недостатнім або

небажаним. Таким чином, обраний підхід поєднує надійність і простоту з потенціалом до подальшого інтелектуального розвитку.

2.4.2 Підходи до підбору фонової музики

Фонова музика у відео виконує не лише естетичну функцію, а й має ключовий вплив на сприйняття контенту глядачем. Вона підсилює емоційне забарвлення, задає ритм, іноді навіть компенсує інформаційні паузи або доповнює настрій візуального ряду. Тому у системах автоматичного створення відео вибір аудіо є важливим етапом, що безпосередньо впливає на якість фінального продукту.

Одним з найпоширеніших підходів є використання музичних бібліотек з безкоштовною ліцензією (так звані Royalty Free Music). Ці ресурси надають велику кількість треків, які дозволяється вільно використовувати в некомерційних і навіть комерційних цілях за умови дотримання умов авторства або публічної ліцензії.

Одним із таких ресурсів є сайт Fesliyan Studios [37]. Ця платформа містить сотні композицій, які класифіковані за жанрами, настроєм, стилем і призначенням (наприклад, для новин, драми, бойовика, спокійної атмосфери тощо). Кожен трек супроводжується коротким описом, що дозволяє швидко зорієнтуватися щодо його змісту й застосування. Такий підхід ідеально підходить для системи автоматичного відео – користувач або система може обрати найбільш відповідний за темою трек на основі аналізу тексту або категорії відео.

Для системного підходу до побудови функціональності вибору музики також можуть використовуватись відкриті датасети. Один із прикладів – «Royalty Free Audio Dataset», розміщений на Kaggle [38]. Цей датасет містить велику кількість звукових файлів, які були зібрані з відкритих джерел. Проте його основним недоліком є відсутність метаданих, зокрема таких характеристик, як жанр, емоційне забарвлення,

стиль або темп. Це ускладнює автоматичну класифікацію композицій та потребує додаткової обробки, наприклад, за допомогою класифікаційних моделей або звукового аналізу (зокрема, через спектрограму чи тембральні характеристики).

Таким чином, для використання такого датасету у системі автоматизації потрібно реалізувати власну модель оцінки музичних треків, що зможе визначати настрій композиції (наприклад, сумна, весела, мотиваційна), її ритмічність, жанр тощо. Цей підхід потребує додаткових обчислювальних ресурсів і попереднього навчання моделі на вручну розміченому наборі треків.

Альтернативним і водночас перспективним методом є генерація музики з нуля за допомогою нейромереж, що дозволяє повністю автоматизувати створення унікального аудіоконтенту під конкретний текст або відео.

Одним із найпотужніших інструментів у сфері генеративної музики є Facebook MusicGen – модель, розроблена Meta AI [39]. Це велика трансформерна модель, натренована на більш ніж 20 тисячах годин музики, яка здатна створювати повноцінні треки на основі простого текстового опису (prompt). На відміну від класичних підходів, які покладаються на пошук існуючих треків, MusicGen дозволяє створювати музику з нуля, точно відповідаючи настрію й змісту тексту.

На рисунку 2.7 зображено схему архітектури моделі Facebook MusicGen. Архітектурно модель складається з текстового енкодера, який обробляє вхідний промпт, та декодера, який генерує аудіорепрезентацію на базі внутрішніх векторів [40]. Спочатку промпт перетворюється у векторний опис емоцій, жанру, темпу та інструментів, а далі через декодер відтворюється у вигляді звукової послідовності. У процесі роботи модель опирається на Codebook-based аудіопредставлення, подібне до технології EnCodec, що дозволяє зменшити об’єм аудіоданих без втрати якості.

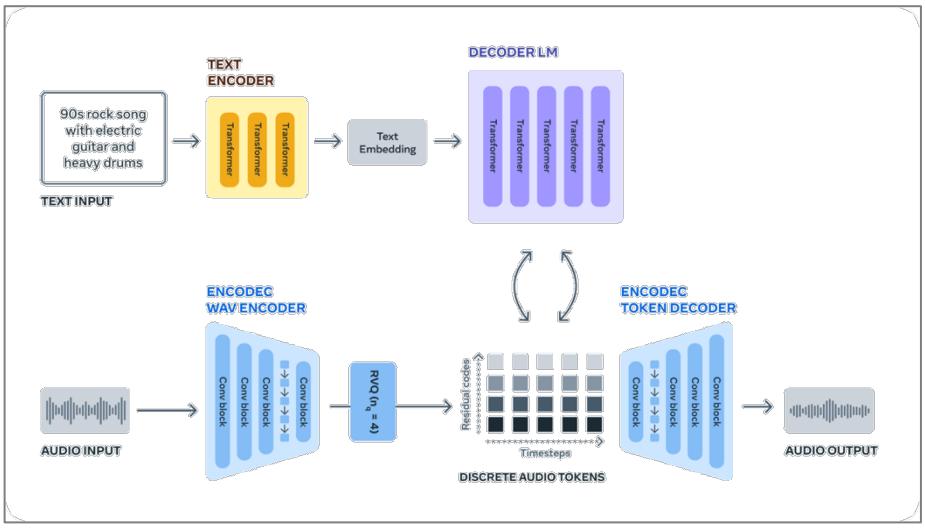


Рисунок 2.7 – Схема архітектури моделі MusicGen

Наприклад, для запиту «relaxing ambient piano music with gentle pads and slow tempo» модель згенерує 30-секундний трек, що повністю відповідає описаному настрою. Перевагою такого підходу є унікальність кожного треку, повна відповідність конкретному контенту, а також відсутність проблем із ліцензуванням.

Однак, як і в будь-якої нейромережевої системи, є низка технічних складнощів. По-перше, генерація музики вимагає ресурсів (особливо GPU для inference), по-друге – довжина треків обмежена (переважно 30 секунд), а тривалішу музику доводиться або циклічно повторювати, або комбінувати з fade-in / fade-out ефектами.

Оскільки якість згенерованої музики у моделі MusicGen безпосередньо залежить від точності та виразності текстового запиту (prompt), важливою складовою процесу є автоматичне формування цих запитів на основі змісту відео або текстового матеріалу. Замість ручного формулювання описів до музичних треків, у рамках цієї роботи використовується підхід із використанням великої мовної моделі (LLM) LLaMA 3.2 3B.

LLaMA (Large Language Model Meta AI) є сучасною архітектурою трансформерного типу, розробленою Meta AI [41]. Модель версії 3.2 з

обсягом 3 мільярди параметрів є оптимальним варіантом з огляду на продуктивність, якість генерації та обчислювальну ефективність. Вона здатна генерувати змістовні, логічно структуровані тексти, зберігаючи стиль і контекст, що робить її ідеальним інструментом для побудови описів настрою, жанру та інструментального складу майбутньої композиції.

Після отримання ключових слів, категорії тексту та емоційного тону (на попередніх етапах обробки), модель LLaMA використовується для генерації описового промпту для MusicGen. Наприклад, на основі аналізу тексту на тему новітніх технологій зі спокійним, оптимістичним емоційним фоном LLM може сформувати промпт: «inspiring ambient electronic music with soft pads, calm melody, and slow tempo, suitable for technology-themed explainer video». Далі цей промпт далі передається до MusicGen для створення відповідного аудіофрагменту.

Перевагою такого підходу є автоматизація генерації промптів високої якості, врахування контексту, адаптивність до мови тексту та категорії відео. Завдяки використанню моделі LLaMA вдається досягти оптимального співвідношення між точністю генерації та швидкістю обробки, що дозволяє застосовувати цей підхід навіть у масштабованих системах із великим потоком відеоматеріалів.

2.5 Технології синтезу мовлення для озвучення відео

Озвучення відео є однією з ключових складових у процесі його сприйняття. Людський голос додає інформації емоційності, робить відео живішим, зрозумілішим та доступнішим для широкої аудиторії. Особливо це актуально для відеоінструкцій, презентацій, новин або будь-яких інформаційних роликів, де текстовий зміст повинен бути донесений до користувача у зручній формі.

У контексті автоматичної генерації відео важливо забезпечити якісне, природне та зрозуміле озвучення тексту без залучення дикторів або ручного редагування. Для цього застосовуються технології синтезу мовлення (Text-to-Speech, TTS), які дозволяють на основі вхідного тексту згенерувати відповідну звукову доріжку з озвученням.

2.5.1 Огляд методів TTS

Розвиток технологій синтезу мовлення пройшов кілька важливих етапів – від найпростіших систем, що поєднували попередньо записані фрагменти, до сучасних нейромережевих моделей, здатних імітувати інтонації, емоції та природну вимову.

Історично перші системи TTS були конкатенативними, тобто заснованими на з’єднанні заздалегідь записаних одиниць мовлення (фонем, слів або фраз). Вони дозволяли створювати доволі розбірливе мовлення, але були обмежені жорстким набором шаблонів, не могли адаптувати інтонацію та звучали неприродньо [42].

Наступним етапом стали формантні системи, що моделювали мовлення за допомогою математичних функцій, що описують форму голосового сигналу. Вони працювали швидко та не потребували великих обсягів пам’яті, але звучали ще менш природно, ніж конкатенативні моделі, і часто сприймалися як «роботизоване мовлення».

Справжній прорив відбувся з появою нейромережевих моделей, які дозволили перейти до високоякісного, реалістичного синтезу голосу [43]. Ці моделі будують мовлення у два етапи: спочатку текст перетворюється на спектрограму (візуальне зображення частотних характеристик), а потім – за допомогою вокодерів (WaveGlow, HiFi-GAN, WaveNet) спектрограма відтворюється у звукову хвилю.

Переваги сучасних моделей – гнучкість, можливість навчання під конкретний голос, варіативність стилю, контроль емоцій та інтонації. У

комерційних продуктах, таких як Google Cloud TTS, Amazon Polly, Microsoft Azure TTS, реалізовано сотні голосів різними мовами, що працюють на базі саме таких архітектур.

2.5.2 Використання нейромережевих моделей для генерації голосу

У сучасних системах синтезу мовлення найвищу якість результату забезпечують нейромережеві моделі, які формують звукову хвилю з нуля або за допомогою проміжних представлень (наприклад, мел-спектрограм). У рамках автоматичної генерації відео, такі моделі є критично важливими для створення природного й виразного озвучення, здатного донести текст до користувача у зручній і приємній для сприйняття формі.

Одним із найпоширеніших інструментів є Google Text-to-Speech – простий сервіс, що використовує хмарну інфраструктуру Google для генерації мовлення [44]. Цей інструмент підтримує десятки мов, має високу швидкість генерації, доступний через бібліотеку gTTS у Python та дозволяє інтегрувати озвучення у проєкти з мінімальними зусиллями. Однак, gTTS має низку обмежень – зокрема, низький рівень налаштування параметрів голосу, відсутність контролю над інтонацією, стилем та емоціями, а також залежність від зовнішнього API, що може створювати проблеми з конфіденційністю або офлайн-використанням.

Для створення повноцінного кастомного голосу з більшою виразністю, гнучкістю та локальним керуванням у межах цієї роботи було обрано модель StyleTTS2 [45]. Це сучасна нейромережева архітектура, яка поєднує високоякісну генерацію мовлення з можливістю керувати стилем, емоціями та навіть індивідуальними характеристиками голосу.

StyleTTS2 – це дифузійна модель, яка розділяє мовлення на два ключових компоненти: зміст (що саме говорить мовник) і стиль (як саме це виголошується) [46]. Завдяки цьому можлива генерація варіативного мовлення з одного і того ж тексту – у різному темпі, з різними інтонаціями,

паузами, емоційною забарвленістю. У моделі використовується декілька модулів:

– Text Encoder – перетворює текст у послідовність лінгвістичних ознак;

– Style Encoder – (опціональний) дозволяє витягнути стиль із референсного голосу або згенерувати новий;

– Diffusion Decoder – відповідає за поступове «збирання» аудіо з латентного простору за допомогою стохастичних кроків.

Архітектура даної моделі зображена на рисунку 2.8.

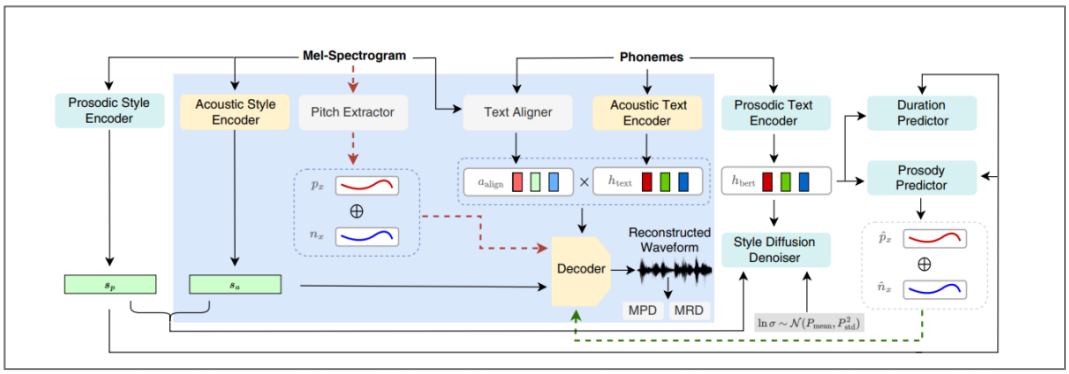


Рисунок 2.8 – Архітектура моделі StyleTTS2

Завдяки використанню цієї моделі StyleTTS2 вдається досягти високої якості та узгодженості звучання, природного темпу мовлення та збереження структури мови. Такий підхід дозволяє повністю автоматизувати озвучення тексту, не втрачаючи при цьому естетичної якості контенту.

2.6 Програмне створення відео

На завершальному етапі генерації відео необхідно об’єднати всі окремі елементи – відеофрагменти, озвучення, фонову музику – в єдиний цілісний відеофайл. Для цього застосовуються спеціалізовані програмні

бібліотеки, які забезпечують монтаж, рендеринг і експорт мультимедійного контенту. У даній роботі для реалізації цього етапу обрано бібліотеку MoviePy, яка є відкритим інструментом для обробки відео в середовищі Python.

MoviePy – це потужна та гнучка бібліотека, яка дозволяє створювати, редагувати, комбінувати та обробляти відеофайли, використовуючи простий і зрозумілий інтерфейс [47]. Вона підтримує роботу з відео, зображеннями, аудіо, текстовими вставками та анімаціями. Бібліотека побудована поверх FFmpeg – одного з найпотужніших інструментів обробки мультимедіа, що забезпечує високий рівень сумісності з різними форматами відео- та аудіофайлів. На рисунку 2.9 зображено схему створення відео за допомогою MoviePy.

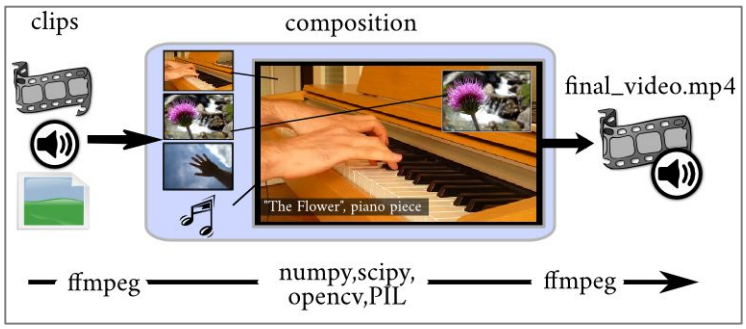


Рисунок 2.9 – Схема використання бібліотеки MoviePy
для створення відео з різних мультимедійних фрагментів

Однією з ключових переваг MoviePy є її інтегрованість з Python-екосистемою, що дозволяє легко взаємодіяти з іншими модулями обробки тексту, аудіо та зображень у межах одного проєкту. Бібліотека підтримує такі базові можливості, як обрізка відео, склеювання фрагментів, накладання звуку, тексту, вставка зображень, створення титрів, застосування ефектів, масштабування та трансформація.

Проте, попри свою гнучкість, MoviePy має і низку обмежень. По-перше, обробка відео виконується у процесі на рівні Python, що

призводить до низької продуктивності при рендерингу великих відеофайлів або при обробці великої кількості сцен. По-друге, бібліотека не забезпечує апаратного прискорення, що може негативно впливати на швидкість створення відео у масштабованих системах. Крім того, під час тривалого або ресурсомісткого рендерингу можуть виникати помилки, пов’язані з обробкою в оперативній пам’яті.

Незважаючи на зазначені недоліки, у межах даної роботи було прийнято рішення використовувати саме бібліотеку MoviePy, оскільки її функціональності достатньо для формування коротких інформаційних відеороликів, а також ця бібліотека забезпечує повний контроль над логікою монтажу. MoviePy легко інтегрується у Python-процес побудови всього відео, що дозволяє зменшити кількість зовнішніх залежностей і забезпечити зручність відлагодження на всіх етапах генерації контенту.

2.7 Інструменти для розробки веб-додатку

Для реалізації повноцінного функціонального інтерфейсу, що забезпечує взаємодію користувача з системою автоматичної генерації відео, було розроблено веб-додаток, у якому реалізовано весь цикл роботи з текстом: від введення посилання на веб-сторінку до отримання фінального відео з можливістю його перегляду та завантаження.

Архітектура веб-додатку побудована на основі стеку LEMP (Linux, Nginx, MySQL, PHP) [48]. Серверна частина реалізована за допомогою фреймворку Laravel, що забезпечує високу продуктивність, безпеку, зручну маршрутизацію запитів, вбудовану роботу з базами даних, а також потужну підтримку шаблонізації, авторизації та роботи з чергами [49]. Веб-сервером виступає Nginx, який відповідає за ефективну обробку HTTP-запитів та маршрутизацію між фронтендом і бекендом. Для зберігання структурованих даних використовується реляційна база даних

MySQL, яка надає зручні засоби для створення запитів, побудови зв’язків між таблицями та забезпечення транзакційної цілісності даних.

Клієнтська частина додатку реалізована з використанням бібліотеки React, яка дозволяє створювати динамічні, адаптивні та масштабовані інтерфейси. React забезпечує оновлення лише тих компонентів, які змінюються, що значно покращує продуктивність взаємодії з користувачем і дозволяє створювати SPA (Single Page Application) [50]. Окрім цього, компонентний підхід React сприяє кращій структуризації коду, повторному використанню логіки й пришвидшує розробку нових елементів інтерфейсу.

Для кешування даних використовується Redis – високопродуктивне сховище типу ключ–значення, яке працює в оперативній пам’яті. Його використання дозволяє значно зменшити затримки при повторному доступі до вже отриманих даних, зменшити навантаження на основну базу даних та забезпечити ефективну реалізацію систем черг у Laravel.

Інтелектуальна частина веб-додатку – модулі, що відповідають за усі процеси аналізу веб-сторінок, тексту та генерації фінального відео – реалізована у вигляді окремого мікросервісу на мові Python. Для його реалізації обрано фреймворк Flask [51], який забезпечує мінімалістичну структуру, швидкість запуску та зручність інтеграції з моделями машинного навчання. Всі фонові обчислення, зокрема парсинг веб-сторінок, генерація аудіо, музики та створення відео, виконуються асинхронно за допомогою системи управління задачами Celery [52], що дозволяє ефективно розподіляти навантаження та забезпечити стабільну роботу навіть за наявності великої кількості запитів.

Розробка програмного забезпечення здійснюється у спеціалізованих інтегрованих середовищах розробки (IDE) – PhpStorm для роботи з бекендом та фронтендом на PHP та React, та PyCharm для реалізації мікросервісів і моделей на Python. Ці середовища надають зручні інструменти для відлагодження, автодоповнення коду, тестування, роботи з БД та інтеграції з системами контролю версій.

Завдяки чіткій структурі системи та використанню сучасного стеку технологій, розроблений веб-додаток є масштабованим, стабільним і придатним для подальшого розширення функціональності, інтеграції з новими моделями та адаптації до змін користувацьких сценаріїв.

2.8 Висновки до розділу

У даному розділі було розглянуто комплекс сучасних методів, моделей та інструментів, необхідних для реалізації системи автоматичного створення відео на основі короткого змісту тексту з веб-сайтів. Проведено аналіз технологій парсингу веб-сторінок та обробки природної мови, досліджено підходи до реферування текстів, визначення ключових слів, категоризації та аналізу сентименту. Окрему увагу приділено методам підбору візуального та звукового супроводу відповідно до змісту.

У розділі також обґрунтовано вибір моделей синтезу мовлення, що забезпечують природне озвучення створюваного відео. Важливою частиною дослідження стало обґрунтування технологій генерації відео: попри наявні обмеження, використання бібліотеки MoviePy дозволяє ефективно реалізувати процес об’єднання аудіо- та візуального контенту в єдиний відеофайл.

Крім того, описано обраний стек технологій для реалізації веб-додатку, який забезпечує взаємодію з користувачем та інтеграцію всіх компонентів системи. Зокрема, використання мікросервісної архітектури дозволяє розділити функціональні частини системи на незалежні модулі, що підвищує її масштабованість, стабільність та гнучкість.

Обрані методи та інструменти повністю відповідають вимогам поставленої задачі та дозволяють побудувати сучасну систему автоматичної генерації відео, здатну працювати з великим обсягом вхідної текстової інформації, адаптуватися до її тематики та забезпечувати якісний мультимедійний результат.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

3.1 Технічне завдання

Метою створення програмного продукту є розроблення, аналіз та програмна реалізація системи для автоматизації процесу перетворення текстової інформації, отриманої з веб-сайтів, у відеоформат з використанням сучасних підходів та методів обробки природної мови та неронних мереж.

Система повинна забезпечувати повний цикл обробки: вилучення тексту з веб-сторінок, його лінгвістичний аналіз, категоризацію, реферування, підбір релевантного мультимедійного контенту, озвучення та формування фінального відеофайлу. Особливу увагу необхідно приділити забезпеченню гнучкості системи, можливості обробки різноманітних форматів веб-сторінок і масштабованості для подальшого розширення функціональності.

