Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Дослідження продуктивності нейромережевих моделей при семантичному аналізі тексту

(тема)

Виконав:

студент II курсу, групи СПм-20-2

Литвиненко В.С.
(прізвище, ініціали)

Спеціальність

123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

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

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: проф. Фесенко Т.Г.
(посада, прізвище, ініціали)

Допускається до захисту

В.о. зав. кафедри ЕОМ _____ Волк М. О.
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

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

Освітня програма Комп'ютерні системи та мережі
(повна назва)

ЗАТВЕРДЖУЮ:
В.о. зав.
кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Литвиненку Владиславу Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження продуктивності нейромережевих моделей при семантичному аналізі тексту

затверджена наказом по університету від " 24 " березня 2022 р. № 413 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 18 травня 2022р.

3. Вхідні дані до роботи

Тип обладнання – відеокарта NVIDIA GeForce GTX 1650 Mobile, процесор Intel Core i7-9750H, аудіофайли формату wav, мова програмування – Python

4. Перелік питань, що потрібно опрацювати у роботі

Розробка системи для розпізнавання мовлення та сумаризації тексту

Дослідження методів роботи із розпізнавання мовлення

Дослідження методів роботи сумаризації тексту

Оцінка таймінгу роботи етапу обробки розпізнавання мовлення

Оцінка таймінгу роботи етапу сумаризації тексту

Оцінка якості роботи запропонованої системи

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 16 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Дослідження загальних параметрів систем	30.03.22-05.04.22	
	розпізнавання сканованих документів		
2	Дослідження методів роботи із розпізнавання	06.04.22-16.04.22	
	мовлення		
3	Дослідження методів роботи сумаризації	17.04.22-22.04.22	
4	Оцінка таймінгу роботи етапу обробки	23.04.21-28.04.22	
	розпізнавання мовлення		
5	Оцінка таймінгу роботи етапу сумаризації	29.04.22-30.04.22	
6	Оцінка таймінгу роботи розробленої системи	01.05.22-03.05.22	
7	Оформлення пояснювальної записки	04.05.22-13.05.22	
8	Подання кваліфікаційної роботи на	14.05.22-18.05.22	
	рецензування		

Дата видачі завдання 28 березня 2022 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Фесенко Т. Г.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 77 с., 17 рис., 10 табл., 1 дод., 21 джерел.

NLP, ПЕРЕТВОРЕННЯ АУДІОФАЙЛІВ В ТЕКСТ, СУМАРИЗАЦІЯ ТЕКСТУ, АНАЛІЗ ТЕКСТУ, XL-SUM, wav2vec, mT5.

Метою кваліфікаційної роботи є створення гібридної моделі, яка надає можливість розпізнання мовлення, перетворення наявних даних в текст і останнім етапом проведення сумаризації даного тексту, що містить лекційний матеріал, в текстовий вигляд, зберігаючи лише важливу змістовну частину лекції. А також порівняти ефективність нейромережевих моделей використаних в розробці запропонованої системи на різних GPGPU.

У ході виконання кваліфікаційної роботи було проведено дослідження параметрів, які повинна задовольняти система розпізнавання голосу та система сумаризації тексту. Також були розглянуті існуючі методи обробки NLP у випадку з перетворенням мовлення в текст і його сумаризації.

На підставі проведеного дослідження був запропонований та розроблений комплекс обробки вхідних аудіозаписів та їх сумаризації. В процесі розробки системи було проведено порівняння розпізнавання мовлення англійською і українською мовами за допомогою використання моделі, яка розроблена з використанням згорткової нейроної мережі. А також проведено сумаризації текстів на вище вказаних мовах, з використанням новітньої моделі від компанії Google.

ABSTRACT

Master’s thesis: 77 pages, 17 figures, 10 tables, 1 appendices, 21 sources.

NLP, CONVERTING AUDIO FILES INTO TEXT, TEXT SUMMARIZATION, TEXT ANALYSIS, XL-SUM, wav2vec, mT5.

The purpose of the qualification work is to create a hybrid model that provides speech recognition, conversion of existing data into text and the last stage of summarization of the text containing lecture material in text form, retaining only an important part of the lecture. And also to compare efficiency of the neural network models used in development of the offered system on various GPGPU.

In the course of the qualification work, a study of the parameters to be met by the voice recognition system and the text summarization system was conducted. Existing methods of NLP processing in the case of speech-to-text conversion and summarization were also considered.

On the basis of the conducted research the complex of processing of input audio recordings and their summarization was offered and developed. During the development of the system, a comparison of speech recognition in English and Ukrainian was performed using a model developed using a convolutional neural network. We also summarized the texts in the above languages, using the latest model from Google.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

NLP – Обробка природної мови

LSTM – довга короткочасна пам’ять (Long short-term memory)

ASR – автоматичне розпізнавання мови

LSA – Латентно-семантичний аналіз

SVD – Сингулярний розклад матриці

GMM – Узагальнений метод моментів

WER – Рівень помилок слів

MMSE – Мінімальна середньоквадратична помилка

ВСТУП

Мовлення є найважливішою частиною спілкування між людьми. Хоча існують різні засоби для вираження наших думок і почуттів, мова вважається основним засобом спілкування. Розпізнавання мовлення – це процес, коли машина розпізнає мову різних людей на основі певних слів або фраз. Варіації у вимові досить очевидні в мовленні кожної людини. Початкова форма промови – це сигнал, і сигнал обробляється таким чином, що вся інформація, наявна в сигналі, перетворюється в текстовий формат. Вилучення ознак – це процес отримання сигналу та перетворення його в необхідний формат з певною логікою. Незважаючи на те, що мовлення є найпростішим способом спілкування, існують деякі проблеми з розпізнаванням мовлення, як от плавність мовлення, вимова, порушення слів, проблеми з заїканням тощо. Усе це потрібно вирішувати під час обробки мовлення. Реферування тексту є одним з основних понять, що використовуються в області документації. Довгі документи важко читати та розуміти, оскільки вони забирають багато часу. Сумаризація тексту вирішує цю проблему, надаючи його скорочене резюме із семантикою. У пропонованій роботі реалізовано поєднання перетворення мови в текст та узагальнення тексту. Цей гібридний метод допоможе програмам, які вимагають короткого резюме довгих виступів, що дуже корисно для документації.

Особливо дана проблема стала актуальною під час пандемії або нестабільної ситуації в країнах. Основний акцент даної роботи є направленим на забезпечення здатності отримавати якісну освіту, а також порівняння різних алгоритмів для більш ефективної обробки даних.

Інформація подана у текстовому вигляді є цінним джерелом знань, однак, часто її необхідно ефективно обробляти, щоб почерпнути з неї якомога більше користі. З кожним роком все більш актуальним стає створення анотації (резюме, реферату). Для цього необхідно стиснути

фрагменти тексту до більш короткого варіанту, зменшити розмір початкового тексту при одночасному збереженні ключових інформаційних елементів і змісту. Оскільки створення анотації вручну є часозатратним і, як правило, трудомістким завданням, питання автоматизації цього процесу набуває все більшої популярності у академічних дослідженнях.

Одним із напрямків досліджень є обробка мовлення та перетворення звукових файлів у текстовий матеріал, при цьому зберігаючи лише важливу та доцільну інформацію. Ключовими проблемами є визначення теми, інтерпретація, генерування анотації та її якісна оцінка. Найважливішими завданнями є визначення ключових фраз та використання їх для вибору речень, які увійдуть до анотованого тексту.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Період пандемії та військового конфлікту значно підвищив актуальність розвитку та розширення функціоналу різних цифрових освітніх платформ. Дані інформаційні простори потрібні на різних рівнях освіти в країнах світу – від початкової школи до закладів вищої освіти, а також навчальних курсів у різних сферах бізнесу, дозволяючи забезпечити учнів методичними матеріалами, спілкуванням з викладачами, а також даючи можливість для віддаленого контролю рівня знань. Таким чином, організовується віртуальна взаємодія розподіленої спільноти користувачів. Кількість інтерактивних можливостей цифрових освітніх платформ постійно розширюється. До базового функціоналу (доступ до навчальних матеріалів на хмарних сервісах зберігання файлів, дистанційна взаємодія з викладачами) додаються нові опції. Наприклад, блоки тестування, контролю знань, управління рівнем доступу до навчального контенту в залежності від ролі користувача, аналізу статистичних даних (відвідуваності) тощо. Важливою складовою інформаційного простору для віддаленої освіти залишаються онлайн-лекції з експертами, які можуть проводитися в режимі питання-відповідь та висвітлювати питання від слухачів курсу. Подальший доступ до матеріалів онлайн-лекцій має бути зручним для розуміння та засвоєння [1]. Запис лекції забезпечує доступ до звукових файлів, які припускаються прослуховування, але не призначені для друкованого відтворення.