У межах технічного завдання передбачено розробку веб-додатку з сучасним і зручним користувацьким інтерфейсом, який дозволяє користувачам взаємодіяти з системою без необхідності технічних знань. Серверна частина повинна забезпечувати обробку запитів, зберігання даних, взаємодію з інтелектуальними мікросервісами та виконання основних обчислювальних операцій.

Результатом роботи системи повинно бути готове відео, яке відповідає змісту вхідного текстового матеріалу та містить відповідний візуальний і аудіосупровід.

3.1.1 Експлуатаційне призначення

Розроблена система орієнтована на власників інформаційних сайтів, авторів блогів, маркетологів, освітні платформи, digital-агенції та

SMM-фахівців, які мають потребу в створенні відеоконтенту на основі вже існуючих текстових матеріалів.

Сервіс дозволить користувачам:

– автоматично створювати відео з веб-сторінок без участі відеоредакторів;

– візуалізувати навчальний чи інформаційний матеріал;

– підвищити охоплення аудиторії за допомогою мультимедійного контенту;

– адаптувати текстовий контент для платформ, що переважно споживають відео (YouTube, TikTok, Instagram Reels тощо);

– генерувати озвучені відео для користувачів з порушенням зору або читання.

Система може застосовуватись як у приватному, так і в корпоративному середовищі, у внутрішніх інформаційних системах компаній, ЗМІ, освітніх установ тощо.

3.1.2 Функціональне призначення

Функціональне призначення системи полягає в автоматизації процесу перетворення текстового контенту з веб-сайтів у повноцінні відеоролики з озвученням та візуальним супроводом. Система повинна забезпечити всі етапи цього процесу – від отримання та обробки тексту до створення фінального відео та його подальшого використання. Користувач, який взаємодіє з додатком, має мати змогу ввести посилання на веб-статтю або текстовий фрагмент, після чого система самостійно виконає всі необхідні перетворення і сформує відео, яке можна переглядати, завантажувати або вставити на інші сайти.

Після отримання посилання система повинна проаналізувати структуру веб-сторінки та витягти з неї основний текст. У разі потреби вона має підтримувати рендеринг JavaScript-контенту для роботи з

динамічними сторінками. Витягнутий текст обробляється засобами попередньої фільтрації, очищення від шуму, визначення мови та граматичного структурування. Після цього здійснюється семантичний аналіз, що дозволяє визначити ключові думки та сформувати короткий зміст, придатний для сценарію відео.

На основі скороченого змісту система повинна автоматично підібрати відповідні ключові слова, що використовуватимуться для пошуку візуального супроводу – відеофрагментів, зображень, а також фонової музики. Для озвучення сценарію має бути реалізований модуль синтезу мовлення, який формуватиме аудіодоріжку природного звучання. Після збору всіх компонентів відбувається їх програмне поєднання – створюється відео зі сценами, що відповідають змісту тексту, з накладенням голосу та музики [53]. Загальна схема роботи системи зображена на рисунку 3.1.

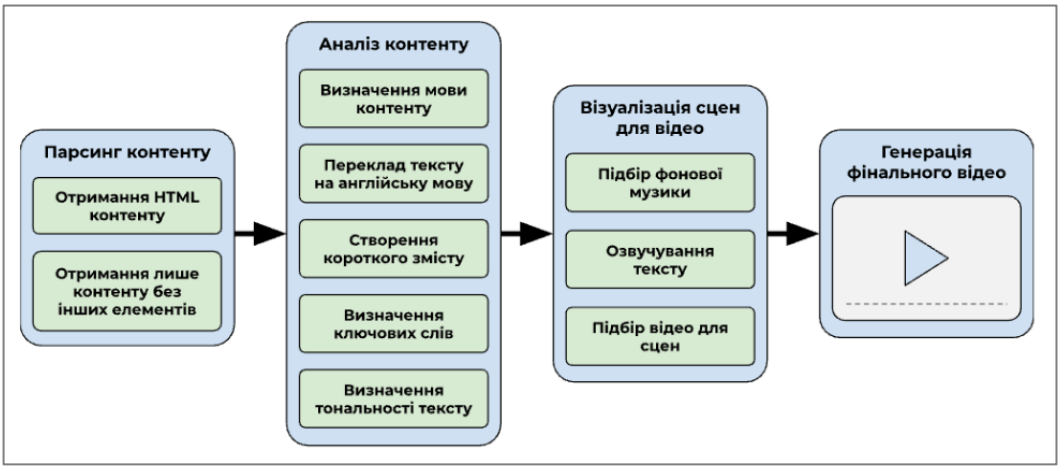


Рисунок 3.1 – Узагальнена схема технології створення відео з текстового контенту

Веб-додаток має надавати користувачу доступ до всіх етапів цього процесу через зручний інтерфейс. У ньому передбачається можливість створювати нові проєкти, налаштовувати параметри генерації відео, а також зберігати або видаляти вже згенеровані відео.

Уся система повинна працювати у фоновому режимі, не блокуючи користувача, забезпечувати масштабованість, обробку декількох запитів одночасно та високий рівень доступності. Задля цього використовуються сучасні технології бекенду (Laravel, Redis), асинхронні черги завдань і мікросервісна взаємодія з окремими модулями на Python.

3.1.3 Типові дані, які використовуються у програмному рішенні

Функціонування системи автоматичного створення відео на основі тексту з веб-сайтів передбачає обробку та збереження різних типів даних, які формують ядро всього процесу. Всі ці дані логічно поділяються на кілька груп: користувацькі дані, вхідні дані з веб-ресурсів, проміжні результати обробки та кінцеві дані у вигляді відеофайлів. Збереження даних реалізується у базі даних MySQL, частково – у файловій системі, а для проміжного кешування використовується Redis.

Користувацькі дані – це основна інформація про зареєстрованих користувачів веб-додатку, що дозволяє налаштувати персоналізовану взаємодію з системою:

– ім’я та прізвище користувача;
– електронна адреса;
– захешований пароль (з використанням bcrypt);
– роль користувача (звичайний / адміністратор).

Вхідні дані з веб-сторінок – ці дані автоматично витягуються з наданого користувачем посилання:

– URL-адреса веб-сторінки;
– мова сторінки;
– заголовок статті;
– основний текстовий контент (тіло статті);
– метадані (дата публікації, автор, ключові слова, джерело).

Проміжні дані, що виникають у процесі обробки – це результат аналізу, обробки та реферування, який надалі використовується для формування відео:

– короткий зміст тексту (summary);
– розбиття на сцени або фрагменти (скрипт відео);
– визначені ключові слова;
– визначена тональність тексту;
– озвучений текст у форматі аудіофайлу (.mp3 або .wav);
– посилання на підібрані відеофрагменти;
– фонова музика.

Дані, пов’язані зі створеними відео користувача:

– назва відео;
– налаштування відео (колір та позиціонування тексту);
– статус;
– пов’язані медіафайли (збережені на сервері або у хмарному сховищі);
– згенероване відео;
– тривалість відео;
– дата створення та оновлення даних.

Всі дані між собою пов’язані за допомогою унікальних ідентифікаторів (ID), що дозволяє ефективно будувати реляційні зв’язки в базі даних.

3.1.4 Опис сторінок додатку

Веб-додаток, що створюється в межах цього програмного рішення, повинен забезпечувати інтуїтивно зрозумілий та функціонально завершений інтерфейс для користувачів, які бажають створювати відео на основі текстових матеріалів з веб-сайтів. Структура додатку включає низку

сторінок, кожна з яких виконує чітко визначену роль у загальному процесі взаємодії з системою.

Після переходу на веб-додаток користувача зустрічає сторінка авторизації або реєстрації, залежно від потреби. У разі нових користувачів передбачається можливість реєстрації з підтвердженням електронної пошти, а також сторінка відновлення паролю у разі втрати доступу. Після входу користувач потрапляє на головну сторінку, де відображається перелік усіх його створених відео.

Користувач може перейти до сторінки перегляду кожного відео, де відображається детальна інформація: оригінальний текст веб-сторінки, згенерований короткий зміст, ключові слова, налаштування тексту для сцен відео та налаштування його кольору та позиції. На етапі створення нового відео користувач переходить на головну сторінку, де вводить посилання на текст, після чого відбувається переадресація на сторінку з відео, на якій відображується процес парсингу веб-сторінки і подальший контент з налаштуваннями. Після генерації відео користувач повинен мати змогу завантажити його.

Для користувача також наявні налаштування його профілю, де в нього є можливість оновити особисті дані, пароль, або ж видалити свій акаунт.

У разі введення некоректної адреси, несанкціонованого доступу або внутрішньої помилки, користувачу відображається сторінка з повідомленням про помилку (404 або 500). Додатково можуть бути реалізовані модальні вікна для відображення інформаційних повідомлень, попереджень чи підтверджень дій, що покращує UX-досвід користувача.

Загальна структура сторінок спрямована на логічну та зручну навігацію між основними етапами роботи з системою: від введення тексту – до отримання повноцінного відео, доступного для подальшого використання.

3.1.5 Вимоги до надійності та безпеки

У контексті надійності система повинна гарантувати стабільну та передбачувану роботу в умовах багатокористувацького середовища. Одним із ключових аспектів є обробка завдань у фоновому режимі за допомогою черг, що дозволяє уникати перевантаження серверу та гарантує виконання завдань у порядку надходження, незалежно від кількості одночасних запитів.

Система повинна бути захищеною від втрати даних – як користувацьких, так і системних. Для цього необхідно реалізувати регулярне резервне копіювання бази даних та можливість відновлення проєктів у разі збоїв. Надійність системи також передбачає автоматичне відновлення обробки відео після критичних помилок або падіння окремих мікросервісів.

З погляду безпеки особливу увагу слід приділити зберіганню особистих даних користувачів. Усі паролі повинні зберігатися у вигляді хешів за допомогою надійного алгоритму (наприклад, bcrypt), а обмін даними між клієнтом і сервером має бути захищений за допомогою HTTPS. Доступ до функціоналу повинен бути розмежований – лише авторизовані користувачі можуть створювати, редагувати або переглядати свої відео. При цьому кожне відео повинно бути ізольований від інших, тобто жоден користувач не може отримати доступ до чужого відео або тексту без явного дозволу.

Окрім базового рівня безпеки, необхідно врахувати загрози, пов’язані з обробкою зовнішніх посилань (наприклад, на веб-сторінки). Для цього передбачено перевірку URL-адрес на коректність, недопущення виконання потенційно небезпечних скриптів, а також обмеження часу рендерингу динамічних сторінок для захисту від DDoS-атак або спроб навмисного перевантаження системи.

Загалом, система повинна відповідати сучасним вимогам до безпеки веб-сервісів, бути стійкою до типових загроз, забезпечувати конфіденційність і цілісність даних, а також гарантувати стабільну роботу при масштабуванні.

3.2 Діаграма варіантів використання

Для забезпечення чіткого розуміння функціональних можливостей системи та сценаріїв її використання було розроблено діаграму варіантів використання (use case diagram), яка зображена на рисунку 3.2. Дана діаграма відображає основні взаємодії користувачів із системою та взаємодію між окремими компонентами програмного продукту.

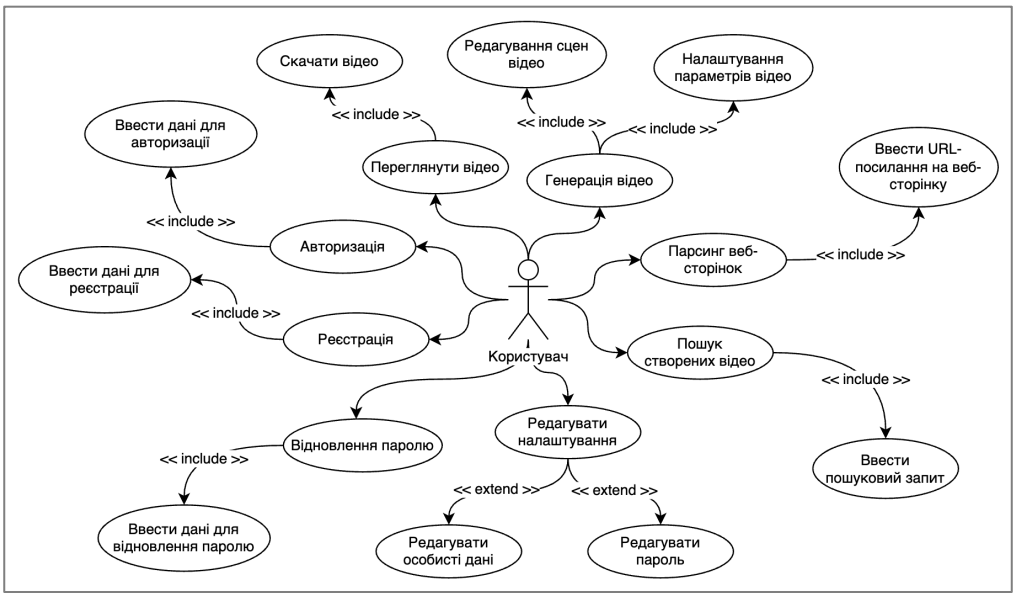


Рисунок 3.2 – Діаграма варіантів використання

Головним користувачем системи є кінцевий користувач, який взаємодіє з веб-додатком для створення відеоконтенту. Основні сценарії використання включають: введення посилання на веб-сторінку, перегляд попереднього короткого змісту, вибір параметрів відео, ініціалізацію процесу генерації відео та перегляд/завантаження готового результату.

Дана діаграма дозволяє відобразити ключові взаємодії в системі, структурувати процес розробки та забезпечити зрозуміле представлення функціональності проєкту.

3.3 Опис архітектури програмного продукту

Архітектура розробленого програмного продукту побудована за принципами клієнт-серверної моделі з використанням мікросервісного підходу для обробки інтелектуальних задач. Це дозволяє забезпечити масштабованість, розподіл навантаження, а також спрощує підтримку та розширення системи в майбутньому.

На рисунку 3.3 представлено основні компоненти системи та взаємозв'язки між ними, що демонструє чіткий розподіл обов'язків і взаємодію клієнтської частини, серверного додатку, мікросервісу інтелектуальної обробки та бази даних.

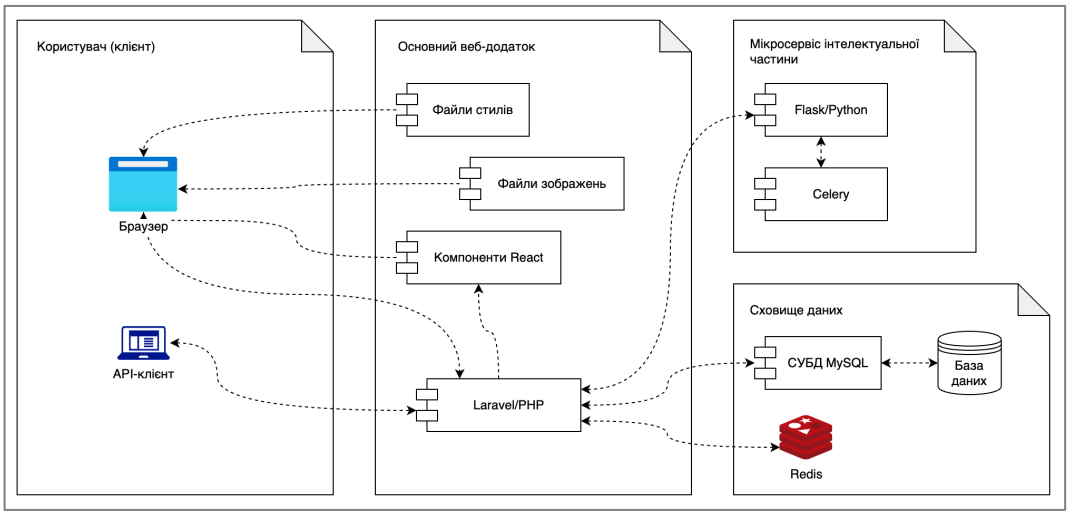


Рисунок 3.3 – Діаграма архітектури проєкту

На стороні клієнта передбачено використання браузера для кінцевих користувачів, а також можливість підключення зовнішніх API-клієнтів для автоматизації певних задач чи інтеграції з іншими сервісами. Веб-

інтерфейс реалізується за допомогою фреймворку React, що забезпечує високу швидкість роботи та зручність взаємодії для користувачів. Компоненти React відповідають за відображення інтерфейсу, обробку подій та взаємодію з серверною частиною через API.

Основна серверна частина реалізується на базі фреймворку Laravel (PHP), який забезпечує логіку обробки запитів від клієнтів, управління користувачами, авторизацію та автентифікацію, а також роботу з базою даних.

Для обробки тривалих чи ресурсомістких задач в серверній частині використовується механізм черг, вбудований у Laravel. В якості драйвера для черг застосовується Redis, а процеси-воркери контролюються за допомогою Supervisor, що дозволяє надійно обробляти фонові задачі та автоматично відновлювати виконання у разі помилок.

Інтелектуальні функції системи (обробка тексту, категоризація, реферування тексту, виділення іменованих сутностей, генерація музики, озвучення текстів та генерація фінальних відео) винесено в окремий мікросервіс, який реалізується на мові програмування Python з використанням веб-фреймворку Flask. Для асинхронного запуску задач та їх розподілу між воркерами використовується система Celery. Взаємодія між основним веб-додатком і мікросервісом здійснюється через API з використанням механізму вебхуків (webhooks).

Дані користувачів, текстовий контент та інформація про створені відео зберігаються у реляційній базі даних MySQL. Для оптимізації роботи з тимчасовими даними, кешування та обробки черг Laravel застосовується сховище Redis.

3.4 Діаграма класів серверної частини основного додатку

Серверна частина розробленого програмного продукту побудована за класичною багаторівневою архітектурою з використанням принципів

об’єктно-орієнтованого програмування (ООП), а також шаблонів проєктування «Контролер-Сервіс-Репозиторій». Це дозволяє досягти чіткого розподілу обов’язків між компонентами системи, спрощує супровід і подальший розвиток програмного забезпечення.

На рисунку 3.4 зображено діаграму класів, на якій представлено представлено основні класи серверної частини, які взаємодіють між собою для забезпечення функціональності веб-додатку.

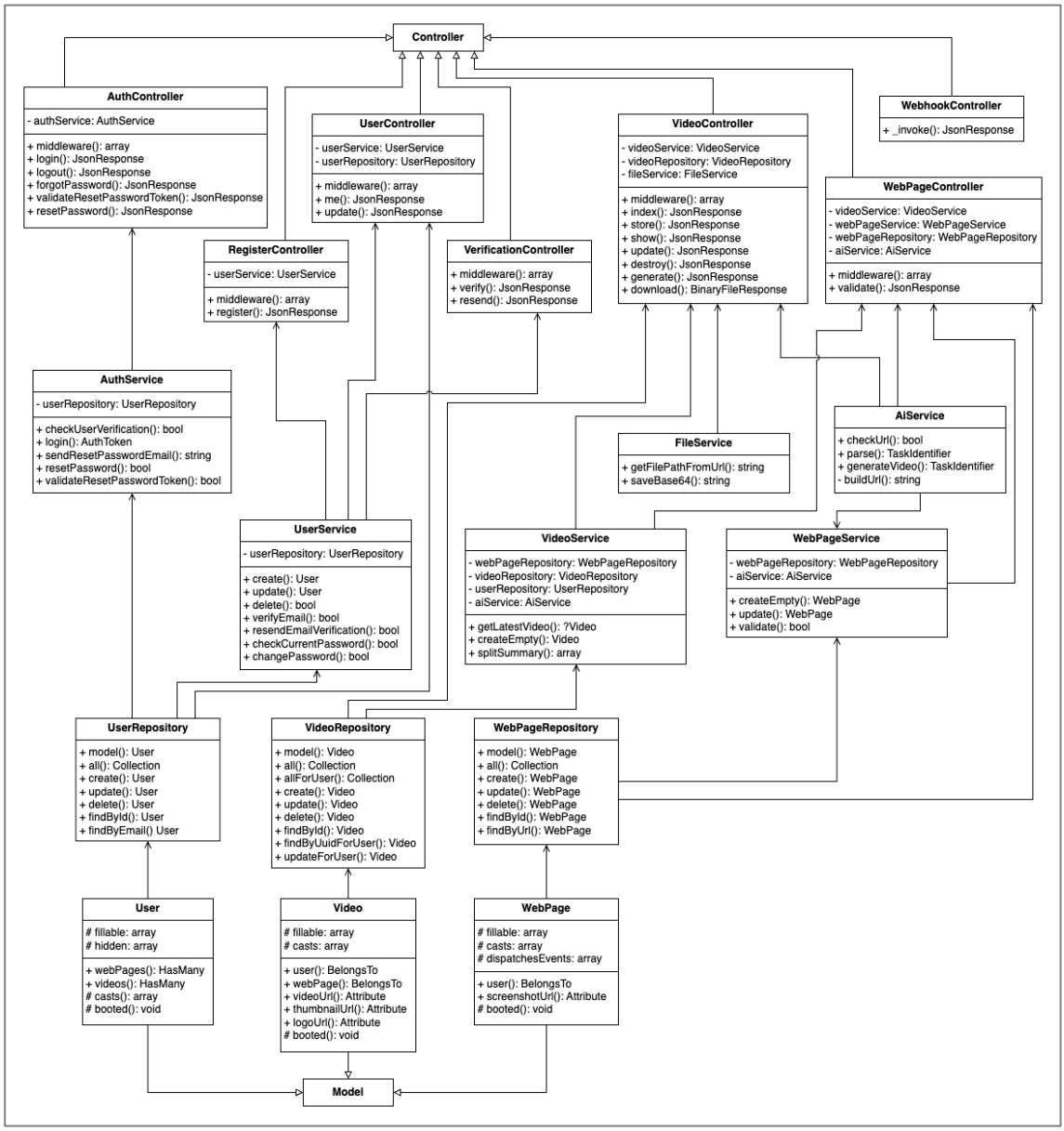


Рисунок 3.4 – Діаграма класів серверної частини основного веб-додатку

Усі HTTP-запити від користувачів обробляються контролерами, які виступають посередниками між зовнішнім інтерфейсом і внутрішньою бізнес-логікою. У системі передбачено такі контролери: AuthController, RegisterController, UserController, VerificationController, VideoController, WebPageController та WebhookController. Кожен контролер відповідає за обробку певних типів запитів. Наприклад, AuthController здійснює авторизацію та автентифікацію користувачів, VideoController – обробку операцій із відео, а WebPageController – управління та парсинг веб-сторінкок.