Тому, розширення існуючих цифрових освітніх платформ можливістю формування анотації (резюме, реферату) лекції та подання її у вигляді текстографічних матеріалів для подальшого використання слухачами курсу на паперових носіях, є завданням актуальним, оскільки здатне покращувати якість пред'явлення навчальної інформації та умов роботи з нею, а також підвищити оцінку якості дистанційного освітнього ресурсу з погляду змістовно методологічного аспекту.

1.1 Перетворення мовлення в текст

Перетворення мовлення в текст (Speech to text) – це здатність комп’ютерного програмного забезпечення розпізнавати слова та фрази у розмовній мові та перетворювати їх у текст, зрозумілий людині. Розпізнавання мови сягає своїм корінням у дослідженнях, проведених у Bell Labs на початку 1950-х років. Ранні системи були обмежені одним носієм і мали обмежений словниковий запас приблизно з десятка слів. Сучасні системи розпізнавання мовлення пройшли довгий шлях від своїх стародавніх аналогів. Вони можуть розпізнавати мову кількох мовців і мають величезний словниковий запас численними мовами [2].

Це не нова технологія, перші експерименти датуються 1970-ми роками, але безсумнівно, що останні роки розвиток у сфері великих даних та штучного інтелекту дало потужний поштовх до вдосконалення цієї технології, а отже й її надійність. Порівняно з минулим, точність транскрипції фактично покращилася до такого ступеня, що з чітким і чітко визначеним джерелом звуку рівень точності може перевищувати 99%.

Цей момент необхідно виділити, оскільки багато залежить від умов запису. Програмне забезпечення для розпізнавання мовлення все ще не може інтерпретувати мову в шумному оточенні або коли багато людей говорять одночасно. Крім того, якість і точність транскрипції визначають не тільки складність мовлення, а й умови навколишнього середовища.

Ядром системи автоматичної транскрипції є автоматичне розпізнавання мовлення, яке об'єднує акустичні та мовні компоненти. Акустичний компонент відповідає за перетворення аудіофайлів у послідовність дуже маленьких акустичних одиниць. «Аналоговий звук», тобто вібрації, створювані під час розмови, перетворюються в цифрові сигнали, які можна сканувати програмним забезпеченням. Потім акустичні одиниці асоціюються з існуючими «фонемами», тобто звуками, які використовуються в конкретній мові для формування значущих виразів. Тоді лінгвістичний компонент відповідає за перетворення послідовностей акустичних одиниць у слова,

речення та абзаци. Лінгвістичний компонент аналізує всі попередні слова та їх взаємозв'язок, щоб оцінити ймовірність використання того чи іншого слова в продовженні мовлення. Технічно вони називаються «Приховані моделі Маркова» і широко використовуються у всьому програмному забезпеченні розпізнавання мовлення. Обидва компоненти мають бути коректно «навченими» для розуміння певної мови: однаково акустична та лінгвістична частина мають вирішальне значення для точності транскрипції [3]. На рисунку 1.1 показано блок-схему типової системи перетворення голосу в текст.

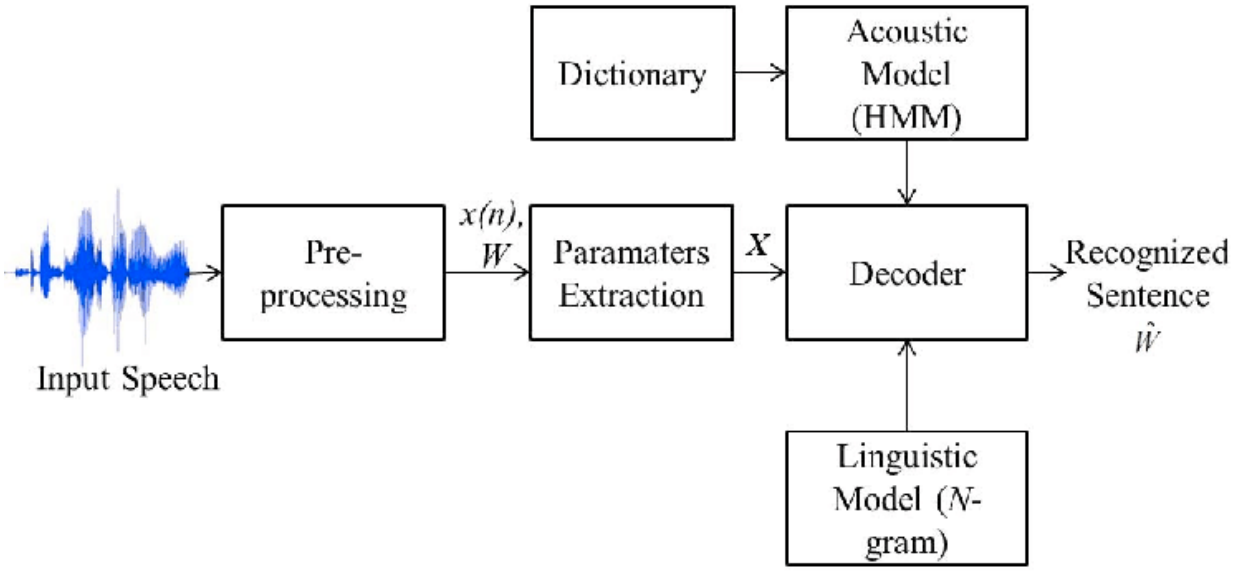

Рисунок 1.1 – Типова схема Speech to text

1.2 Реферування тексту

Реферування тексту – це завдання стиснення фрагменту тексту до більш короткого варіанту, зменшення розміру початкового тексту при одночасному збереженні ключових інформаційних елементів і змісту. Оскільки узагальнення тексту вручну є трудомістким за часом і, як правило, трудомістким завданням, автоматизація завдання набуває все більшої

популярності і тому є сильною мотивацією для академічних досліджень [4].

Існують важливі завдання для узагальнення тексту, пов’язаних з NLP, таких як класифікація текстів, відповіді на запитання, узагальнення юридичних текстів, узагальнення новин та створення заголовків. Крім того, створення резюме може бути інтегровано в ці системи як проміжний етап, який допомагає зменшити довжину документа [5].

В епоху великих даних відбувся вибух у кількості текстових даних з різних джерел. Даний обсяг тексту є неоціненним джерелом інформації та знань, які необхідно ефективно узагальнити, щоб він був корисним. Зростаюча доступність документів потребує вичерпного дослідження в області обробці природної мови для автоматичного узагальнення тексту. На рисунку 1.2 показано схему типового робочого процесу сумаризації тексту.