Контролери звертаються до сервісів, які реалізують бізнес-логіку. Це дозволяє уникнути дублювання коду в контролерах і забезпечити гнучкість при зміні логіки роботи додатку. На діаграмі відображено сервіси: AuthService, UserService, VideoService, WebPageService, AIService та FileService. Наприклад, VideoService відповідає за операції створення відео, розподіл сценарію відео (summary) по сценах та інтеграцію з іншими сервісами. AIService реалізує взаємодію з мікросервісами штучного інтелекту для виконання завдань парсингу, аналізу контенту та генерації фінального відео.

Кожен сервіс для роботи з даними використовує відповідний репозиторій, який інкапсулює взаємодію з моделями бази даних та виконує CRUD-операції. Це дозволяє централізовано керувати операціями доступу до даних і дотримуватися принципу єдиного джерела істини для кожної сутності.

Моделі представляють сутності, що зберігаються у базі даних. Вони визначають властивості об’єктів і зв’язки між ними. Наприклад, модель User пов’язана з багатьма відео (HasMany), а модель Video – із конкретною веб-сторінкою (BelongsTo). Властивості моделей відповідають полям у базі даних і визначають їхні типи, обмеження та додаткові атрибути.

Додатково у складі системи присутній WebhookController, який дозволяє обробляти запити від мікросервісу інтелектуальної частини, наприклад, отримання результатів обробки веб-сторінок чи генерації відео.

На основі цієї структури забезпечено високу модульність та масштабованість програмного продукту. Така архітектура дозволяє легко додавати нові функції або модифікувати існуючі без суттєвого втручання в інші частини коду.

3.5 Опис бази даних

Для забезпечення зберігання та ефективної обробки даних у межах розробленої системи було спроєктовано реляційну базу даних з 10 таблиць, яка відображає всі необхідні сутності та взаємозв’язки між ними. Архітектура бази даних побудована таким чином, щоб забезпечити максимальну гнучкість, масштабованість і простоту в обслуговуванні. Враховуючи використання Laravel як основного фреймворку для бекенд-розробки, частина таблиць була створена автоматично для підтримки авторизації, роботи з токенами та міграціями.

Таблиця users містить відомості про всіх зареєстрованих користувачів системи. Кожен запис включає унікальний ідентифікатор ID, унікальний UUID, ім’я та прізвище, електронну пошту, пароль, а також поля для підтвердження електронної пошти і токен запам’ятовування користувача при повторних входах у систему. Поля created_at і updated_at фіксують час створення та останнього оновлення запису.

Для адміністраторів використовується окрема таблиця admin_users, яка має схожу структуру, але доповнена полем role, що визначає рівень доступу адміністратора.

Для керування авторизаційними токенами Laravel автоматично створив таблицю personal_access_tokens, яка містить токени для

авторизації користувачів через API, та таблицю password_reset_tokens для обробки запитів на відновлення пароля.

Таблиця web_pages є центральною для зберігання інформації про веб-сторінки, з яких отримується контент для подальшої обробки. Вона містить URL сторінки, статус обробки, мову, заголовок, вміст сторінки, короткий зміст (summary), результати аналізу сентименту (sentiment), ключові слова, категорії, авторів, а також посилання на скріншот сторінки. Для гнучкості дані про ключові слова, категорії та зображення зберігаються у форматі JSON.

Таблиця videos призначена для зберігання інформації про всі відеофайли, створені на основі аналізованих веб-сторінок. Вона пов’язана із таблицями users і web_pages через відповідні зовнішні ключі user_id і web_page_id. Кожен запис містить дані про сцени відео, шляхи до відеофайлів, фонової музики, озвучення, логотипу, а також інформацію про колірне оформлення, орієнтацію відео та якість. Для кожної сцени можуть бути збережені шляхи до зображень і текстових підписів, що дозволяє автоматизовано додавати візуальні елементи у відео.

Для обробки фонових завдань (черг), які використовуються для асинхронного виконання ресурсомістких процесів (наприклад, фонові запити на API мікросервісів чи інші сервіси, відправлення листів на електронні пошти користувачів, тощо), застосовуються стандартні таблиці Laravel: jobs, job_batches та failed_jobs. Таблиця jobs містить інформацію про всі заплановані завдання, включаючи назву черги, навантаження (payload), кількість спроб виконання і час доступності завдання. Таблиця job_batches дозволяє групувати завдання у пакети для більш ефективної обробки, а failed_jobs зберігає інформацію про завдання, які завершились невдало.

Таблиця migrations є службовою і використовується для відстеження стану виконаних міграцій бази даних у Laravel. Вона дозволяє керувати

змінами структури бази даних у процесі розробки та оновлення програмного продукту.

Структура бази даних відображена на рисунку 3.5, де показані всі таблиці, їхні основні поля та взаємозв'язки між ними.

Рисунок 3.5 – Загальна структура бази даних

Розроблена структура бази даних забезпечує повний цикл роботи системи: від обліку користувачів та веб-сторінок до обробки фонових завдань і зберігання даних про згенеровані відео. Вона оптимізована для масштабування, розширення функціоналу та ефективної роботи в умовах високого навантаження.

3.6 Висновки до розділу

У даному розділі було здійснено детальне проєктування програмного продукту, що дозволило сформувати чітке технічне завдання та визначити ключові функціональні й експлуатаційні характеристики системи. Визначено основні вимоги до функціонування, надійності та безпеки програмного забезпечення, а також описано типові дані, які використовуються у процесі роботи додатку. Це створило міцну основу для подальшої розробки і впровадження.

Розроблено діаграму варіантів використання, яка відобразила основні сценарії взаємодії користувачів із системою. Це дало змогу краще зрозуміти поведінку користувачів та оптимізувати функціональність веб-додатку.

У рамках архітектурного проєктування було побудовано компонентну структуру системи, яка забезпечує модульність, гнучкість і масштабованість рішення. Діаграма класів серверної частини детально описала взаємозв’язки між основними об’єктами системи, зокрема контролерами, сервісами, репозиторіями та моделями. Це дозволило забезпечити чіткий розподіл обов’язків та спростити подальший супровід і розвиток програмного продукту.

Окремо було спроєктовано структуру бази даних, яка повністю відповідає функціональним вимогам додатку та підтримує ефективне зберігання й обробку всіх необхідних даних. Враховано взаємозв’язки між користувачами, веб-сторінками, відео та іншими сутностями, що дозволяє забезпечити цілісність і узгодженість даних.

Таким чином, результати проєктування створили ґрунтовну технічну базу для реалізації програмного продукту, забезпечивши високий рівень організації коду, ефективність обробки даних та можливість подальшого розширення функціоналу системи.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

4.1 Технічні характеристики обладнання

Для виконання програмної реалізації та проведення експериментальних досліджень у межах даної кваліфікаційної роботи було використано два комп’ютери з різними технічними характеристиками: один для розробки і тестування інтелектуальної частини, інший – у якості основного серверу веб-додатку.

Комп’ютер для експериментальних досліджень, на якому здійснювалася навчання та тестування моделей, обробка відеоконтенту та виконання обчислювально-інтенсивних завдань, мав такі характеристики:

– процесор AMD Ryzen 5 5600 6-Core 3.50 ГГц;
– оперативна пам’ять 32 ГБ DDR4 з частотою 3600 МГц;
– графічний адаптер AMD Radeon RX 6700 XT 16 ГБ.

Це обладнання забезпечило достатній рівень обчислювальної потужності для роботи з нейронними мережами, обробки відео високої роздільної здатності та тестування програмного продукту в умовах, наближених до реальних сценаріїв використання.

Комп’ютер, використаний як серверна платформа, на якому було розгорнуто основну серверну частину додатку та мікросервіс інтелектуальної частини:

– процесор Intel Core i5-8500T 2.10 ГГц;
– оперативна пам’ять 16 ГБ DDR4 з частотою 2666 МГц.

Серверне обладнання продемонструвало стабільну роботу додатку при виконанні запитів користувачів, обробці даних та взаємодії з мікросервісами. Для обробки тривалих або ресурсомістких задач, таких як навчання моделей чи генерація відео, використовувалися асинхронні черги завдань і кешування, що дозволило зменшити навантаження на сервер.

Використання двох різних конфігурацій забезпечило повноцінне тестування усіх аспектів роботи системи: як з боку кінцевого користувача, так і в умовах серверного навантаження.

4.2 Отримання та обробка тексту

На першому етапі роботи системи відбувається отримання текстової інформації з веб-сторінок. Для цього було розроблено програмний модуль, що реалізує парсинг сторінок за допомогою бібліотеки Playwright, яка дозволяє здійснювати взаємодію з динамічними веб-сторінками, включно з тими, які потребують виконання JavaScript-коду для повного завантаження контенту. Це забезпечує коректне отримання текстових і медіа-даних навіть із сучасних сайтів зі складною структурою.

Процес отримання даних реалізовано у вигляді асинхронного завдання, яке виконується у черзі Celery. Для контролю та перегляду інформації про виконання асинхронних завдань, використовується пакет Flower, зображений на рисунку 4.1. Це дозволяє виявляти помилки при виконанні різних завдань з обробки текстових даних і створенні відео.

Рисунок 4.1 – Перегляд завдань у черзі Celery
за допомогою інструменту Flower

Спочатку за допомогою Playwright завантажується сторінка у браузероподібному середовищі, створюється її знімок і зберігається HTML-вміст. В подальшому скріншот веб-сторінки використовується як

частина UI-інтерфейсу створення відео для наглядного зображення сторінки, яку було оброблено.

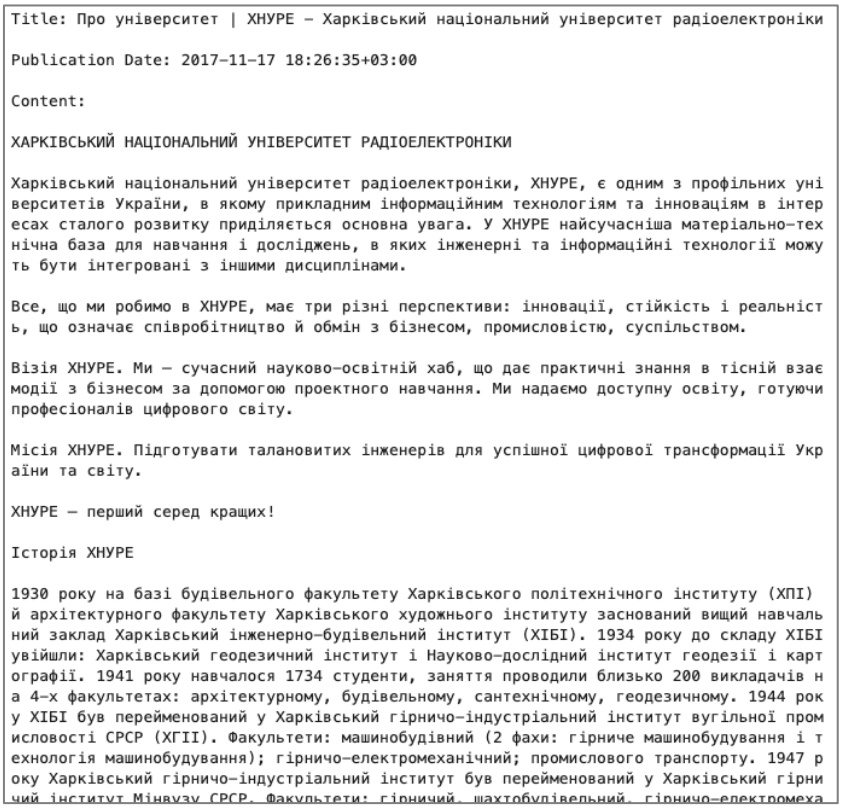
На рисунку 4.2 зображено приклад знімку веб-сторінки з сайту Харківського національного університету радіоелектроніки.

Рисунок 4.2 – Приклад знімку веб-сторінки, отриманого за допомогою Playwright

Далі отриманий скріншот завантажується до хмарного сховища MiniO [54], а посилання на нього передається через вебхук назад у головну систему. Цей етап забезпечує централізоване зберігання медіа-файлів і доступ до них для інших компонентів додатку. MiniO було обрано як основне рішення для об’єктного зберігання даних завдяки його сумісності з протоколом Amazon S3, високій швидкодії, масштабованості та можливості локального або хмарного розгортання. Це дозволяє легко інтегрувати зберігання файлів із різними сервісами додатку та забезпечити надійне управління доступом до ресурсів.

Після завантаження контенту здійснюється його попередня обробка за допомогою бібліотеки newspaper4k, яка автоматично виділяє основний текст статті, авторів, дату публікації та інші метадані. На рисунку 4.3

зображено приклад результату обробки отриманого HTML-контенту за допомогою бібліотеки newspaper4k.



```
Title: Про університет | ХНУРЕ – Харківський національний університет радіоелектроніки

Publication Date: 2017-11-17 18:26:35+03:00

Content:

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Харківський національний університет радіоелектроніки, ХНУРЕ, є одним з профільних уні
верситетів України, в якому прикладним інформаційним технологіям та інноваціям в інтер
есах сталого розвитку приділяється основна увага. У ХНУРЕ найсучасніша матеріально-тех
нічна база для навчання і досліджень, в яких інженерні та інформаційні технології можу
ть бути інтегровані з іншими дисциплінами.

Все, що ми робимо в ХНУРЕ, має три різні перспективи: інновації, стійкість і реальніст
ь, що означає співробітництво й обмін з бізнесом, промисловістю, суспільством.

Візія ХНУРЕ. Ми – сучасний науково-освітній хаб, що дає практичні знання в тісній взає
модії з бізнесом за допомогою проектного навчання. Ми надаємо доступну освіту, готуючи
професіоналів цифрового світу.

Місія ХНУРЕ. Підготувати талановитих інженерів для успішної цифрової трансформації Укр
аїни та світу.

ХНУРЕ – перший серед кращих!

Історія ХНУРЕ

1930 року на базі будівельного факультету Харківського політехнічного інституту (ХПІ)
й архітектурного факультету Харківського художнього інституту заснований вищий навчаль
ний заклад Харківський інженерно-будівельний інститут (ХІБІ). 1934 року до складу ХІБІ
увійшли: Харківський геодезичний інститут і Науково-дослідний інститут геодезії і карт
ографії. 1941 року навчалося 1734 студенти, заняття проводили близько 200 викладачів н
а 4-х факультетах: архітектурному, будівельному, сантехнічному, геодезичному. 1944 рок
у ХІБІ був перейменований у Харківський гірничо-індустріальний інститут вугільної пром
ісловості СРСР (ХГІІ). Факультети: машинобудівний (2 фахи: гірниче машинобудування і т
ехнологія машинобудування); гірничо-електромеханічний; промислового транспорту. 1947 р
оку Харківський гірничо-індустріальний інститут був перейменований у Харківський гірни
чий інститут Мінвузу СРСР. Факультети: гірничий, шахтобудівельний, гірничо-електромеха
```

Рисунок 4.3 – Приклад результату обробки отриманого HTML-контенту

Результат обробки HTML-структури веб-сторінки – а саме її контент – використовується для подальшого аналізу.

4.3 Категоризація тексту

Визначення категорії тексту є важливим етапом у процесі аналізу тексту. Від правильної категоризації залежить не лише точність підбору мультимедійного контенту, але й загальна релевантність згенерованого відео для кінцевого користувача. Автоматичне присвоєння категорії текстовим даним дозволяє оптимізувати вибір візуальних та

аудіоелементів відповідно до тематики, підвищуючи якість фінального продукту.

4.3.1 Опис набору даних

Для реалізації задачі категоризації текстових матеріалів було використано два ключові датасети, доступні на ресурсі Kaggle. Перший із них – «Trending YouTube Video Statistics» – містить відомості про відео з різних країн, зокрема США, Великої Британії, Німеччини, Канади та Франції [55]. У цьому наборі представлено такі характеристики, як назви відео, їхні описи, кількість переглядів, вподобань, категорії та інші супровідні метадані. Загальна кількість записів у цьому корпусі сягає приблизно 200 тисяч, що створює належну базу для навчання алгоритмів.

Другий набір – «YouTube Trending Video Dataset (updated daily)» – також містить інформацію про популярні відео з різних країн, таких як США, Індія, Німеччина, Південна Корея та інші [56]. Він має схожі характеристики з першим датасетом, однак основний акцент зроблено на відео, які здобули популярність протягом конкретних часових проміжків. Важливо зазначити, що цей набір даних постійно оновлюється майже щодня, що забезпечує високу актуальність інформації. Загальний обсяг записів у ньому перевищує 600 тисяч, що дає змогу моделі навчатися на широкому спектрі даних, які відображають сучасні тренди та тематику відеоконтенту.

Об’єднавши два набори даних, було вибрано відомості про популярні відео з таких країн, як США, Канада та Велика Британія, що дозволило створити великий і різноманітний датасет, який містить дані про понад 800 тисяч відео з різних категорій. Це дає змогу моделі враховувати різні аспекти популярності відео та їх жанрову приналежність. Важливою частиною підготовки даних є їх очищення і попередня обробка, що включає видалення дублюючих записів, обробку відсутніх значень і

нормалізацію текстових даних. Ці кроки гарантують високу якість вхідних даних для навчання моделі і підвищують точність класифікації.

На рисунку 4.4 зображено приклад даних необробленого датасету.

	title	category_id	view_count
0	I ASKED HER TO BE MY GIRLFRIEND...	22	1514614
1	Apex Legends | Stories from the Outlands – "Th...	20	2381688
2	I left youtube for a month and THIS is what ha...	24	2038853
3	XXL 2020 Freshman Class Revealed - Official An...	10	496771
4	Ultimate DIY Home Movie Theater for The LaBran...	26	1123889

Рисунок 4.4 – Приклад даних необробленого набору даних

На рисунку 4.5 зображено розподіл категорій відео, який свідчить про те, що деякі категорії є більш популярними, ніж інші.

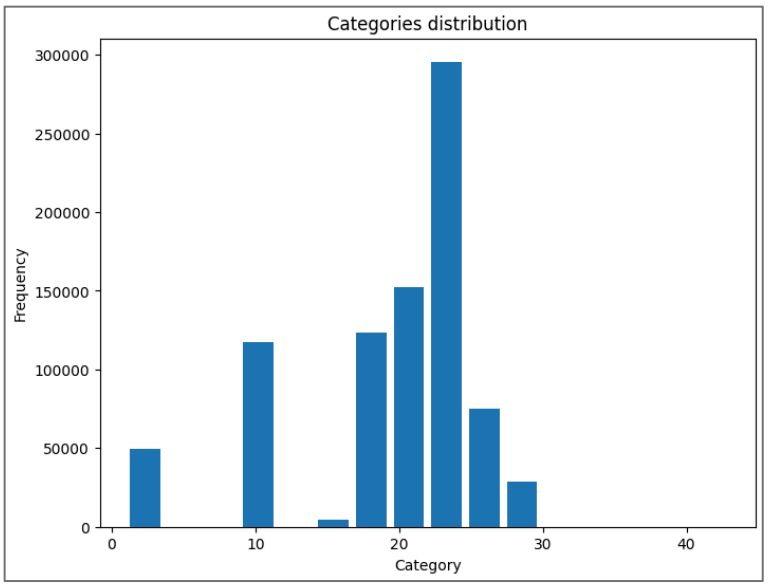


Рисунок 4.5 – Розподіл кількості відео за категоріями

Довжина заголовків відео також варіюється. Розподіл довжини заголовків, представлений на рисунку 4.6, показує, що більшість з них мають довжину від 25 до 100 символів. Ця інформація є важливою для попередньої обробки тексту, оскільки вона допомагає визначити

оптимальні параметри для токенізації та підготовки текстових даних для навчання моделі.

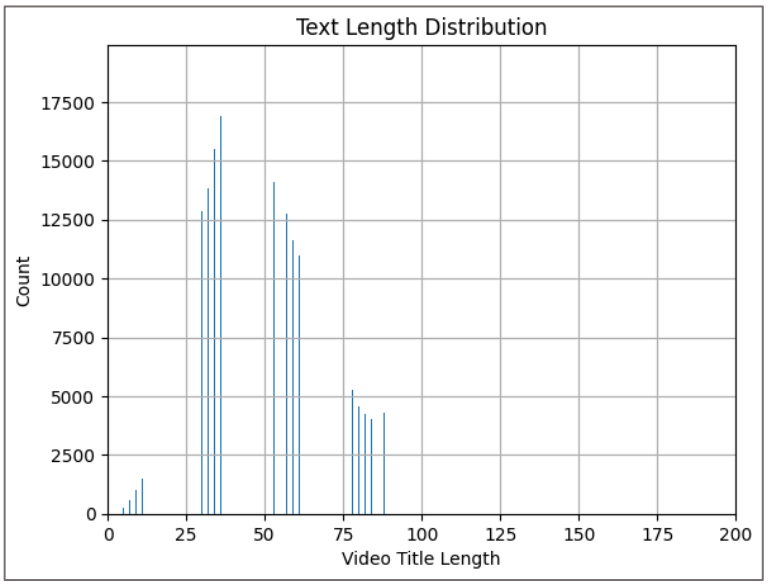


Рисунок 4.6 – Розподіл відео за довжиною назв до обробки

На рисунку 4.7 показано хмару найбільш популярних слів, що відображає частотність використання різних слів у заголовках відео на YouTube.