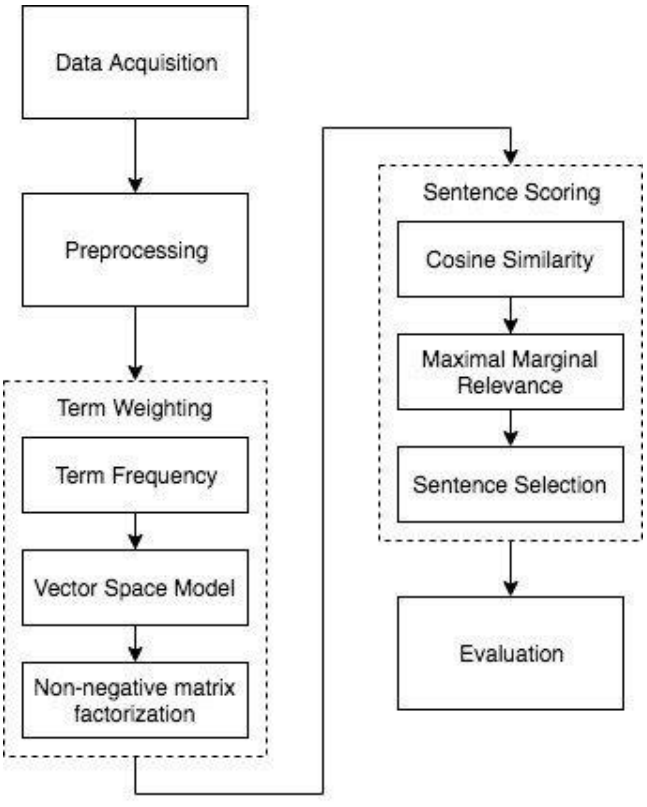

Рисунок 1.2 – Робочий процес сумаризації тексту

Для цього завдання були запропоновані різні моделі, засновані на машинному навчанні. Більшість із цих підходів моделюють цю проблему як

проблему класифікації, яка приймає рішення, чи включати речення до резюме чи ні. Інші підходи використовували інформацію про тему, латентний семантичний аналіз (LSA), моделі від послідовності до послідовності, навчання з підкріпленням і процеси змагальності.

1.3 Огляд існуючих програмних рішень

В даний час існують деякі високотехнологічні рішення від різних технологічних компаній. Але кожне з рішень має свої переваги і недоліки, які розглянуті нижче.

1.3.1 Dragon Anywhere

Dragon Anywhere – це програмне забезпечення для розпізнавання голосу. Дане рішення дозволяє користувачеві диктувати великі документи без обмежень на час диктування або номери сторінок. Якщо зроблено помилку під час диктування, існує можливість виправити її або відредагувати попереднє речення за допомогою простих голосових команд, наприклад «виправити». Меню корекції, яке з’явиться, надасть контекстний список альтернативних фраз для вибору.

Переваги Dragon Anywhere:

- висока точність розпізнавання голосу (~ 99%);
- не має ліміту на кількість слів;
- декілька шляхів для обміну документами.

Недоліки Dragon Anywhere:

- відсутність можливості сумаризації тексту;
- ціни можуть бути непомірно високими;
- для вивчення вбудованих команд може знадобитися час.

1.3.2 Amazon Transcribe

Amazon Transcribe – це служба автоматичного розпізнавання мовлення, яка дозволяє легко додавати можливості мовлення до тексту до будь-якої програми. Функції Transcribe надають змогу отримувати аудіо, створювати легко читані й переглядати стенограми, підвищувати точність за допомогою налаштування та фільтрувати вміст, щоб забезпечити конфіденційність клієнтів.

До переваг Amazon Transcribe можна віднести:

- висока точність розпізнавання голосу;
- можливість взаємодії з іншими рішеннями екосистеми Amazone.

Однак у Amazon Transcribe є і мінуси:

- висока вартість використання;
- відсутність можливості сумаризації тексту (існує можливість використання окремих модулів, що призведе до збільшення вартості використання);
- потрібне розуміння екосистеми AWS.

1.3.3 QuillBot

QuillBot – це інструмент для перефразування й узагальнення, який допомагає мільйонам студентів і професіоналів скоротити час письма більш ніж наполовину, використовуючи найсучасніший AI для переписування будь-якого речення, параграфа чи статті. Має як безкоштовну так і преміум версію. Також є доступ до використання API.

Плюси:

- існує можливість додати браузерне роширення;
- простота використання.

Мінуси:

- працює тільки з англійською мовою;

- відстуність можливості надиктовувати текст;
- має обмеження в використані в безкоштовній версії.

1.3.4 Результати аналізу існуючих систем

Розглянутим програмним рішенням притаманні такі найбільш поширені недоліки:

- висока вартість;
- обмежена підтримка мов;
- відсутня можливість модифікувати рішення.

Результати аналізу існуючих рішень представлені в таблиці 1.1. Знак «+» означає наявність функціоналу, знак «–» – відсутність можливості.

Таблиця 1.1 – Порівняння існуючих рішень

Критерії	Програмні рішення		
	Amazon Transcribe	Dragon Anywhere	QuillBot
Підтримка обробки голосу в текст	+	+	–
Підтримка функціоналу сумаризації	–	–	+
Функціонал працює з українською мовою	+	–	–
Функціонал працює з англійською мовою	+	+	+
Варіативність режимів роботи	–	–	–

1.4 Постановка задачі

Метою кваліфікаційної роботи є створення гібридної моделі, яка надає можливість розпізнання мовлення, перетворення наявних даних в текст і останнім етапом проведення сумаризації даного тексту, що містить лекційний матеріал, в текстовий вигляд, зберігаючи лише важливу змістовну частину лекції. А також порівняти ефективність нейромережевих моделей використаних в розробці запропонованої системи на різних GPGPU. На рисунку 1.3 зображені запропоновані основні модулі роботи комплексу.

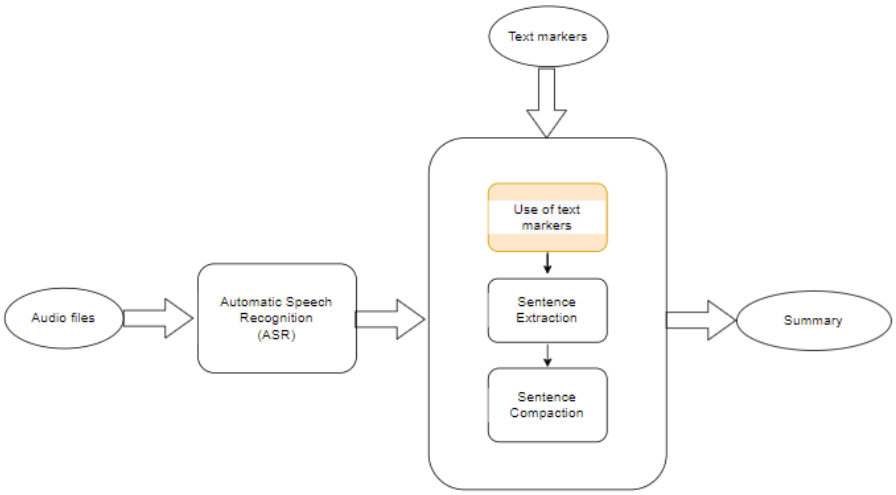

Рисунок 1.3 – Основні модулі пропонованої системи

Оскільки достовірність інформації, що міститься в освітніх ресурсах дистанційних курсів, є однією з ключових вимог до цифрових освітніх платформ, скорочення появи хибних або спотворених даних під час перетворення звукового ряду на текстові дані для подальшого семантичного аналізу є головною метою роботи.

Новизною даного дослідження є використання слів маркерів на етапі сумаризації тексту, а також порівняння ефективності обробки даних на різних етапах роботи даної моделі при використанні різного апаратного забезпечення.

Важливою особливістю рішення, що планується розробити є розподілена обробка даних та масштабованість, що дозволяє використовувати даний підхід при обробці великих об'ємів даних. Задача обробки голосу та тексту потребує значних ресурсів і для оптимізації та пришвидшення даного завдання використовується паралельне виконання на GPGPU. Що надає можливість ефективно виконувати розпаралелювання задачі NLP на великих об’ємах вхідних даних. Також гостро стоїть питання можливісті розгортання даної моделі в хмарних рішеннях, таких як Amazon Web Services або Google Cloud Platform для запобігання втрати даних у випадку війни або природніх катаклізмів. Хмарні рішення забезпечують надійність зберігання та обробки даних, за рахунок створення знімків даних і їх реплікації між серверами, які мають різні географічні локації [6]. А також у випадку відстуності необхідного апаратного забезпечення – використання виділених потужностей хмарних рішень.

Для досягнення поставленої мети, мають бути вирішені наступні задачі:

- розглянути і порівняти існуючі моделі для перетворення мовлення в текст;
- розглянути і порівняти існуючі моделі для сумаризації тексту;
- забезпечити можливість працювати вище перелічним моделям з українською мовою;
- оцінити якість роботи даних моделей з українською та англійською мовами;
- оцінити таймінгу роботи розробленої системи;
- проаналізувати отримані результати.

2 АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ СЕМАНТИЧНОГО АНАЛІЗУ

Для виконання поставленої задачі було прийнято використовувати мову програмування Python. Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднування наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована. Дана мова має широкий функціонал для роботи з нейронними мережами.