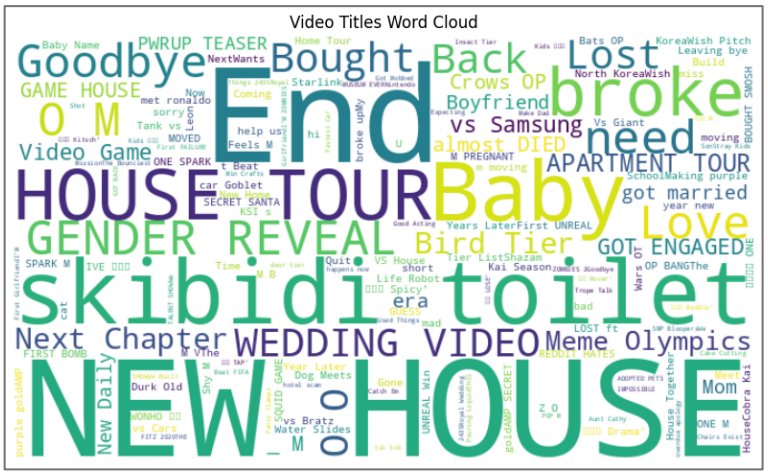


Рисунок 4.7 – Хмара найпопулярніших слів у назвах відео

У хмарі слів більший розмір шрифту вказує на вищу частоту вживання слова. Наприклад, слова «HOUSE», «TOUR», «NEW», «WEDDING» та «VIDEO» мають великий шрифт, що свідчить про їх популярність. Це показує, що відео на теми, пов’язані з домашніми турами, новинами, весіллями та відеоблогами, є дуже поширеними на платформі. Хмара слів дає швидке уявлення про найпоширеніші теми та контент, що зустрічаються в заголовках.

На рисунку 4.8 представлено графік, що показує частотність найпопулярніших слів у наборі даних. Він відображає 40 найбільш вживаних слів у заголовках відео, надаючи точнішу інформацію про їх поширеність. Наприклад, слово «mission» є найпоширенішим, зустрічається близько 800 разів, після нього йдуть слова «new», «goodbye», «house» та інші.

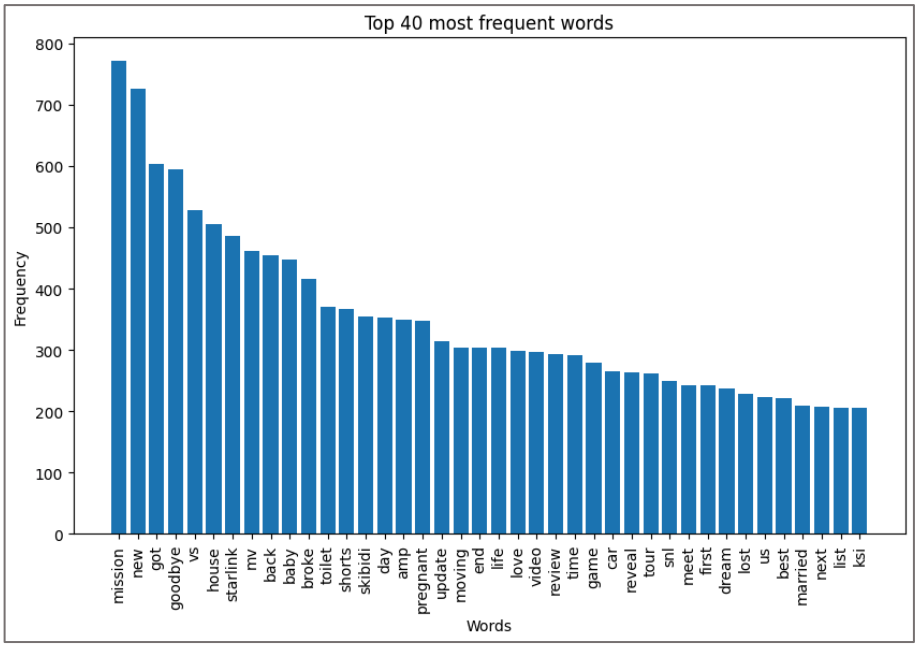


Рисунок 4.8 – Найпопулярніші слова у датасеті

Розподіл частоти категорій у датасеті, представлений на рисунку 4.9, демонструє кількість відео в різних категоріях, що дає уявлення про популярність жанрів на платформі YouTube. Найбільше відео належить до

категорії «Entertainment» (Розваги), яка містить понад 178 000 записів. Також популярними є категорії «Gaming» (Ігри) з більш ніж 152 000 відео та «Music» (Музика) з понад 117 000 записами. Категорії «People & Blogs» (Люди та блоги) і «Comedy» (Комедія) мають значну кількість відео, тоді як категорії «Shows» (Шоу) і «Nonprofits & Activism» (Неприбуткові організації та активізм) включають менше відео, що може вказувати на їх меншу популярність або рідше використання. Загалом, цей розподіл підкреслює різноманітність контенту та визначає найбільш популярні категорії серед користувачів.

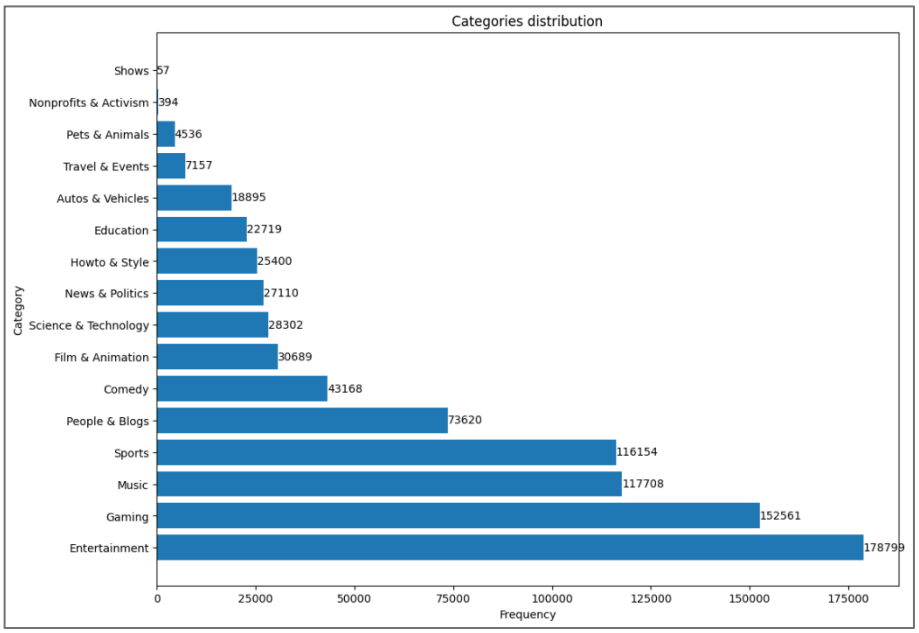


Рисунок 4.9 – Розподіл частоти категорій у датасеті

Отже, отриманий комбінований датасет є підходящим варіантом для задачі категоризації текстів, що постає в рамках даної кваліфікаційної роботи. Різноманітність категорій, таких як «Entertainment», «Gaming», «Music», а також чисельність записів у кожній категорії забезпечують модель достатньою кількістю даних для навчання та тестування. Велика кількість відео з різних тематичних напрямів дозволяє навчити модель

враховувати широкий спектр контенту, що є важливим для точної категоризації текстів на основі їх змісту.

4.3.2 Обробка даних

Процес обробки даних для класифікації текстів, отриманих з веб-сторінок, включає кілька важливих етапів, кожен з яких сприяє підвищенню якості та релевантності вхідних даних. Початковим етапом є видалення коротких текстів, зокрема заголовків відео довжина яких менше 10 символів. Це дозволяє уникнути обробки неповних або малозмістовних заголовків, які можуть погіршити результати класифікації.

Наступним кроком є видалення URL-адрес та HTML-тегів, які часто зустрічаються в заголовках. Ці елементи не мають сенсу для моделі, тому їх вилучення дозволяє сконцентруватися на корисній інформації. Цей етап значно очищує текст, підвищуючи його якість для подальшої обробки.

Після цього здійснюється нормалізація тексту, яка включає приведення всіх символів до нижнього регістру та видалення пунктуації і спеціальних символів. Це забезпечує однорідність тексту, що є важливим для коректної роботи алгоритмів обробки природної мови. Видалення пунктуації дозволяє уникнути помилкових сегментацій слів, що можуть вплинути на точність класифікації.

Далі було проведено очищення тексту від помилок і неправильних написань, а також видалення повторюваних слів. Це покращує точність аналізу, оскільки неправильні написання можуть створювати зайвий шум. Розширення скорочень також є важливим етапом, оскільки вони можуть викликати неоднозначність у тексті.

Один з ключових етапів – це видалення стоп-слів та лематизація. Використання цих методів дозволяє зменшити розмір тексту, залишаючи лише значущі слова, що допомагає моделі краще розуміти контекст і

значення. Лематизація також сприяє приведенню слів до базових форм, що зменшує варіативність і підвищує точність класифікації.

Заключним етапом є видалення зайвих пробілів і проведення частиномовного аналізу (POS-тегування). Це дозволяє виділяти важливі слова (наприклад, іменники, прикметники, дієслова), що забезпечує краще розуміння контексту та покращує якість тексту для класифікації.

На рисунку 4.10 зображені статистичні дані датасету після обробки.

	category_id	view_count	text_length
count	816324.000000	8.163240e+05	816324.000000
mean	19.021989	2.397633e+06	36.555761
std	6.649988	7.546069e+06	15.757396
min	1.000000	0.000000e+00	10.000000
25%	17.000000	4.108520e+05	25.000000
50%	20.000000	8.602970e+05	34.000000
75%	24.000000	1.980893e+06	46.000000
max	43.000000	1.407644e+09	93.000000

Рисунок 4.10 – Статистичні дані датасету після обробки

Після обробки даних кількість записів у наборі зменшилася до 816324, що свідчить про успішне видалення коротких і неінформативних заголовків. Середня довжина заголовків зросла до 36,56 символів, що підтверджує, що були видалені надто короткі заголовки, які не мали значної інформаційної цінності.

Результат обробки датасету показано на рисунку 4.11, де порівнюється кількість записів до та після обробки. Відображається значне зменшення обсягу даних після очищення, зокрема, внаслідок видалення дублікатів, непотрібних символів і незначущих слів та елементів. Зменшення кількості записів після обробки покращує якість і точність подальшого аналізу та навчання моделі.

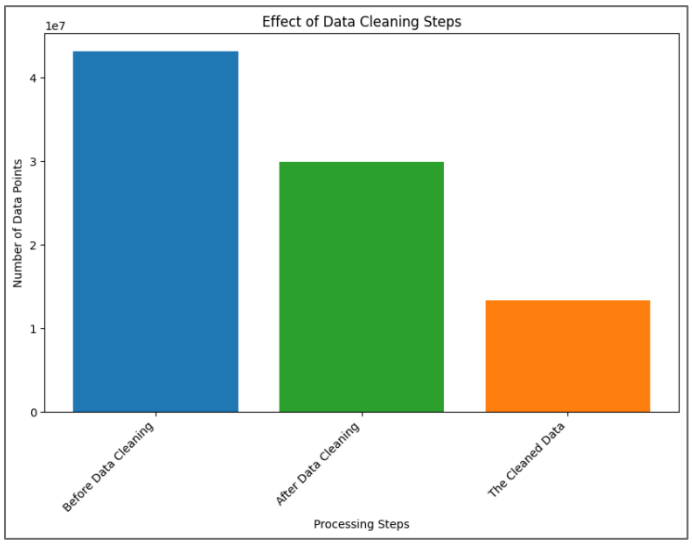


Рисунок 4.11 – Результат обробки датасету

Розподіл довжини заголовків відео після обробки, показаний на рисунку 4.12, демонструє, що більшість заголовків мають довжину від 15 до 70 символів. Це підтверджує, що етап попередньої обробки допоміг стандартизувати довжину заголовків, видаливши занадто короткі та неінформативні записи. Такий розподіл дозволяє моделі краще аналізувати та класифікувати заголовки, оскільки більш структуровані та довгі заголовки містять більше корисної інформації.

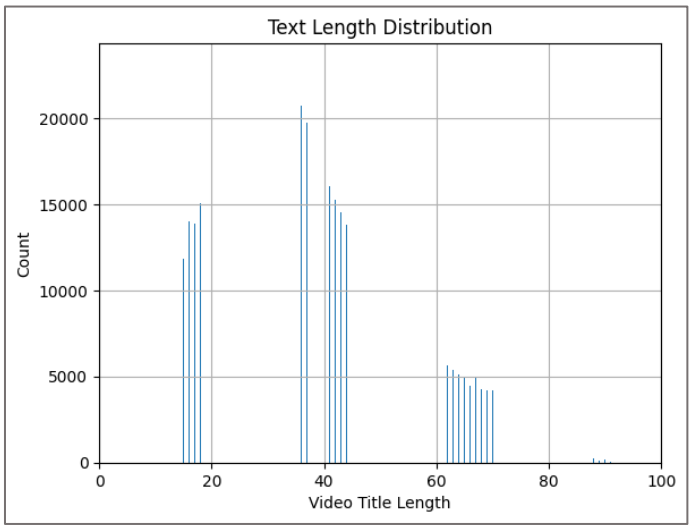


Рисунок 4.12 – Розподіл відео за довжиною назв після обробки

Після завершення обробки даних було отримано більш чистий і структурований набір, готовий до навчання моделі. Приклад оброблених даних зображено на рисунку 4.13.

	title	category_id	view_count	category_name
0	ask girlfriend	22	1514614	People & Blogs
1	apex legend stori outland endors	20	2381688	Gaming
2	left youtub month happen	24	2038853	Entertainment
3	xxl freshman class reveal offici announc	10	496771	Music
4	ultim diy home movi theater labrant famili	26	1123889	Howto & Style

Рисунок 4.13 – Приклад даних обробленого датасету

Статистичні дані та графіки підтверджують ефективність очищення даних, що сприяло підвищенню якості вхідної інформації. Це важливий крок у процесі розробки моделі, який дозволяє досягти високої точності та ефективності в класифікації текстів.

4.3.3 Розробка та тренування моделі нейронної мережі

Для класифікації контенту з веб-сторінок була розроблена нейронна мережа, яка включає кілька ключових компонентів. Початковий шар Embedding перетворює індекси слів у вектори фіксованої довжини, що покращує розпізнавання контексту. Далі йде двонаправлений GRU, що обробляє текст у двох напрямках, дозволяючи враховувати як попередні, так і наступні слова.

Шар GlobalAveragePooling1D агрегує інформацію, зменшуючи розмірність і покращуючи узагальнення моделі. Три щільні шари з активацією tanh дозволяють моделі навчатися складніших залежностей, а шар Dropout допомагає уникнути перенавчання.

Модель завершується шаром Dense з функцією softmax, що класифікує відео за категоріями. Оптимізація навчання здійснюється за

допомогою Adam з початковою швидкістю 0.001. Для підбору оптимальних параметрів використовувався метод grid search, що дозволив знайти найкращі гіперпараметри: 256 блоків GRU, 256 нейронів у щільних шарах і коефіцієнт Dropout 0.4.

Модель була натренована з використанням крос-валідації, що підвищило її точність та ефективність у класифікації відео. Архітектура нейронної мережі зображена на рисунках 4.14 і 4.15.

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 92, 16)            596528

 bidirectional_3 (Bidirecti  (None, 92, 512)           420864
 onal)

 global_average_pooling1d_3  (None, 512)               0
  (GlobalAveragePooling1D)

 dense_12 (Dense)            (None, 256)               131328

 dense_13 (Dense)            (None, 256)               65792

 dense_14 (Dense)            (None, 256)               65792

 dropout_3 (Dropout)         (None, 256)               0

 dense_15 (Dense)            (None, 44)                11308

=================================================================
Total params: 1291612 (4.93 MB)
Trainable params: 1291612 (4.93 MB)
Non-trainable params: 0 (0.00 Byte)
```

Рисунок 4.14 – Архітектура нейронної мережі

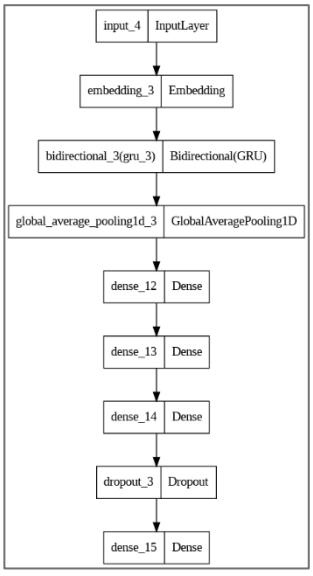


Рисунок 4.15 – Графічне зображення архітектури нейронної мережі

Після налаштування архітектури та гіперпараметрів модель була готова до навчання. Використовувався датасет, поділений на навчальну та валідаційну вибірки. Навчання проводилось протягом 12 епох із батчем розміром 256, що сприяло швидшій конвергенції.

Для уникнення перенавчання застосовувався метод ранньої зупинки (early stopping), який припиняє навчання, якщо валідаційна помилка не покращується протягом 3 епох. Параметр restore_best_weights був увімкнений, щоб відновити ваги моделі з найкращою валідаційною помилкою, забезпечуючи оптимальний результат для подальшого використання.

На рисунку 4.16 зображено процес навчання нейронної мережі.