2.1 Розпізнавання мовлення

Транскрипція мовленнєвих документів, таких як виступи, презентації, лекції та телевізійні новини, є одним із найважливіших застосувань автоматичного розпізнавання мовлення. Хоча мовлення є найбільш природною та успішною формою людського спілкування, важко швидко оцінити, отримати та повторно використати мовні документи, які просто записуються як звукові сигнали. Хоча високої точності розпізнавання можна досягти за допомогою мовлення, зчитованого з тексту, наприклад, мовлення ведучих мовців, що транслюють новини, здатність технології розпізнавати спонтанне мовлення наразі обмежена. Мимовільне мовлення неструктуроване і несхоже на письмовий матеріал. У спонтанному мовленні часто зустрічається повторювана інформація, така як наповнювачі, повтори,

виправлення та фрагменти слів. Крім того, це нормально, якщо сторонній матеріал вставляється в транскрипцію в результаті помилок розпізнавання. В результаті для спонтанного мовлення метод, при якому записуються всі слова, неефективний. Для виявлення спонтанного мовлення необхідне узагальнення мовлення, яке витягує релевантну інформацію, видаляючи зайву та неправильну інформацію. Підведення підсумків промови скорочує час читання мовленнєвих документів і підвищує ефективність пошуку документів.

Першим компонентом розпізнавання мовлення є, безумовно, мовлення. Мовлення необхідно перетворити з фізичного звуку в електричний сигнал за допомогою мікрофона, а потім у цифрові дані за допомогою аналого-цифрового перетворювача. Після оцифрування кілька моделей можна використовувати для транскрибування аудіо в текст.

Приховані ланцюги Маркова або глибокі нейронні мережі, такі як рекурентні нейронні мережі та згорткові нейронні мережі, є найпоширенішими способами вирішення цієї проблеми. У наступних підрозділах надано детальний огляд кожного з цих підходів. Наступні критерії будуть використовуватися для порівняння основних методів і підходів, таких як кількість даних, необхідних для методу навчання, швидкість навчання та точність розпізнавання.

2.1.1 Розпізнавання мовлення за допомогою динамічного викривлення часу (DTW)

Динамічне викривлення часу колись було популярним для розпізнавання мови, але тепер його в основному замінили більш ефективними методами на основі HMM.

Динамічне викривлення часу – це метод визначення подібності двох послідовностей, що відрізняються за часом або швидкістю. Схожість у моделях ходьби, наприклад, можна було б помітити, навіть якщо людина

ходила повільно в одному фільмі і швидше в іншому, або якщо б протягом одного спостереження спостерігалися прискорення та уповільнення. DTW використовувався для аналізу відео, аудіо та зображень, а також будь-яких даних, які можна перетворити в лінійне представлення [7].

Автоматичне розпізнавання голосу є добре відомим завданням для роботи зі змінною частотою мовлення. Це метод, який дозволяє комп'ютеру визначати найкращу відповідність між двома наданими послідовностями (наприклад, часовими рядами) у межах певних обмежень. Іншими словами, послідовності «викривляються» нелінійно. Цей підхід вирівнювання послідовності часто використовується в прихованих марковських моделях.

2.1.2 Прихована модель Маркова та GMM

Нові алгоритми машинного навчання можуть призвести до значного прогресу в автоматичному розпізнаванні мовлення (ASR). Найбільший прогрес відбувся майже чотири десятиліття тому з введенням алгоритму максимізації очікування (EM) для навчання HMM. За допомогою алгоритму EM стало можливим розробити системи розпізнавання мовлення для завдань реального світу, використовуючи багатство GMM для представлення взаємозв'язку між станами HMM та акустичним входом. У цих системах акустичний вхід, як правило, представлений конкатенацією кепстральних коефіцієнтів частоти Mel (MFCC) або коефіцієнтів перцептивного лінійного прогнозування (PLP) [8], обчислених на основі вихідної форми хвилі та їх тимчасових відмінностей першого та другого порядку [9].

Ця неадаптивна, але високотехнологічна попередня обробка сигналу призначена для відкидання великої кількості інформації в формах сигналу, яка вважається нерелевантною для розрізнення, і для вираження решти інформації у формі, яка полегшує дискримінацію з GMM-HMM.