```
Epoch 1/12
2233/2233 [==============================] - 131s 57ms/step - loss: 1.1514 - accuracy: 0.6308 - val_loss: 0.6838 - val_accuracy: 0.7902
Epoch 2/12
2233/2233 [==============================] - 106s 48ms/step - loss: 0.5577 - accuracy: 0.8306 - val_loss: 0.4481 - val_accuracy: 0.8669
Epoch 3/12
2233/2233 [==============================] - 98s 44ms/step - loss: 0.3685 - accuracy: 0.8917 - val_loss: 0.3358 - val_accuracy: 0.9025
Epoch 4/12
2233/2233 [==============================] - 97s 43ms/step - loss: 0.2587 - accuracy: 0.9264 - val_loss: 0.2506 - val_accuracy: 0.9301
Epoch 5/12
2233/2233 [==============================] - 95s 42ms/step - loss: 0.1879 - accuracy: 0.9477 - val_loss: 0.1899 - val_accuracy: 0.9496
Epoch 6/12
2233/2233 [==============================] - 94s 42ms/step - loss: 0.1419 - accuracy: 0.9611 - val_loss: 0.1468 - val_accuracy: 0.9614
Epoch 7/12
2233/2233 [==============================] - 93s 42ms/step - loss: 0.1109 - accuracy: 0.9697 - val_loss: 0.1284 - val_accuracy: 0.9679
Epoch 8/12
2233/2233 [==============================] - 94s 42ms/step - loss: 0.0878 - accuracy: 0.9763 - val_loss: 0.1031 - val_accuracy: 0.9744
Epoch 9/12
2233/2233 [==============================] - 92s 41ms/step - loss: 0.0740 - accuracy: 0.9799 - val_loss: 0.0948 - val_accuracy: 0.9762
Epoch 10/12
2233/2233 [==============================] - 93s 42ms/step - loss: 0.0630 - accuracy: 0.9830 - val_loss: 0.0840 - val_accuracy: 0.9798
Epoch 11/12
2233/2233 [==============================] - 93s 41ms/step - loss: 0.0542 - accuracy: 0.9853 - val_loss: 0.0738 - val_accuracy: 0.9834
Epoch 12/12
2233/2233 [==============================] - 92s 41ms/step - loss: 0.0515 - accuracy: 0.9859 - val_loss: 0.0749 - val_accuracy: 0.9821
```

Рисунок 4.16 – Процес навчання нейронної мережі

Після тренування модель була протестована на тестовій вибірці для оцінки її ефективності та точності класифікації жанрів відео. Результати показали високі показники: Test Loss склав 0.0778, а Test Accuracy 0.9820. Це свідчить про те, що модель ефективно узагальнює знання і здатна точно класифікувати нові, невідомі заголовки відео.

Графік точності на рисунку 4.17 демонструє постійне зростання точності, що досягає майже 98% до кінця навчання. Також точність на

валідаційній вибірці залишається близькою до точності на навчальній, що вказує на відсутність перенавчання.

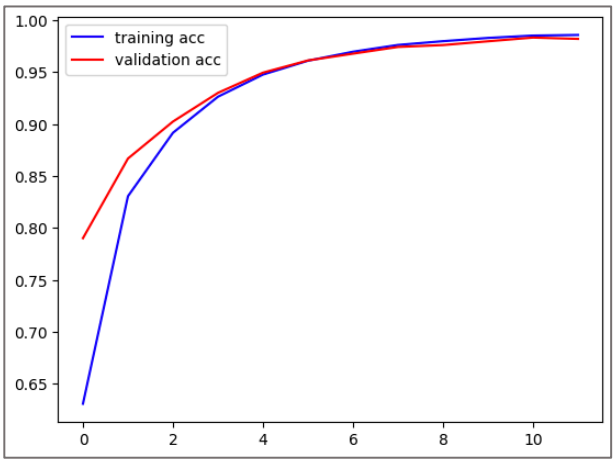


Рисунок 4.17 – Графік точності моделі в процесі навчання

Графік втрат, представлений на рисунку 4.18, показує поступове зниження втрат як на навчальній, так і на валідаційній вибірках. Це свідчить про ефективність навчання та підтверджує, що модель навчалася коректно, не зазнаючи перенавчання. Зниження валідаційних втрат до рівня, близького до навчальних, також вказує на здатність моделі добре узагальнювати знання.

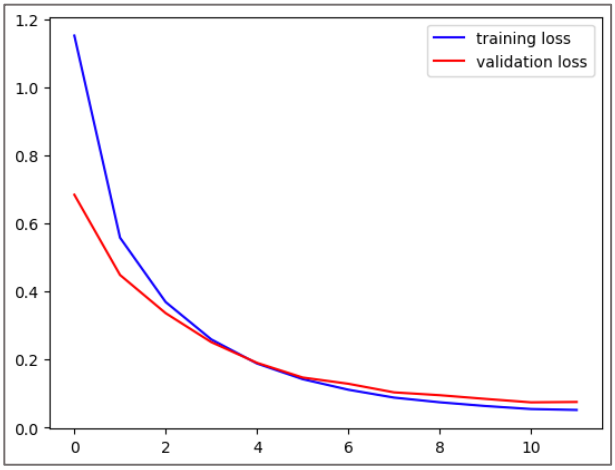


Рисунок 4.18 – Графік втрат моделі в процесі навчання

Метрики, отримані після оцінки моделі, підтверджують її високу ефективність:

– Accuracy – 0.9802, точність моделі становить 98.02%, що свідчить про здатність моделі правильно класифікувати більшість заголовків відео;

– Precision – 0.9844, точність визначає, наскільки ефективно модель ідентифікує лише правильні категорії, мінімізуючи хибнопозитивні результати;

– Recall – 0.9802, повнота вказує на здатність моделі правильно знаходити всі позитивні приклади, не пропускаючи жодної категорії;

– F1 Score – 0.9823, F1-міра є гармонічним середнім значенням точності і повноти, що забезпечує збалансовану оцінку ефективності моделі.

Ці результати демонструють високу ефективність моделі, яка точно класифікує заголовки відео та контент з них. Модель показує високу точність, здатність правильно визначати категорії та ефективно узагальнювати знання на нових даних, що робить її потужним інструментом для автоматичної класифікації контенту.

4.4 Сентиментальний аналіз тексту

Для визначення емоційного забарвлення текстового контенту у розробленій системі застосовано модель sentiment-roberta-large-english [57], яка показує гарні результати для завдання сентиментального аналізу англійською мовою. Ця модель побудована на основі архітектури RoBERTa Large та була додатково навчена на великій кількості анотованих даних, що забезпечило високу точність і стабільність результатів.

Основне завдання моделі полягає у класифікації тексту за двома класами: позитивний та негативний сентимент. Завдяки попередньому перекладу українських текстів на англійську, модель ефективно

використовується для мультимовного аналізу в рамках розроблюваної системи.

Важливою перевагою моделі є її висока продуктивність за метриками Accuracy (точність), Precision (точність позитивних передбачень), Recall (повнота) та F1-Score. В офіційних тестах модель досягає Accuracy = 98.0% на стандартних тестових наборах [57], що свідчить про її високу якість у порівнянні з іншими доступними рішеннями.

Для оцінки продуктивності моделі на реальних даних було проведено експеримент із використанням відкритого датасету IMDb Movie Reviews Dataset [58], який містить тисячі рецензій на фільми із зазначеним сентиментом.

На рисунку 4.19 зображено результати тестування моделі sentiment-roberta-large-english. Ці результати підтверджують високу якість моделі при класифікації сентименту в текстах різного обсягу та тематики.

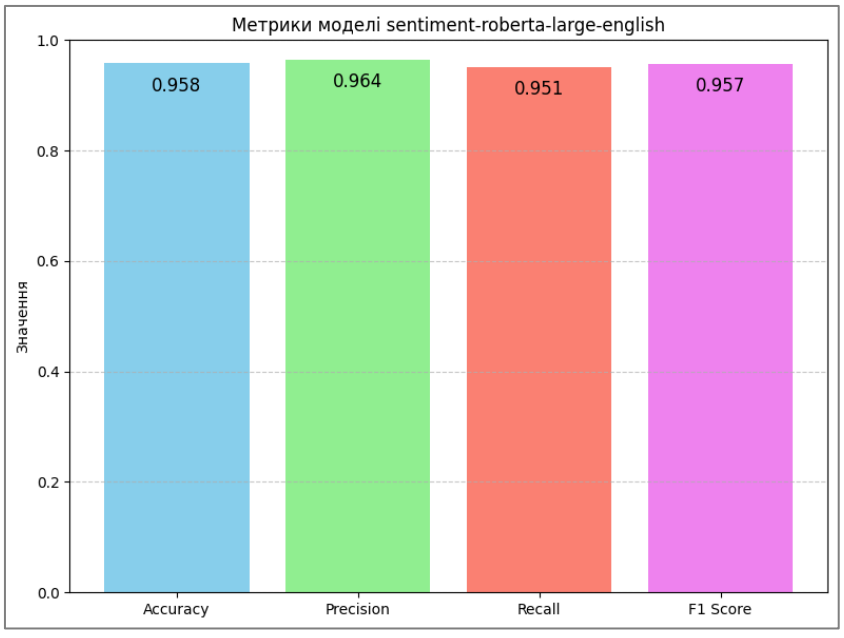


Рисунок 4.19 – Значення метрик Accuracy, Precision, Recall та F1-Score для моделі sentiment-roberta-large-english

На рисунку 4.20 зображено матрицю невідповідностей для передбачень моделі на тестовому наборі даних IMDb. Дана матриця показує високу кількість вірних позитивних та негативних передбачень із мінімальною кількістю помилок.

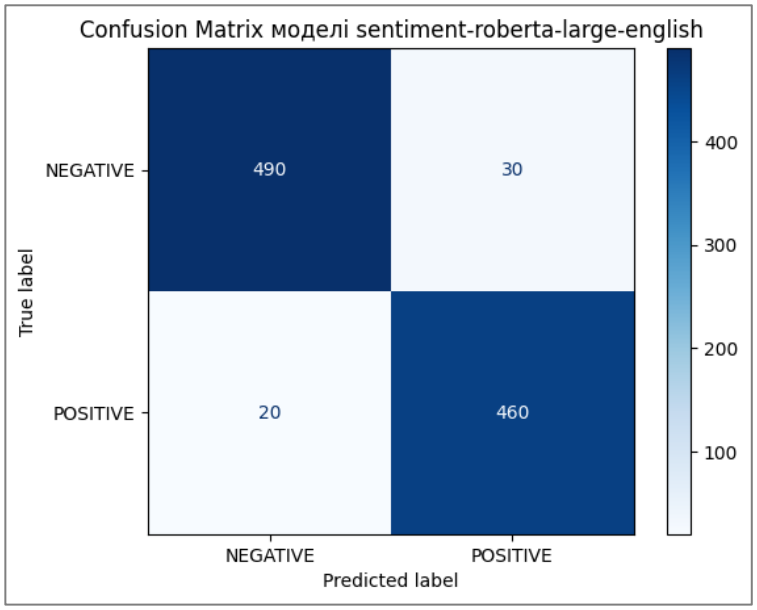


Рисунок 4.20 – Confusion Matrix для передбачень моделі на тестовому наборі IMDb

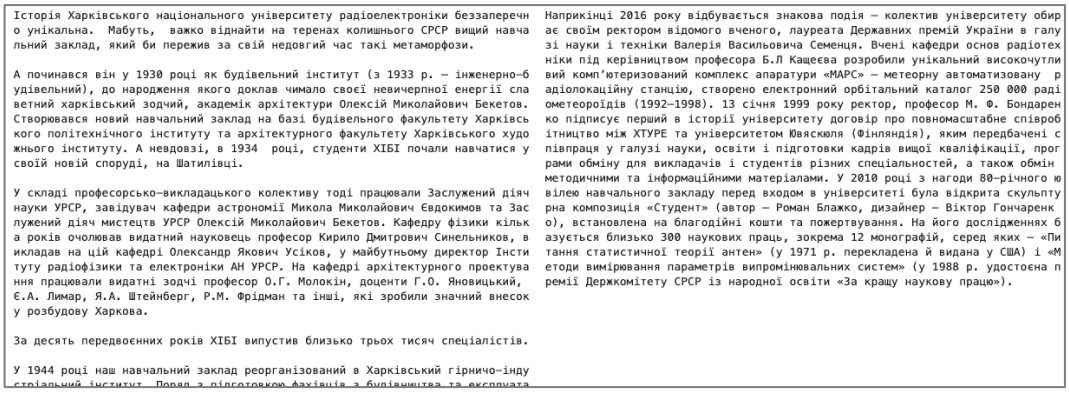
Отримані метрики свідчать про здатність моделі ефективно розпізнавати сентимент навіть у складних і неоднозначних текстах. Це особливо важливо для запропонованої системи, де тексти новинних статей, блогів або аналітичних матеріалів можуть мати змішані емоційні відтінки. Стабільність та висока якість результатів дозволяють використовувати обрану модель як надійний компонент для етапу емоційного аналізу в автоматичній генерації відео.

Таким чином, використання цієї моделі значно підвищує релевантність підбору музичного супроводу та візуальних ефектів відповідно до емоційного змісту тексту, що є важливою складовою створення привабливого відеоконтенту.

4.5 Створення короткого змісту для контенту з веб-сторінок

У процесі автоматизації створення відео важливим кроком є формування короткого змісту (реферування) текстових матеріалів. Це дозволяє суттєво зменшити обсяг вхідного тексту, виділити ключові моменти та адаптувати інформацію для подальшої генерації відеоконтенту. У межах даної роботи застосовано два підходи до створення короткого змісту тексту: екстрактивне та абстрактивне.

Екстрактивне реферування відбувається шляхом виділення найбільш інформативних речень із початкового тексту. Для цього реалізовано алгоритм, що базується на методі TextRank. Попередньо текст розбивається на речення за допомогою бібліотеки nltk. Для кожної пари речень обчислюється подібність на основі косинусної міри. На основі отриманої матриці подібності будується граф суміжності, для якого визначаються ваги вершин за алгоритмом PageRank. Найвищі ваги відповідають реченням із найбільшим інформаційним внеском. Результатом є зведений текст, що зберігає найважливіші факти початкового матеріалу. На рисунку 4.21 зображено приклад роботи екстрактивного підходу для статті з веб-сайту Харківського національного університету радіоелектроніки.


Історія Харківського національного університету радіоелектроніки беззаперечн
о унікальна. Мабуть, важко віднайти на теренах колишнього СРСР вищий навча
льний заклад, який би пережив за свій недовгий час такі метаморфози.

А починався він у 1930 році як будівельний інститут (з 1933 р. — інженерно-б
удівельний), до народження якого доклав чимало своєї невичерпної енергії сла
ветний харківський зодчий, академік архітектури Олексій Миколайович Бекетов.
Створювався новий навчальний заклад на базі будівельного факультету Харківсь
кого політехнічного інституту та архітектурного факультету Харківського худо
жнього інституту. А невдовзі, в 1934 році, студенти ХІБІ почали навчатися у
своїй новій споруді, на Шатилівці.

У складі професорсько-викладацького колективу тоді працювали Заслужений діяч
науки УРСР, завідувач кафедри астрономії Микола Миколайович Євдокимов та Зас
лужений діяч мистецтв УРСР Олексій Миколайович Бекетов. Кафедру фізики кільк
а років очолював видатний науковець професор Кирило Дмитрович Синельников, в
икладав на цій кафедрі Олександр Якович Усіков, у майбутньому директор Інсти
туту радіофізики та електроніки АН УРСР. На кафедрі архітектурного проектува
ння працювали видатні зодчі професор О.Г. Молокін, доценти Г.О. Яновицький,
Є.А. Лимар, Я.А. Штейнберг, Р.М. Фрідман та інші, які зробили значний внесок
у розбудову Харкова.

За десять передвоєнних років ХІБІ випустив близько трьох тисяч спеціалістів.

У 1944 році наш навчальний заклад реорганізований в Харківський гірничо-інду
стріальний інститут. Поряд з підготовкою фахівців з будівництва та експлуата
Наприкінці 2016 року відбувається знакова подія — колектив університету обир
ає своїм ректором відомого вченого, лауреата Державних премій України в галу
зі науки і техніки Валерія Васильовича Семенця. Вчені кафедри основ радіотех
ніки під керівництвом професора Б.Л Кащеєва розробили унікальний високочутли
вий комп'ютеризований комплекс апаратури «МАРС» — метеорну автоматизовану р
адіолокаційну станцію, створено електронний орбітальний каталог 250 000 раді
ометеороїдів (1992–1998). 13 січня 1999 року ректор, професор М. Ф. Бондарен
ко підписує перший в історії університету договір про повномасштабне співроб
ітництво між ХТУРЕ та університетом Ювяскюля (Фінляндія), яким передбачені с
півпраця у галузі науки, освіти і підготовки кадрів вищої кваліфікації, прог
рами обміну для викладачів і студентів різних спеціальностей, а також обмін
методичними та інформаційними матеріалами. У 2010 році з нагоди 80-річного ю
вілею навчального закладу перед входом в університеті була відкрита скульпту
рна композиція «Студент» (автор — Роман Блажко, дизайнер — Віктор Гончаренк
о), встановлена на благодійні кошти та пожертвування. На його дослідженнях б
азується близько 300 наукових праць, зокрема 12 монографій, серед яких — «Пи
тання статистичної теорії антен» (у 1971 р. перекладена й видана у США) і «М
етоди вимірювання параметрів випромінювальних систем» (у 1988 р. удостоєна п
ремії Держкомітету СРСР із народної освіти «За кращу наукову працю»).

Рисунок 4.21 – Приклад роботи екстрактивного підходу для статті з веб-сайту університету

Для створення абстрактивного реферування тексту застосовано сучасну модель на основі трансформерів – facebook/bart-large-cnn [59]. Ця модель використовує енкодер-декодерну архітектуру та навчена на великому корпусі CNN/DailyMail [60], що дозволяє їй генерувати нові речення, які стисло й граматично коректно переказують зміст вхідного тексту. На відміну від екстрактивного підходу, абстрактивне реферування не обмежується вибором існуючих речень, а створює власні формулювання. Результати роботи абстрактивного підходу зображені на рисунку 4.22.

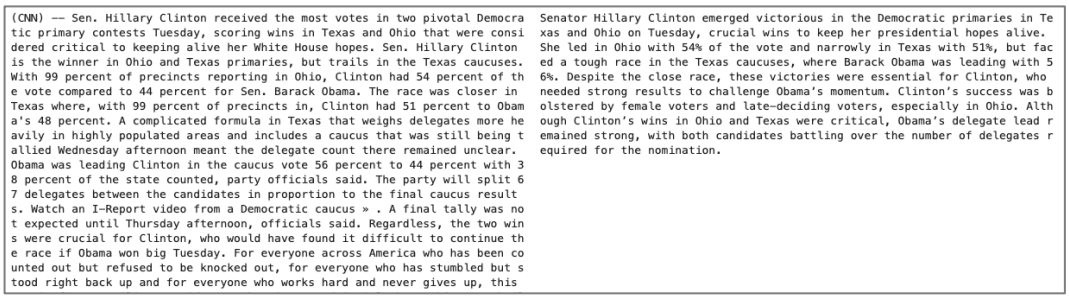


Рисунок 4.22 – Приклад роботи абстрактивного підходу на основі моделі BART

Для оцінки якості роботи моделі було проведено експеримент на популярному датасеті CNN/DailyMail [60], який є стандартом для задач реферування. З вибірки було відібрано 500 статей, до яких застосовано екстрактивний і абстрактивний підходи.

Результати тестування екстрактивного підходу свідчать про те, що модель змогла зберегти деяку кількість важливих одиничних слів (ROUGE-1), але значно слабше впоралася з відтворенням складніших конструкцій, таких як двослівні фрази (ROUGE-2), та зберіганням структури оригінального тексту (ROUGE-L):

– ROUGE-1 (0.287) – результат вказує на те, що екстрактивний підхід зміг виділити 28.7% одиничних слів з референсного тексту. Це

означає, що основні факти і важливі слова були збережені, але алгоритм не зміг повністю зберегти всю інформацію;

– ROUGE-2 (0.076) – значення цієї метрики є досить низьким, що свідчить про те, що алгоритм не зміг ефективно зберігати більш складні двослівні конструкції, які можуть бути важливими для передачі точного змісту;

– ROUGE-L (0.138) – це значення також є низьким, що вказує на те, що структура тексту була значно змінена в згенерованому короткому змісті порівняно з референсним текстом. Алгоритм не зміг точно відобразити порядок слів та фраз.

Результати для абстрактивного підходу з використанням моделі facebook/bart-large-cnn показують більш високі значення метрик, що свідчить про те, що цей підхід зміг зберегти більше важливих ідей і структури тексту, навіть якщо не точні слова були використані:

– ROUGE-1 (0.448) – значення 44.8% вказує на те, що абстрактивний підхід добре зберіг основні слова та концепти. Хоча це й не так високо, як можна було б очікувати, але для цього підходу це доволі непогано;

– ROUGE-2 (0.223) – це значення показує, що абстрактивний підхід зміг зберегти певну кількість двослівних фраз, що є важливими для збереження точності та змісту тексту;

– ROUGE-L (0.394) – для абстрактивного реферування це досить хороше значення, оскільки воно вказує на те, що структура тексту була збережена в більшій мірі, а зміст відповідає референсному тексту, навіть якщо точні слова були змінені.

На рисунку 4.23 зображено графік порівняння метрик ROUGE для екстрактивного та абстрактивного підходу реферування тексту.

Отримані результати свідчать, що абстрактивна модель BART забезпечує вищу якість реферування за всіма показниками ROUGE порівняно з екстрактивним підходом. Особливо це помітно для метрики

ROUGE-L, яка враховує послідовність фраз і є критично важливою для нашої задачі, де важлива логічна цілісність тексту.

Екстрактивний підхід, хоча і поступається за метриками, має свої переваги – він швидший та не вимагає потужних обчислювальних ресурсів, тому може використовуватись як проміжний етап для первинного скорочення тексту перед застосуванням абстрактивного реферування. На рисунку 4.23 представлено порівняльний графік метрик ROUGE для екстрактивного та абстрактивного підходу.

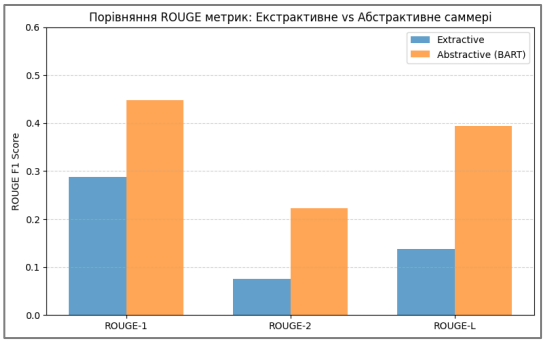


Рисунок 4.23 – Графік порівняння метрик ROUGE
для екстрактивного та абстрактивного підходу

На рисунку 4.24 зображено приклад порівняння обох підходів до реферування для тексту веб-сторінки з сайту Харківського національного університету радіоелектроніки.

Наприкінці 2016 року відбувається знакова подія – колектив університету обирає своїм ректором відомого вченого, лауреата Державних премій України в галузі науки і техніки Валерія Васильовича Семенця. Вчені кафедри основ радіотехніки під керівництвом професора Б.Л Кащєєва розробили унікальний високочутливий комп'ютеризований комплекс апаратури «МАРС» – метеорну автоматизовану радіолокаційну станцію, створено електронний орбітальний каталог 250 000 радіометеороїдів (1992–1998). 13 січня 1999 року ректор, професор М. Ф. Бондаренко підписує перший в історії університету договір про повномасштабне співробітництво між ХТУРЕ та університетом Ювяскюля (Фінляндія), яким передбачені співпраця у галузі науки, освіти і підготовки кадрів вищої кваліфікації, програми обміну для викладачів і студентів різних спеціальностей, а також обмін методичними та інформаційними матеріалами. У 2010 році з нагоди 80-річного ювілею навчального закладу перед входом в університеті була відкрита скульптурна композиція «Студент» (автор – Роман Блажко, дизайнер – Віктор Гончаренко), встановлена на благодійні кошти та пожертвування. На його дослідженнях базується близько 300 наукових праць, зокрема 12 монографій, серед яких – «Питання статистичної теорії антен» (у 1971 р. перекладена й видана у США) і «Методи вимірювання параметрів випромінювальних систем» (у 1988 р. удостоєна премії Держкомітету СРСР із народної освіти «За кращу наукову працю»).

Харківський національний університет радіоелектроніки (ХНУРЕ) має унікальну та багатогранну історію, еволюціонуючи від заснування як будівельний інститут у 1930 році до становлення провідним закладом у галузі радіоелектроніки та інформаційних технологій. Спочатку університет зосереджувався на будівництві та гірничому інженерії, але незабаром розширився, включивши радіоелектроніку та обчислювальну техніку, створивши різноманітні кафедри та факультети, присвячені цим напрямкам. Протягом років університет значно сприяв розвитку наукових досліджень, зокрема в галузях радіотехніки, автоматизації та обчислювальної техніки, а такі видатні науковці, як Б.Л. Кащєєв та Ю.І. Волощук, очолили новаторські проекти в метеорології та радіоелектроніці. В кінці 20-го та на початку 21-го століть ХНУРЕ здобув визнання як на національному, так і на міжнародному рівнях, посилюючи академічні партнерства та впроваджуючи нові програми у відповідь на технологічні досягнення. Сьогодні університет продовжує сприяти інноваціям, розширюючи свій вплив через глобальні співпраці, сучасні дослідження та підготовку у передових технологіях, таких як кібербезпека та телекомунікації.

Рисунок 4.24 – Порівняння згенерованих текстів двома методами для
однієї з протестованих статей

Для абстрактивного реферування було виконано переклад на англійську мову за допомогою моделі Helsinki-NLP, після чого було застосовано модель facebook/bart-large-cnn, і потім результат було перекладено знову на українську мову. Це робиться так, оскільки дана модель підтримує лише тексту англійською мовою.

У рамках реалізації було визначено, що використання моделі facebook/bart-large-cnn є доцільним для завдань автоматизації узагальнення змісту текстів із веб-сторінок. Модель показала стабільно високі результати та здатність генерувати зрозумілі та компактні тексти, які чудово підходять для подальшого мультимедійного опрацювання.

4.6 Використання BERT для виявлення іменованих сутностей

Виявлення іменованих сутностей (Named Entity Recognition, NER) є одним із ключових етапів аналізу текстового контенту в межах запропонованої системи. Основною метою цієї задачі є виділення важливих об’єктів у тексті, таких як імена людей, організації, географічні назви, дати тощо. У контексті даної роботи іменовані сутності використовуються для подальшого підсвічування цієї інформації у тексті сцен відео, що дозволяє користувачу вручну підібрати відповідне зображення.

Процес виявлення іменованих сутностей у системі реалізовано з використанням попередньо натренованих багатомовних трансформерних моделей сімейства RoBERTa. Для української мови застосовано модель EvanD/xlm-roberta-base-ukrainian-ner-ukrner [61], а для англійської – Davlan/xlm-roberta-base-ner-hrl [62]. Обидві моделі побудовані на основі XLM-RoBERTa – трансформерної моделі, що забезпечує високу якість багатомовної обробки тексту.

На рисунках 4.25 і 4.26 зображено приклади тестування обраних моделей для українського та англійського текстів відповідно.

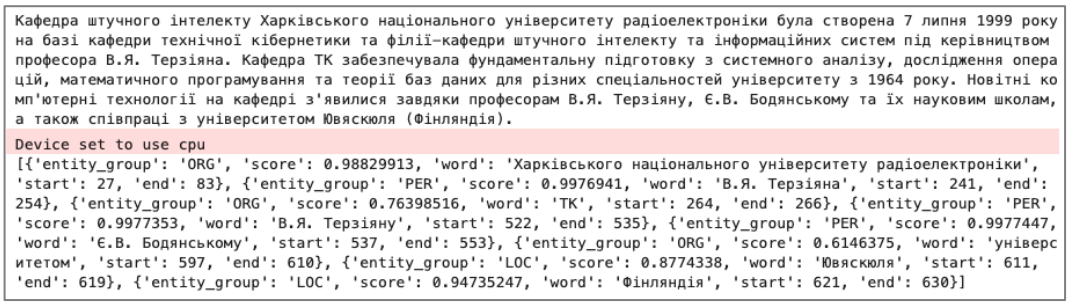


Рисунок 4.25 – Приклад виявлення іменованих сутностей за допомогою моделі EvanD/xlm-roberta-base-ukrainian-ner-ukrner

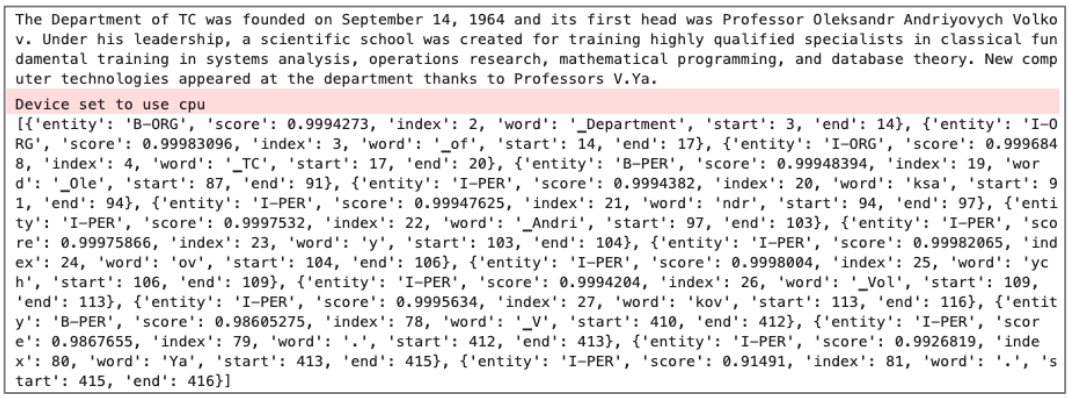


Рисунок 4.26 – Приклад виявлення іменованих сутностей за допомогою моделі Davlan/xlm-roberta-base-ner-hrl

Структура відповідей цих моделей відрізняється, тому для подальшого використання отримані результати приводяться до єдиного вигляду, використовуються лише такі типи сутностей, як: PER (персона), ORG (організація) та LOC (локація).

4.7 Підбір відповідного мультимедійного контенту

Після завершення попередніх етапів обробки тексту – категоризації, визначення сентименту, реферування та виявлення іменованих сутностей – система переходить до формування візуального супроводу для майбутнього відео. Основна задача цього етапу полягає у підборі

релевантного відеоконтенту для кожної сцени, сформованої на основі короткого змісту тексту.

Для кожного речення короткого змісту, яке відповідає окремій сцені відео, система здійснює виділення ключових слів. Цей процес реалізовано з використанням моделі KeyBERT, яка базується на сучасних мовних моделях трансформерного типу та дозволяє знаходити найбільш релевантні ключові слова чи фрази в межах заданого контексту.

Виділені ключові слова використовуються як запити до API сервісу Pexels, що надає доступ до великої колекції стокових відео. Для кожної сцени формується окремий пошуковий запит на основі отриманих ключових слів. Це дозволяє підібрати унікальний відеоряд, що відповідає змісту конкретного речення короткого змісту.

На рисунку 4.27 наведено приклад виділення ключових слів за допомогою моделі KeyBERT.

```
Харківський національний радіоелектроніки ХНУРЕ провідним навчальним закладом Україні який спеціалізується на
прикладних інформаційних та інноваціях університеті надається увага інтеграції інженерних та інформаційних
технологій іншими також співпраці бізнесом промисловістю та суспільством ХНУРЕ готує фахівців для успішної цифрової
трансформації та світу маючи сучасну матеріально технічну базу факультетів 33 кафедри та навчаючи близько тисяч
студентів них близько 600 іноземців

 [('цифрової трансформації', 0.4524),
  ('університеті надається', 0.4535),
  ('хнуре провідним', 0.4546),
  ('харківський', 0.4702),
  ('технологій іншими', 0.4928),
  ('провідним вищим', 0.496),
  ('прикладних інформаційних', 0.5087),
  ('університет радіоелектроніки', 0.5219),
  ('харківський національний', 0.5568),
  ('інформаційних технологіях', 0.5587)]
```

Рисунок 4.27 – Виділення ключових слів за допомогою KeyBERT

Для підвищення надійності системи та забезпечення повноти відеоряду реалізовано механізм fallback-пошуку. На цьому етапі система генерує додатковий запит до API Pexels, використовуючи основні ключові слова, виділені з усього тексту короткого змісту. Відео, отримані за цим запитом, зберігаються в окремому списку і використовуються у тих випадках, коли для певної сцени не вдалося знайти релевантний відеоматеріал.

Такий підхід дозволяє забезпечити максимальну релевантність і різноманітність відеоконтенту, водночас мінімізуючи ймовірність виникнення сцен без візуального супроводу.

На рисунку 4.28 зображено результати пошуку відео за допомогою Pexels API та з використанням ключових слів, отриманих через модель KeyBERT.

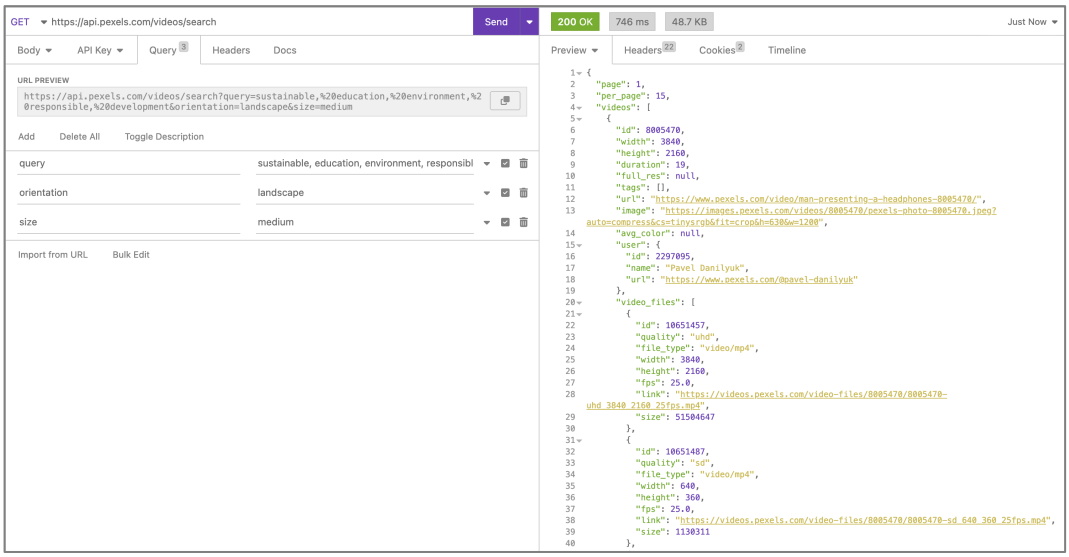


Рисунок 4.28 – Приклад запиту на Pexels API для отримання відео

Таким чином, використання KeyBERT та Pexels API забезпечує високу гнучкість і масштабованість системи, що дозволяє адаптувати її для роботи з текстами різної тематики та структури.

4.8 Озвучення відео синтезованим голосом

Озвучення є важливою складовою створення відеоконтенту, оскільки воно не лише передає інформацію, але й впливає на сприйняття та емоційне залучення глядачів. У межах цієї кваліфікаційної роботи для генерації голосового супроводу було обрано використання сучасних

нейромережевих моделей StyleTTS2, які зарекомендували себе як одні з найефективніших моделей для озвучення тексту (Text-to-Speech, TTS).

Для генерації українськомовного голосу застосовано модель patriotyk/styletts2_ukrainian_single [63], спеціально навченої на українському мовному корпусі. Ця модель забезпечує високу якість синтезованого мовлення з природнім звучанням та правильною інтонацією, що особливо важливо для сприйняття україномовної аудиторії. Для англомовного контенту було обрано модель hexgrad/Kokoro-82M [64], яка демонструє чудові результати генерації голосу в англійській мові, забезпечуючи чіткість вимови, природність і широкий діапазон емоційного забарвлення.

Для забезпечення високої швидкості та якості генерації озвучення обидві моделі були розгорнуті та виконувались на графічному процесорі (GPU). Це дозволило значно скоротити час синтезу навіть для великих обсягів тексту. Проведені експерименти показали, що середній час генерації однієї хвилини аудіо на GPU складав приблизно 8-20 секунд залежно від довжини тексту та складності інтонаційного малюнка. Для порівняння, генерація на CPU займала у 5-8 разів більше часу, що робило б неможливою інтерактивну або потокову обробку запитів у рамках запропонованої системи.

Таким чином, використання моделей StyleTTS2 забезпечило не лише якісне озвучення відео, а й дозволило досягти гнучкості та масштабованості рішення, що є критично важливим для автоматизованої генерації мультимедійного контенту.

4.9 Генерація музичного супроводу

Для створення цілісного та емоційно узгодженого відео важливо не лише підібрати релевантний візуальний контент, а й забезпечити якісний музичний супровід, який підкреслює основний настрій і тематику відео. У

межах цієї роботи реалізовано автоматизований підхід до генерації музики за допомогою сучасних нейромережевих технологій.

Основною моделлю для генерації музики обрано Facebook MusicGen – потужну трансформерну модель, спеціально розроблену для створення музичних композицій за текстовими описами (промптами). MusicGen здатна синтезувати високоякісні аудіотреки різних жанрів та стилів, враховуючи вказані користувачем параметри, зокрема емоційний настрій, жанрову приналежність та додаткові ключові слова.

Для забезпечення максимальної релевантності створюваного музичного супроводу, у системі застосовано багатоступеневий підхід до формування текстового запиту (промпту) для MusicGen. Основними джерелами інформації для промпту виступають визначений на попередніх етапах сентимент тексту, категорія контенту та ключові слова, виділені під час аналізу короткого змісту. Ці параметри дозволяють максимально точно відобразити зміст і емоційний тон відео у музичному супроводі.

Щоб автоматизувати створення промптів, було використано мовну модель TinyLLaMA (1.1B) [65], яка є полегшеною версією великих LLM і чудово підходить для генерації коротких текстових запитів за заданими параметрами. Для TinyLLaMA був розроблений спеціальний промпт шаблон, зображений на рисунку 4.29.

```
from langchain_ollama import OllamaLLM
from langchain_core.prompts import ChatPromptTemplate

llm = OllamaLLM(model="tinyllama")

template = ChatPromptTemplate.from_messages([
    ("system", "You are an AI tasked with generating a detailed music description based on provided text categories, sentiment, and keywords. "
               "Please follow the instructions carefully."),
    ("user", "Based on the following information: "
             "Category: {category}, Sentiment: {sentiment}, Keywords: {keywords}, "
             "generate the following music details:"
             "1. **Tempo:** Describe the tempo of the music (e.g., fast, slow, medium). "
             "2. **Rhythm:** Describe the rhythm of the music (e.g., syncopated, smooth, steady). "
             "3. **Instruments:** List the musical instruments that should be used in the music (e.g., piano, guitar, drums, synth). "
     "Provide this information in a JSON structure: tempo (string), rhythm (string), instruments (array). Return only this JSON."
    )
])
```

Рисунок 4.29 – Створений промпт для TinyLLaMA для генерації промпту для моделі Facebook MusicGen

В результаті TinyLLaMA повертає JSON з необхідною інформацією – темпом, ритмом та музичними інструментами, після чого JSON перетворюється у рядковий тип даних і формується промпт, який передається до моделі MusicGen, яка зі свого боку генерує музичний фрагмент відповідної тривалості (у цій роботі встановлено стандартну тривалість 30–60 секунд для фону відео). У разі потреби можливе подальше корегування промпту для отримання альтернативних варіантів композицій.

У процесі тестування цей підхід продемонстрував високу релевантність отриманих музичних треків до заданого настрою та змісту відео. Використання генеративної моделі MusicGen у поєднанні з адаптивними промптами від TinyLLaMA дозволило досягти гнучкості та індивідуалізації музичного супроводу без необхідності ручного підбору треків або звернення до сторонніх музичних бібліотек.

4.10 Формування фінального відеофайлу

Формування фінального відеофайлу є заключним етапом роботи системи, де всі попередньо підготовлені компоненти об’єднуються у єдиний мультимедійний продукт. Цей процес реалізовано у спеціально створеному модулі VideoGenerator, що побудований із застосуванням бібліотеки MoviePy. Основна його задача – об’єднати відео для кожної сцени, накласти текстові описи, додати озвучення та фонову музику, забезпечивши при цьому узгодженість усіх елементів і високу якість фінального відео.

Кожна сцена створюється на основі відеофрагменту, що масштабується або циклічно повторюється для відповідності тривалості аудіо озвучення. Текстовий опис додається як окремий візуальний елемент з анімованим ковзанням у кадрі. Якщо у сцені передбачено зображення, наприклад, портрет особи або логотип організації, воно розміщується над

текстовим блоком і супроводжується підписом. Усі ці елементи поєднуються у єдиний композиційний кліп. Для забезпечення якісного візуального відображення відео автоматично підганяється за розміром до обраної орієнтації – альбомної або портретної.

Після створення окремих сцен відбувається їх послідовне об’єднання у єдиний відеофайл. До фінального відео додається фоновий музичний супровід, що циклічно повторюється і мікшується з голосовим озвученням сцен. За потреби на відео додається логотип користувача або компанії, розташований у визначеному місці та збережений протягом усієї тривалості відео.

На рисунку 4.30 зображено схему комбінування усіх частин для сцен у фінальне відео.

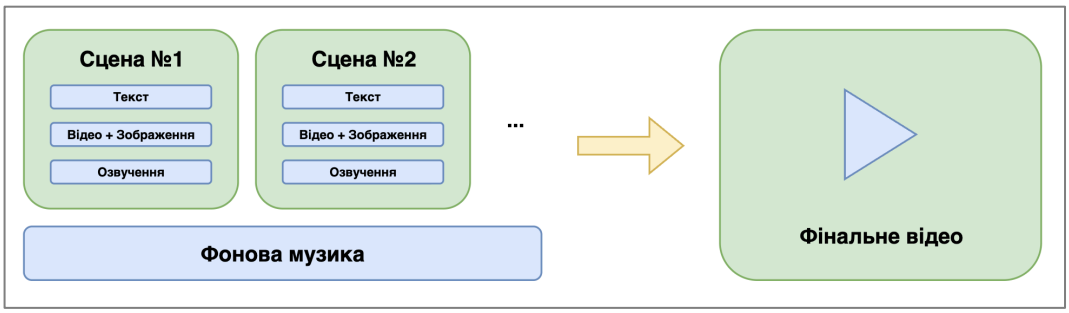


Рисунок 4.30 – Схема комбінування усіх частин для сцен

Система підтримує створення відео у різних форматах якості – від 360p до 1080p – з відповідним налаштуванням параметрів. Окрім цього, реалізовано можливість автоматичного створення ескізів (thumbnail) для зручного попереднього перегляду відео на веб-платформах. Для різних форматів якості й орієнтацій передбачено окремі профілі налаштувань, що дозволяє оптимізувати процес генерації відео та покращити користувацький досвід.

4.11 Розробка веб-додатку

Для забезпечення зручного доступу користувачів до функціональності системи та ефективного управління процесами генерації відеоконтенту було розроблено повнофункціональний веб-додаток. Основною метою створення веб-додатку стало об’єднання всіх компонентів системи в єдиний інтерфейс, що дозволяє виконувати завантаження посилань на веб-сторінки, перегляд результатів обробки тексту, редагування сцен, перегляд згенерованого відео та управління особистими даними користувача.

Веб-додаток реалізовано за клієнт-серверною архітектурою, що забезпечує масштабованість і гнучкість рішення. Серверна частина відповідає за обробку запитів, управління даними та взаємодію з мікросервісами для інтелектуальної обробки тексту та створення відео. Клієнтська частина забезпечує користувачеві інтуїтивно зрозумілий інтерфейс для взаємодії із системою.

4.11.1 Реалізація серверної частини на Laravel

Серверна частина веб-додатку була реалізована з використанням популярного PHP-фреймворку Laravel, що забезпечує зручну структуру проєкту, високу продуктивність та велику кількість готових рішень для розробників. Laravel дозволив ефективно реалізувати REST API для взаємодії клієнтської частини з сервером, а також організувати складну бізнес-логіку додатку.

Для автентифікації та авторизації користувачів був застосований пакет laravel/sanctum [66], який забезпечує безпечну аутентифікацію за допомогою токенів для API. Це дозволяє користувачам отримувати доступ до своїх ресурсів та взаємодіяти із системою через клієнтський інтерфейс або інші інтеграції.

З метою дотримання принципів чистої архітектури та спрощення роботи з вхідними та вихідними даними була використана бібліотека spatie/laravel-data [67]. Вона забезпечила реалізацію патерну Data Transfer Object (DTO), що дозволило структурувати дані та зменшити залежності між різними частинами коду. Крім того, для деяких складних типів даних було застосовано підхід Value Objects, що підвищило надійність і передбачуваність роботи із даними.

Для обробки завдань, що потребують значних ресурсів або тривалого часу виконання (наприклад, надсилання запитів на API мікросервісів чи відправка листів на електронну пошту користувача), було використано вбудований у Laravel механізм черг (queues). Черги дозволяють виконувати задачі у фоновому режимі, не блокуючи основний потік запитів користувачів. У якості драйвера черг був обраний Redis, який забезпечує високу швидкодію та масштабованість. Для керування робочими процесами черг на сервері був налаштований процесний менеджер Supervisor, що гарантує автоматичний контроль і перезапуск воркерів у випадку помилок.

Взаємодія між серверною частиною і мікросервісом інтелектуальної частини відбувається шляхом використання механізму вебхуків (англ. webhooks). Такий підхід дозволяє забезпечити асинхронну взаємодію між компонентами системи, уникнути блокування основного потоку запитів та підвищити загальну продуктивність додатку.

Після отримання відповіді через вебхук серверна частина обробляє отримані дані, оновлює відповідні записи у базі даних та повідомляє клієнтську частину про успішне завершення операції. Це дозволяє користувачу своєчасно отримувати оновлену інформацію про статус завдань, наприклад, завершення обробки тексту, генерації короткого змісту, категоризації, пошуку мультимедійного контенту чи формування фінального відео.

Використання вебхуків у поєднанні з механізмом черг і мікросервісною архітектурою забезпечує масштабованість системи, гнучкість у розподілі обчислювальних навантажень та можливість легкого додавання нових функціональних можливостей у майбутньому.

Окремим важливим компонентом серверної частини є адміністративна панель, яка була створена за допомогою пакету Filament [68]. Цей сучасний адмін-фреймворк для Laravel дозволяє швидко створювати потужні адміністративні інтерфейси з гнучкими налаштуваннями доступу та приємним дизайном. В адміністративній панелі реалізовано управління користувачами, веб-сторінками, що були оброблені системою, та відео, які були створені користувачами.

На рисунку 4.31 зображено сторінку перегляду детальної інформації про веб-сторінки, що були проаналізовані інтелектуальною частиною, збережені в базу даних і які використовуються для подальшого створення відео.

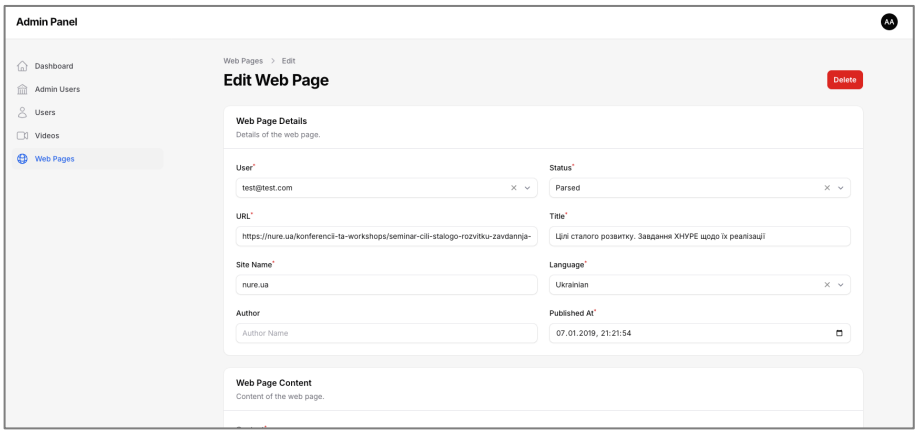


Рисунок 4.31 – Сторінка перегляду детальної інформації про веб-сторінки та результат їхнього аналізу

На рисунку 4.32 зображено сторінку перегляду списку відео у системі. На даній сторінці є можливість переглядати статуси відео і виявляти проблеми при створенні відео (завдяки відповідному статусу з

помилкою). При кліку на певне відео, відкривається більш детальна інформація.

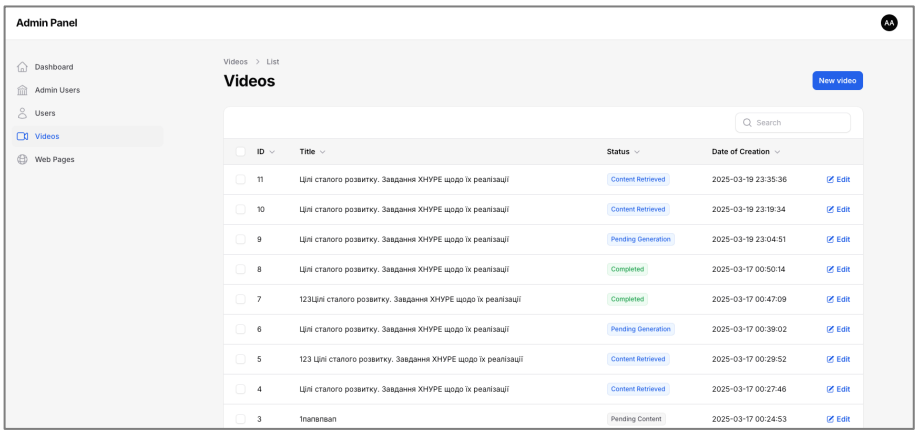


Рисунок 4.32 – Сторінка панелі адміністратора
для перегляду списку відео у системі

На рисунку 4.33 зображено сторінку перегляду списку користувачів у системі. При кліку на кожного користувача відкривається нова сторінка з більш детальною інформацією.

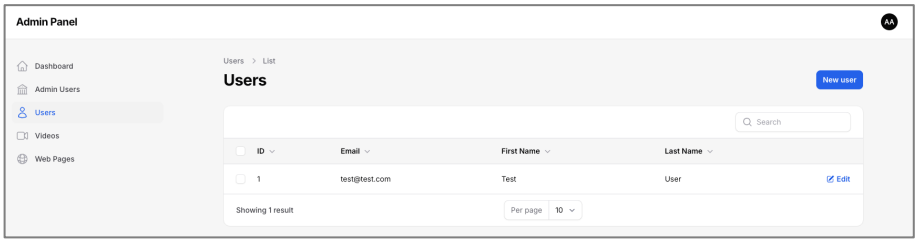


Рисунок 4.33 – Сторінка перегляду списку користувачів

Таким чином, реалізована серверна частина веб-додатку на базі Laravel покриває усі задачі, які ставилися в технічному завданні. Крім того, розроблена архітектура є гнучкою, що дає можливість в майбутньому легко і без зайвих проблем додавати новий функціонал.

4.11.2 Реалізація клієнтської частини на React

Клієнтська частина веб-додатку була реалізована за допомогою сучасної бібліотеки React, яка забезпечує високу продуктивність, компонентний підхід до побудови інтерфейсу та зручність розширення функціональності у майбутньому.

Проєкт має чітко структуровану файлову ієрархію, що спрощує підтримку й масштабування додатку. Дана структура зображена на рисунку 4.34.

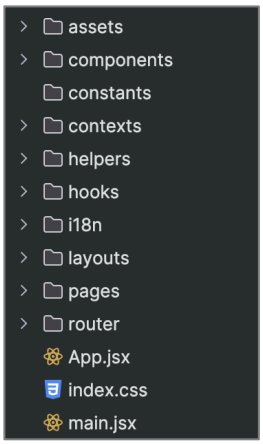


Рисунок 4.34 – Структура папок клієнтської частини веб-додатку

Основні папки та файли мають наступне призначення:

– assets – містить статичні ресурси, такі як зображення та стилі;

– components – набір багаторазових компонентів інтерфейсу;

– constants – файли з константами, що використовуються по всьому додатку;

– contexts – контексти для глобального управління станом;

– helpers – допоміжні функції;

– hooks – користувацькі React-хуки для повторного використання логіки;

– i18n – файли локалізації для підтримки багатомовності;

– layouts – компоненти макетів сторінок (AuthLayout, GuestLayout, Layout), що забезпечують однаковий вигляд для певних груп сторінок;

– pages – основні сторінки додатку;

– router – файли маршрутизації (GuestRoute, ProtectedRoute), що відповідають за доступність сторінок залежно від статусу автентифікації користувача;

– App.jsx – головний компонент додатку;

– index.css – глобальні стилі;

– main.jsx – точка входу у додаток.

Для управління маршрутизацією було реалізовано захищені маршрути (ProtectedRoute) для авторизованих користувачів і гостьові маршрути (GuestRoute) для незареєстрованих користувачів. Це дозволяє забезпечити контроль доступу до сторінок додатку.

Запити до серверної частини виконуються з використанням бібліотеки axios, що забезпечує простий і гнучкий спосіб роботи з HTTP-запитами. Для підтримки багатомовності застосовано i18next, що дозволяє легко додавати нові мови інтерфейсу та керувати перекладами.

Верстка реалізована з використанням Tailwind CSS – сучасної утилітарної CSS-бібліотеки, яка дозволяє швидко створювати адаптивні та стильні інтерфейси без написання великої кількості власних CSS-правил [69].

Крім того, було реалізовано інтуїтивно зрозумілий і привабливий користувацький інтерфейс, що дозволяє користувачам легко працювати з додатком: переглядати та створювати нові відео, завантажувати статті для обробки, керувати профілем тощо.

На рисунку 4.35 зображено першу сторінку веб-додатку – сторінку авторизації користувача, яка забезпечує можливість входу до системи для зареєстрованих користувачів. Інтерфейс сторінки мінімалістичний та

інтуїтивно зрозумілий, містить поля для введення електронної пошти та пароля.

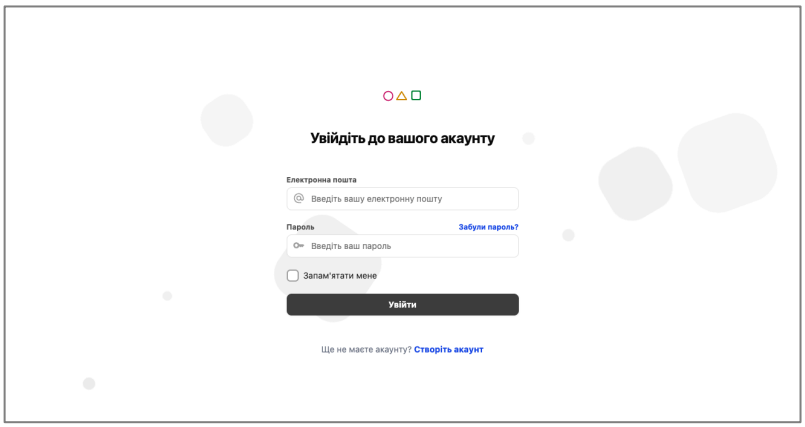


Рисунок 4.35 – Сторінка авторизації користувача

Для нових користувачів доступна сторінка реєстрації, яка зображена на рисунку 4.36, де можна створити обліковий запис, заповнивши основні поля. Валідація введених даних виконується як на клієнтській стороні, так і на серверній, для покращення користувацького досвіду.

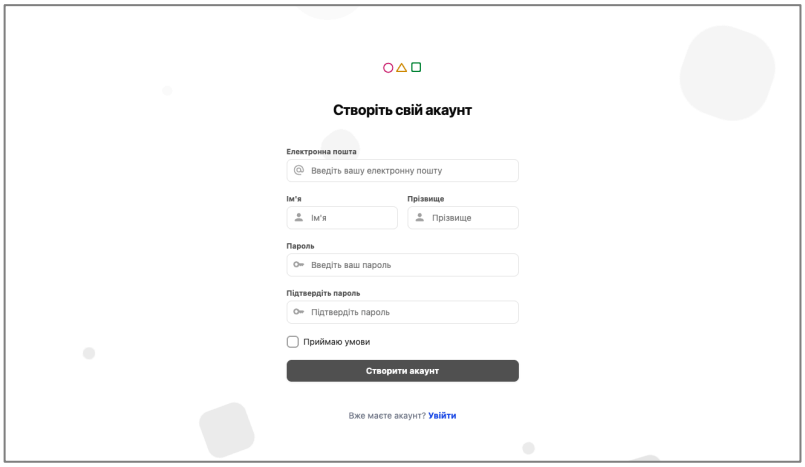


Рисунок 4.36 – Сторінка реєстрації користувача

Після входу користувач потрапляє на сторінку створення відео, яка є відправною точкою роботи із системою. Тут потрібно вказати посилання

на веб-сторінку, яка буде використана для подальшого аналізу та створення відео. Дану сторінку зображено на рисунку 4.37.

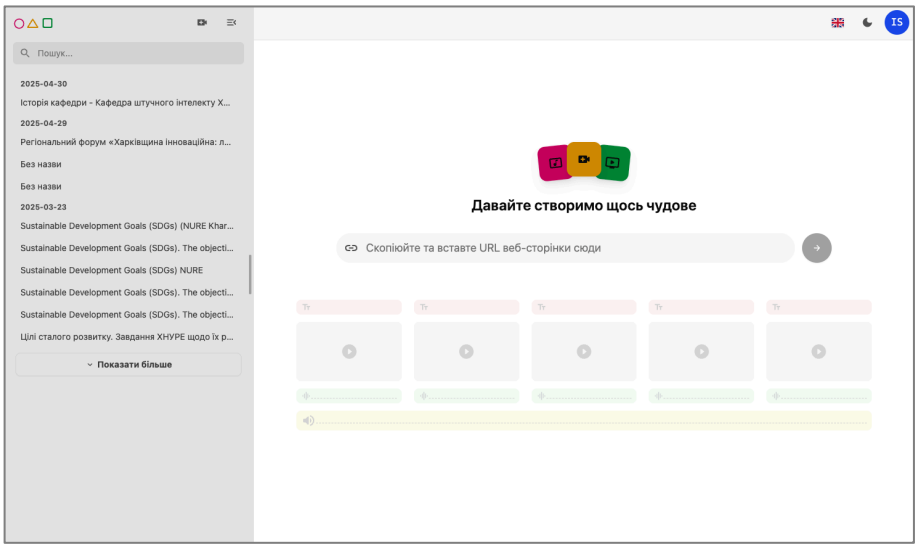


Рисунок 4.37 – Сторінка створення відео

Після запуску процесу обробки контенту користувач потрапляє на сторінку очікування парсингу та аналізу контенту, яка зображена на рисунку 4.38, де відображається статус виконання парсингу веб-сторінки.

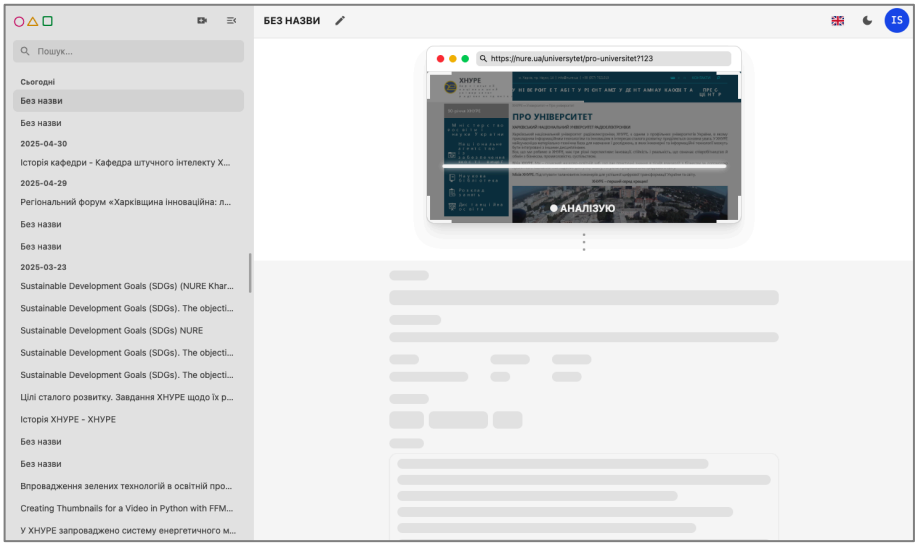


Рисунок 4.38 – Сторінка очікування парсингу
та аналізу контенту веб-сторінки

На рисунку 4.39 зображено сторінку перегляду отриманих результатів, яка відкривається після завершення обробки. На цій сторінці користувач бачить заголовок, короткий зміст, категорію, сентимент тексту, ключові слова та короткий зміст.

Рисунок 4.39 – Перегляд отриманих результатів парсингу та аналізу контенту

На рисунку 4.40 зображено налаштування сцен та відео, де користувач може переглядати автоматично згенеровані сцени, редагувати текст для кожної сцени або видаляти непотрібні.

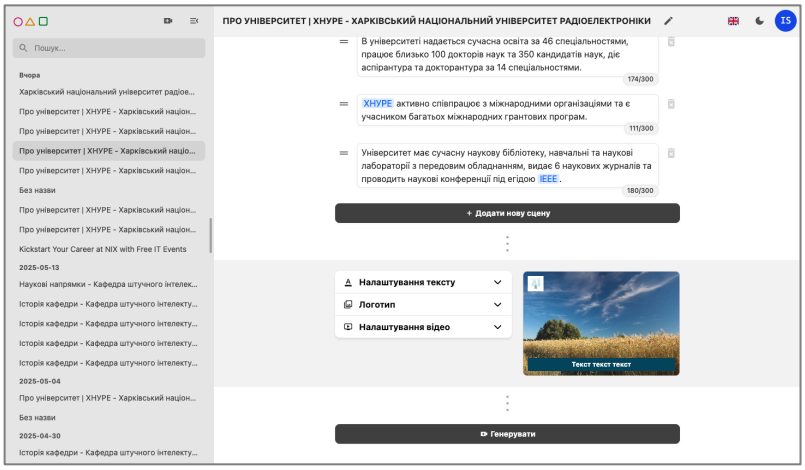


Рисунок 4.40 – Налаштування сцен та відео

У користувача також є можливість налаштування відео, де можна обрати орієнтацію (ландшафтна чи портретна), стилі оформлення, кольори, якість фінального відео та інші параметри. На рисунку 4.41 зображено всі налаштування для створення відео.

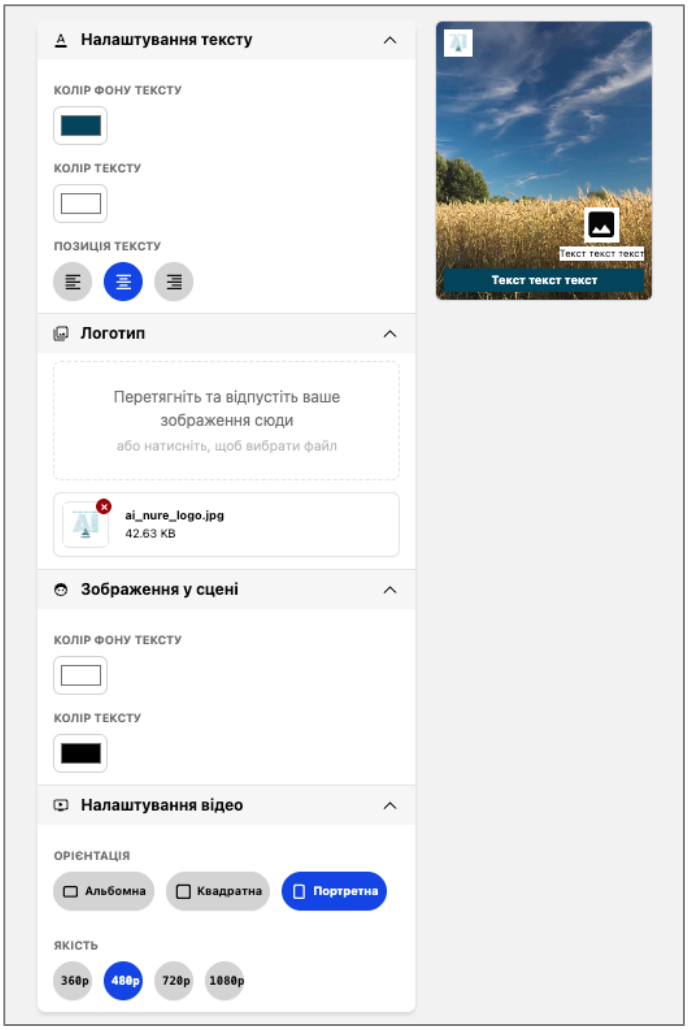


Рисунок 4.41 – Детальні налаштування відео

Для сцен, у яких було виявлено іменовані сутності, доступна можливість завантаження зображень. Це дозволяє прикріпити фотографії або інші зображення для візуального супроводу відповідних сцен. Для кожної сцени є ліміт в одне зображення для однієї іменованої сутності. На рисунку 4.42 зображено приклад виявлення іменованих сутностей в тексті сцени і завантаження відповідного зображення для одної з них.

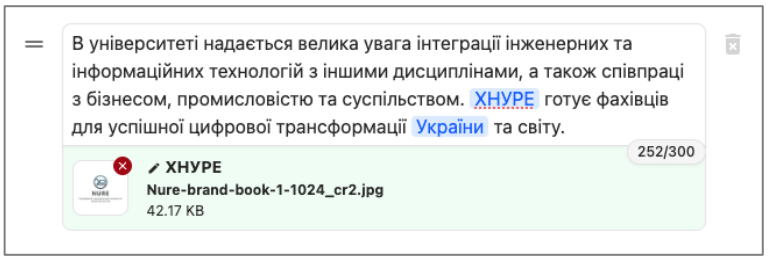


Рисунок 4.42 – Завантаження зображення
для виявлених іменованих сутностей сцени

Після завершення процесу генерації відео користувач отримує доступ до перегляду результату створеного відео, де можна подивитися фінальний відеоролик або завантажити його. На рисунку 4.43 зображено плеєр зі створеним відео.

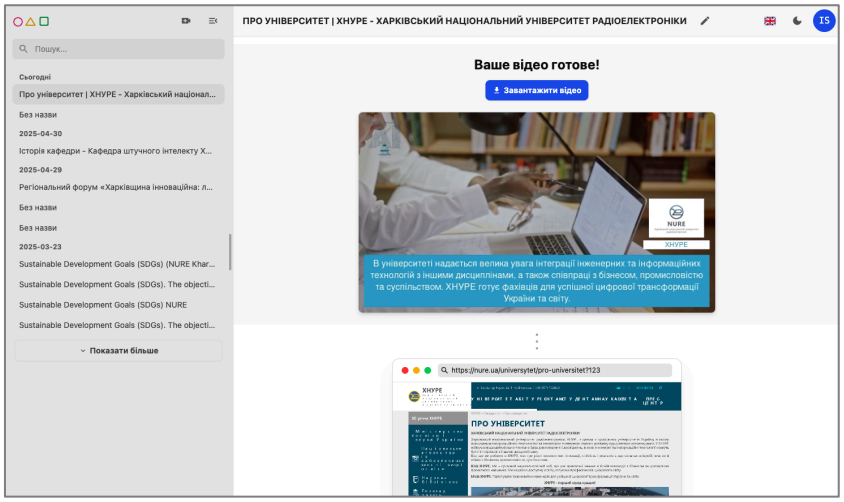


Рисунок 4.43 – Перегляд результату створеного відео

На рисунку 4.44 зображено сторінку налаштування користувача, де можна змінити особисті дані та налаштування облікового запису. Після оновлення паролю усі інші сесії користувача автоматично деактивуються, що підвищує безпеку та захищає від несанкціонованого доступу. Також у користувача є можливість видалити свій акаунт з усіма даними та створеними відео.

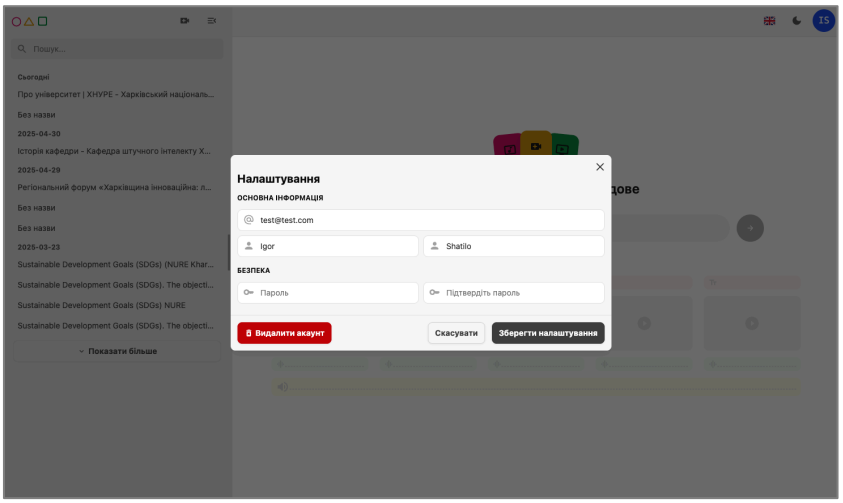


Рисунок 4.44 – Налаштування користувача

Для зручності користувачів реалізовано підтримку темної теми оформлення, яка активується за бажанням і покращує комфорт роботи в умовах низької освітленості. На рисунку 4.45 зображено приклад сторінки перегляду результатів парсингу і аналізу веб-сторінки у темній темі.

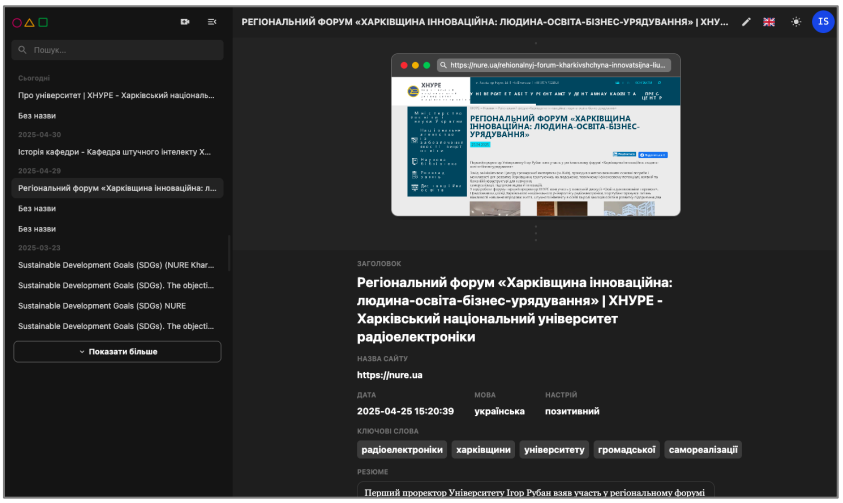


Рисунок 4.45 – Темна тема веб-додатку

Розроблена клієнтська частина веб-додатку повністю відповідає поставленим функціональним і експлуатаційним вимогам. Інтерфейс забезпечує інтуїтивну взаємодію користувача із системою та дозволяє

ефективно управляти процесом створення відео. Використання сучасних технологій фронтенду сприяє стабільності, швидкодії та розширюваності клієнтської частини додатку.

4.12 Перспективи розвитку

У процесі розробки системи було реалізовано повний цикл автоматизації створення відео на основі текстового контенту. Проте, з метою подальшого вдосконалення функціоналу та покращення користувацького досвіду, передбачено низку майбутніх доопрацювань і розширень.

Насамперед, планується реалізація можливості ручного вибору відео для кожної сцени. Хоча наразі система автоматично підбирає мультимедійний контент за допомогою KeyBERT та Pexels API, надання користувачеві можливості самостійно обирати відеофрагменти з бібліотеки стокових відео чи із запропонованого списку дозволить значно підвищити релевантність і креативність створених відеоматеріалів. Така функція також стане в пригоді для тих сценаріїв, де автоматичний підбір не враховує специфіку певного контенту чи індивідуальні побажання користувачів.

Ще одним важливим напрямом розвитку є впровадження генерації відео за допомогою технологій штучного інтелекту (GenAI). Використання сучасних генеративних моделей відео дозволить не лише обирати готові відеофрагменти, а й створювати унікальні відео на основі ключових слів, категорій та заданої тематики сцени. Це розширить творчі можливості системи та відкриє нові перспективи персоналізації контенту.

У сфері обробки тексту можлива інтеграція більш гнучких моделей класифікації та семантичного аналізу, які дозволять краще адаптувати систему до різних мов та тематики вхідного контенту. Для покращення

якості озвучення передбачено підтримку мультимовних моделей синтезу мовлення з можливістю вибору стилю та емоційного тону голосу.

Крім того, заплановано вдосконалення користувацького інтерфейсу веб-додатку, зокрема – розширення налаштувань відео та оптимізація процесів генерації для підвищення швидкості й стабільності роботи системи. Також розглядається можливість додати модуль аналітики, який дозволить користувачам відстежувати популярність і ефективність створених відео, які можна розмістити на різних веб-сайтах.

Таким чином, подальший розвиток системи спрямовано на підвищення її гнучкості, розширення функціональних можливостей і забезпечення максимальної адаптивності до потреб кінцевих користувачів.

4.13 Висновки до розділу

У четвертому розділі було детально описано процес програмної реалізації розробленої системи автоматизованого створення відео на основі текстового контенту веб-сайтів. Реалізовано повний цикл обробки даних: від парсингу веб-сторінок та аналізу отриманого контенту до формування фінального відеофайлу. Для обробки тексту було використано сучасні моделі машинного навчання та глибокого навчання, що дозволило досягти високої точності категоризації, визначення сентименту, виділення іменованих сутностей та створення короткого змісту.

У рамках роботи проведено тренування та тестування власної нейромережі для класифікації тексту, що дало змогу адаптувати систему до специфіки реальних даних. Реалізовано екстрактивне та абстрактивне реферування, що забезпечило скорочення вхідного тексту без втрати основного змісту. Для підбору мультимедійного контенту було застосовано модель KeyBERT та API стокових відео, що дозволило автоматично знаходити релевантний візуальний супровід. Крім того,

впроваджено синтез мовлення для озвучення тексту та генерацію музичного супроводу з використанням сучасних генеративних моделей.

Веб-додаток, розроблений на основі архітектури клієнт-сервер, об’єднує всі функціональні модулі системи та забезпечує зручний інтерфейс для користувачів і адміністраторів. Реалізована мікросервісна архітектура інтелектуальної частини дозволила розподілити обчислювальні задачі та забезпечити масштабованість рішення. Проведені експериментальні дослідження підтвердили ефективність реалізованих підходів та високу якість створюваного відеоконтенту.

Таким чином, результати програмної реалізації демонструють успішне втілення поставлених задач і підтверджують доцільність використання сучасних методів штучного інтелекту для автоматизації створення відео з текстової інформації.

ВИСНОВКИ

У рамках даної кваліфікаційної роботи було розроблено та реалізовано прототип системи для автоматизації створення відео на основі короткого змісту тексту з веб-сайтів. Вона охоплює повний цикл обробки: від отримання текстових матеріалів шляхом парсингу веб-сайтів, їхньої обробки та інтелектуального аналізу до формування готового відеофайлу з озвученням, візуальним супроводом і музичним фоном. Застосування сучасних методів штучного інтелекту, моделей природної мовної обробки, генеративних підходів та алгоритмів мультимедійної компоновки дозволило досягти високого рівня автоматизації та адаптивності системи.

У процесі роботи було виконано глибоке вивчення предметної галузі, проведено аналіз сучасних підходів до вилучення та обробки контенту з веб-сторінок, а також досліджено ефективні методи реферування, класифікації, семантичного аналізу, виділення ключових слів та оцінки емоційного тону тексту. Значну увагу було приділено вивченню принципів підбору візуального та аудіоконтенту відповідно до тематики текстового джерела.

Результатом дослідження стала побудова єдиної інтегрованої системи, яка поєднує всі етапи обробки тексту й формування відео. Запропоноване рішення є універсальним та здатне працювати з широким спектром вхідної інформації, включно з двома мовами (українська та англійська), стилями написання та форматами веб-контенту. Гнучкість архітектури системи дозволяє адаптувати її під різні сценарії використання – від освітніх та інформаційних до маркетингових і розважальних.

Практична цінність реалізованої розробки полягає у її здатності значно скоротити часові та людські ресурси, які зазвичай витрачаються на створення відеоконтенту. Такий підхід дозволяє перетворити текстову інформацію у зручний для сприйняття формат, що, у свою чергу, підвищує

доступність, привабливість і цінність контенту для кінцевого користувача. Система може бути ефективно використана в таких галузях, як онлайн-ЗМІ, освітні платформи, соціальні мережі, корпоративні комунікації, а також у будь-яких інших сферах, де потрібна швидка візуалізація текстових даних.

Перспективи розвитку створеної системи передбачають подальше вдосконалення алгоритмів генерації контенту, розширення підтримки більшої кількості мов, підвищення якості відео та звуку, а також інтеграцію з зовнішніми сервісами для публікації та поширення згенерованих матеріалів. Крім того, можлива реалізація механізмів персоналізації відео відповідно до профілю або інтересів користувача.

Таким чином, результати виконаної роботи підтверджують ефективність використання інтелектуальних підходів до автоматизації створення відеоконтенту та демонструють реальні можливості їх застосування в сучасному цифровому середовищі. Запропоноване рішення є актуальним, практично значущим і може стати основою для подальших наукових досліджень і прикладних розробок у сфері автоматизованих мультимедійних систем.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Burchill A. Video marketing statistics for your 2025 campaigns. Dash, Digital Asset Management (DAM) for Growing Brands. URL: https://www.dash.app/blog/video-marketing-statistics (дата звернення: 21.04.2025).

2. Victor Blasco. 11 Video Marketing Statistics of 2025: Numbers Tell the Story. Outbrain. URL: https://www.outbrain.com/blog/video-marketing-statistics/ (дата звернення: 21.04.2025).

3. Michele Majidi. Topic: Digital video advertising and marketing worldwide. Statista. URL: https://www.statista.com/topics/5960/digital-video-advertising (дата звернення: 21.04.2025).

4. Saket Dandotia. The Rise Of Generative AI In Video Production: Opportunities And Challenges. BW Disrupt. URL: https://www.bwdisrupt.com/article/the-rise-of-generative-ai-in-video-production-opportunities-and-challenges-485847 (дата звернення: 21.04.2025).

5. Travel B. The 25 best places to travel in 2025. BBC Home – Breaking News, World News, US News, Sports, Business, Innovation, Climate, Culture, Travel, Video & Audio. URL: https://www.bbc.com/travel/article/20250115-the-25-best-places-to-travel-in-2025 (дата звернення: 21.04.2025).

6. Головна сторінка. Інтернет-магазин ROZETKA: офіційний сайт онлайн-гіпермаркету Розетка в Україні. URL: https://rozetka.com.ua/ (дата звернення: 21.04.2025).

7. Про університет | ХНУРЕ – Харківський національний університет радіоелектроніки. NURE. URL: https://nure.ua/universytet/pro-universitet (дата звернення: 21.04.2025).

8. Rebelo M. The 11 best AI video generators in 2025 | Zapier. Automate without limits | Zapier. URL: https://zapier.com/blog/best-ai-video-generator/ (дата звернення: 21.04.2025).

9. Pictory – Easy Video Creation For Content Marketers. Pictory.ai. URL: https://pictory.ai/ (дата звернення: 21.04.2025).

10. Invideo AI – Bring your Ideas to life. InVideo – Online Video Creator for Content and Marketing Videos. URL: https://invideo.io/ (дата звернення: 21.04.2025).

11. Free AI Video Generator – Create AI Videos in 140 Languages. Synthesia. URL: https://www.synthesia.io/ (дата звернення: 21.04.2025).

12. Knowledge Graph, AI Web Data Extraction and Crawling. Diffbot. URL: https://www.diffbot.com/ (дата звернення: 21.04.2025).

13. Nehal. Guide to Web Scraping | Tools and Techniques. PromptCloud. URL: https://www.promptcloud.com/blog/the-ultimate-guide-to-web-scraping-tools-techniques-and-use-cases/ (дата звернення: 22.04.2025).

14. Antonello Zanini. Web Scraping With Playwright and Node.JS in 2025. Bright Data. URL: https://brightdata.com/blog/how-tos/playwright-web-scraping (дата звернення: 22.04.2025).

15. mozilla/readability: A standalone version of the readability lib. GitHub. URL: https://github.com/mozilla/readability (дата звернення: 22.04.2025).

16. Web article scraping, analysis & processing. Newspaper4k. URL: https://newspaper4k.readthedocs.io/en/latest/ (дата звернення: 22.04.2025).

17. kohlschutter/boilerpipe: Work in progress transmit from Google Code. GitHub. URL: https://github.com/kohlschutter/boilerpipe (дата звернення: 22.04.2025).

18. Nishtha. How to do Web Scraping with LLMs for Your Next AI Project? ProjectPro. URL: https://www.projectpro.io/article/web-scraping-with-llms/1081 (дата звернення: 22.04.2025).

19. Grancharov S. Text Language Detection with Python. Medium. URL: https://medium.com/@monigrancharov/text-language-detection-with-python-beb49d9667b3 (дата звернення: 23.04.2025).

20. What is neural machine translation, and how does it work? Language Translation and Content Localization Solutions | Smartling. URL: https://www.smartling.com/blog/neural-machine-translation (дата звернення: 23.04.2025).

21. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I. Attention is all you need. arXiv preprint arXiv:1706.03762, 2017. URL: https://doi.org/10.48550/arXiv.1706.03762 (дата звернення: 23.04.2025).

22. Documentation. Marian NMT. URL: https://marian-nmt.github.io/ (дата звернення: 23.04.2025).

23. Keyword Extraction Methods in NLP – GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/keyword-extraction-methods-in-nlp/ (дата звернення: 23.04.2025).

24. Rapid Keyword Extraction (RAKE) Algorithm in Natural Language Processing. Analytics Vidhya. URL: https://www.analyticsvidhya.com/blog/2021/10/rapid-keyword-extraction-rake-algorithm-in-natural-language-processing/ (дата звернення: 23.04.2025).

25. Mudadla S. What is Parts of Speech (POS) Tagging Natural Language Processing? In what kind of applications we can use Parts of Speech (POS) Tagging in Natural Language Processing. Medium. URL: https://medium.com/@sujathamudadla1213/what-is-parts-of-speech-pos-tagging-natural-language-processing-in-2b8f4b07b186 (дата звернення: 24.04.2025).

26. What is BERT? NVIDIA Data Science Glossary. URL: https://www.nvidia.com/en-us/glossary/bert/ (дата звернення: 24.04.2025).

27. Schopf T. Keyphrase Extraction with BERT Transformers and Noun Phrases. Medium. URL: https://medium.com/data-science/enhancing-keybert-keyword-extraction-results-with-keyphrasevectorizers-3796fa93f4db (дата звернення: 24.04.2025).

28. Patricia Orza. Text Classifiers in Machine Learning: A Practical Guide. Levity | Streamline Your Freight Email Operations with AI and automation. URL: https://levity.ai/blog/text-classifiers-in-machine-learning-a-practical-guide (дата звернення: 24.04.2025).

29. Asrawi H., Utami E., Yaqin A. LSTM and Bidirectional GRU Comparison for Text Classification. sinkron. 2023. Т. 8, № 4. С. 2264–2274. URL: https://doi.org/10.33395/sinkron.v8i4.12899 (дата звернення: 24.04.2025).

30. What is text summarization? IBM – United States. URL: https://www.ibm.com/think/topics/text-summarization (дата звернення: 24.04.2025).

31. Payong A. BART Model for Text Summarization. DigitalOcean | Cloud Infrastructure for Developers. URL: https://www.digitalocean.com/community/tutorials/bart-model-for-text-summarization-part1 (дата звернення: 25.04.2025).

32. Sovit Rath. Text Summarization using T5. LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow with code, & tutorials. URL: https://learnopencv.com/text-summarization-using-t5/ (дата звернення: 25.04.2025).

33. Hashemi-Pour C., Barney N. What Is Named Entity Recognition (NER)? Definition from TechTarget. TechTarget. URL: https://www.techtarget.com/whatis/definition/named-entity-recognition-NER (дата звернення: 25.04.2025).

34. Monica Munnangi. A Brief History of Named Entity Recognition. arXiv.org. URL: https://arxiv.org/abs/2411.05057 (дата звернення: 26.04.2025).

35. Morrison R., Caswell A. I’ve spent 200 hours testing the best AI video generators – here’s my top picks. Tom’s Guide. URL: https://www.tomsguide.com/features/5-best-ai-video-generators-tested-and-compared (дата звернення: 26.04.2025).

36. Free Stock Photos, Royalty Free Stock Images & Copyright Free Pictures. Pexels. URL: https://www.pexels.com/ (дата звернення: 26.04.2025).

37. Royalty Free Background Music Downloads. Fesliyan Studios. URL: https://www.fesliyanstudios.com/ (дата звернення: 26.04.2025).

38. Darshan Deshpande. Royalty-Free Audio Dataset. Kaggle. URL: https://www.kaggle.com/datasets/darshan1504/royaltyfree-audio-dataset (дата звернення: 26.04.2025).

39. facebook/musicgen-large. Hugging Face – The AI community building the future. URL: https://huggingface.co/facebook/musicgen-large (дата звернення: 27.04.2025).

40. Sankar S. MusicGen from Meta AI – Model Architecture, Vector Quantization and Model Conditining explained. AI Bites. URL: https://www.ai-bites.net/musicgen-from-meta-ai-model-architecture-vector-quantization-and-model-conditining-explained/ (дата звернення: 27.04.2025).

41. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. AI at Meta. URL: https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/ (дата звернення: 27.04.2025).

42. Tyler Weitzman. A short history of text to speech. Speechify. URL: https://speechify.com/blog/history-of-text-to-speech/ (дата звернення: 27.04.2025).

43. Jean-Rémi Larcelet-Prost. Neural Text to Speech (TTS): Making Voice Experiences More Human. ReadSpeaker. URL: https://www.readspeaker.com/blog/neural-text-to-speech/ (дата звернення: 27.04.2025).

44. Text-to-Speech AI: Lifelike Speech Synthesis. Google Cloud. URL: https://cloud.google.com/text-to-speech (дата звернення: 27.04.2025).

45. StyleTTS 2: Towards Human-Level Text-to-Speech through Style Diffusion and Adversarial Training with Large Speech Language Models / Yinghao Aaron Li та ін. arXiv.org. URL: https://doi.org/10.48550/arXiv.2306.07691 (дата звернення: 28.04.2025).

46. Kunal Kejriwal. StyleTTS 2: Human-Level Text-to-Speech with Large Speech Language Models. Unite.AI. URL: https://www.unite.ai/styletts-2-human-level-text-to-speech-with-large-speech-language-models/ (дата звернення: 28.04.2025).

47. Automate Video Editing with MoviePy in Python. Best AI Tools Directory & AI Tools List – Toolify. URL: https://www.toolify.ai/ai-news/automate-video-editing-with-moviepy-in-python-1200309 (дата звернення: 28.04.2025).

48. What is LEMP Stack? GeeksforGeeks. URL: https://www.geeksforgeeks.org/what-is-lemp-stack/ (дата звернення: 29.04.2025).

49. What is Laravel? Explain it like I'm five. RunCloud Blog. URL: https://runcloud.io/blog/what-is-laravel (дата звернення: 29.04.2025).

50. Brian Dainis. How to Build a Basic Single-Page Application (SPA) with React: A Step-by-Step Guide. Curotec. URL: https://www.curotec.com/insights/how-to-build-a-react-spa/ (дата звернення: 29.04.2025).

51. Build Your Own Microservices in Flask. GeeksforGeeks. URL: https://www.geeksforgeeks.org/build-your-own-microservices-in-flask/ (дата звернення: 29.04.2025).

52. Introduction to Celery – Celery 5.5.2 documentation. Celery – Distributed Task Queue – Celery 5.5.2 documentation. URL: https://docs.celeryq.dev/en/stable/getting-started/introduction.html (дата звернення: 30.04.2025).

53. Шатило І.Ю, Чала Л.Е. Використання технологій штучного інтелекту для створення відео на основі тексту. Радіоелектроніка та молодь у XXI столітті : матеріали 29-го Міжнар. молодіж. форуму, м. Харків, 16-19 квіт. 2025 р. Харків, 2025. Т. 6. С. 87–89 (дата звернення: 30.04.2025).

54. MinIO | Enterprise Grade, High Performance Object Storage. MinIO. URL: https://min.io/product/overview (дата звернення: 30.04.2025).

55. Mitchell J. Trending YouTube Video Statistics Dataset. Kaggle. URL: https://www.kaggle.com/datasets/datasnaek/youtube-new (дата звернення: 02.05.2025).

56. Rishav Sharma. YouTube Trending Video Dataset (updated daily). Kaggle. URL: https://www.kaggle.com/datasets/rsrishav/youtube-trending-video-dataset (дата звернення: 02.05.2025).

57. siebert/sentiment-roberta-large-english. Hugging Face – The AI community building the future. URL: https://huggingface.co/siebert/sentiment-roberta-large-english (дата звернення: 03.05.2025).

58. Maas A. Large Movie Review Dataset. Stanford Artificial Intelligence Laboratory. URL: https://ai.stanford.edu/~amaas/data/sentiment/ (дата звернення: 03.05.2025).

59. facebook/bart-large-cnn. Hugging Face – The AI community building the future. URL: https://huggingface.co/facebook/bart-large-cnn (дата звернення: 03.05.2025).

60. CNN Dailymail Dataset. Hugging Face – The AI community building the future. URL: https://huggingface.co/datasets/abisee/cnn_dailymail (дата звернення: 04.05.2025).

61. EvanD/xlm-roberta-base-ukrainian-ner-ukrner. Hugging Face – The AI community building the future. URL: https://huggingface.co/EvanD/xlm-roberta-base-ukrainian-ner-ukrner (дата звернення: 05.05.2025).

62. Davlan/xlm-roberta-base-ner-hrl. Hugging Face – The AI community building the future. URL: https://huggingface.co/Davlan/xlm-roberta-base-ner-hrl (дата звернення: 05.05.2025).

63. patriotyk/styletts2_ukrainian_single. Hugging Face – The AI community building the future. URL: https://huggingface.co/patriotyk/styletts2_ukrainian_single (дата звернення: 07.05.2025).

64. hexgrad/Kokoro-82M. Hugging Face – The AI community building the future. URL: https://huggingface.co/hexgrad/Kokoro-82M (дата звернення: 07.05.2025).

65. GitHub – jzhang38/TinyLlama. GitHub. URL: https://github.com/jzhang38/TinyLlama (дата звернення: 08.05.2025).

66. Laravel Sanctum. Laravel – The PHP Framework For Web Artisans. URL: https://laravel.com/docs/12.x/sanctum (дата звернення: 09.05.2025).

67. Introduction | laravel-data. Websites & web applications in Laravel | Spatie. URL: https://spatie.be/docs/laravel-data/v4/introduction (дата звернення: 09.05.2025).

68. Goetz D. Why We Love Filament: A Powerful Tool for Laravel Developers. Kirschbaum. URL: https://kirschbaumdevelopment.com/insights/why-we-love-filament (дата звернення: 09.05.2025).

69. Introduction to Tailwind CSS – GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/introduction-to-tailwind-css/ (дата звернення: 09.05.2025).