GMM мають ряд переваг, які роблять їх придатними для моделювання розподілів ймовірностей над векторами вхідних ознак, які пов'язані з кожним

станом HMM. Маючи достатню кількість компонентів, вони можуть моделювати розподіл ймовірностей з будь-яким необхідним рівнем точності, і їх досить легко підігнати до даних за допомогою алгоритму EM. Величезна кількість досліджень була спрямована на пошук шляхів обмеження GMM, щоб збільшити швидкість їх оцінки та оптимізувати компроміс між їх гнучкістю та кількістю навчальних даних, необхідних, щоб уникнути серйозного переобладнання [10].

Точність розпізнавання системи GMM-HMM може бути додатково покращена, якщо її дискримінаційно тонко налаштувати після її генеративного навчання, щоб максимізувати її ймовірність генерування спостережуваних даних, особливо якщо дискримінаційна цільова функція, що використовується для навчання, тісно пов’язана з частота помилок у телефонах, словах чи реченнях. Точність також можна підвищити, доповнюючи (або об’єднуючи) вхідні функції (наприклад, MFCC) за допомогою «тандемних» або вузьких функцій, створених за допомогою нейронних мереж [11].

GMM настільки успішні, що будь-якому новому методу важко перевершити їх для акустичного моделювання. Незважаючи на всі переваги, GMM мають серйозний недолік – вони статистично неефективні для моделювання даних, які лежать на нелінійному різноманітті або поблизу нього в просторі даних. Наприклад, моделювання набору точок, які лежать дуже близько до поверхні сфери, вимагає лише кількох параметрів з використанням відповідного класу моделі, але вимагає дуже великої кількості діагональних гаусів або досить великої кількості повноковаріаційних гауссів. Мовлення створюється шляхом модуляції відносно невеликої кількості параметрів динамічної системи, і це означає, що її справжня базова структура набагато нижча за розмірами, ніж це відразу видно у вікні, яке містить сотні коефіцієнтів. Тому ми вважаємо, що інші типи моделей можуть працювати краще, ніж GMM.

Але багато сучасних систем розпізнавання мовлення спираються на так

звану модель прихованої Маркова (HMM). Цей підхід ґрунтується на припущенні, що мовний сигнал, якщо розглядати його на досить короткому часовому масштабі (скажімо, десять мілісекунд), можна розумно апроксимувати як стаціонарний процес, тобто процес, у якому статистичні властивості не змінюються з часом.

У типовому HMM мовний сигнал розбивається на 10-мілісекундні фрагменти. Спектр потужності кожного фрагмента, який по суті є графіком залежності потужності сигналу від частоти, відображається у вектор дійсних чисел, відомий як кепстральні коефіцієнти. Розмір цього вектора зазвичай невеликий – іноді до 10, хоча більш точні системи можуть мати розмірність 32 або більше. Кінцевим результатом HMM є послідовність цих векторів.

Щоб розшифрувати мовлення в текст, групи векторів узгоджуються з однією або кількома фонемами – основною одиницею мовлення. Цей розрахунок вимагає підготовки, оскільки звучання фонеми різниться від мовця до мовця і навіть змінюється від одного висловлювання до іншого одним і тим же мовцем. Потім використовується спеціальний алгоритм, щоб визначити найбільш вірогідне слово (або слова), які утворюють задану послідовність фонем.

Можна уявити, що весь цей процес може бути дорогим з точки зору обчислень. У багатьох сучасних системах розпізнавання мовлення нейронні мережі використовуються для спрощення мовного сигналу за допомогою методів перетворення ознак і зменшення розмірності перед розпізнаванням HMM. Детектори голосової активності (VAD) також використовуються для зменшення звукового сигналу лише до тих частин, які, ймовірно, містять мовлення. Це запобігає розпізнавачу витрачати час на аналіз непотрібних частин сигналу [12].

Використання цієї моделі для вирішення проблем має також ряд переваг. Метою навчання параметрів у прихованих марківських моделях є визначення оптимального набору ймовірностей для заданої множини учасників набору переходів станів і виходів для даної множини учасників. З

глибокими нейронними мережами цей метод також демонструє швидше навчання. Співвідношення точності з іншими підходами є одним із недоліків цього підходу. На рисунку 2.1 зображений приклад прихованої моделі Маркова для розпізнавання голосу.

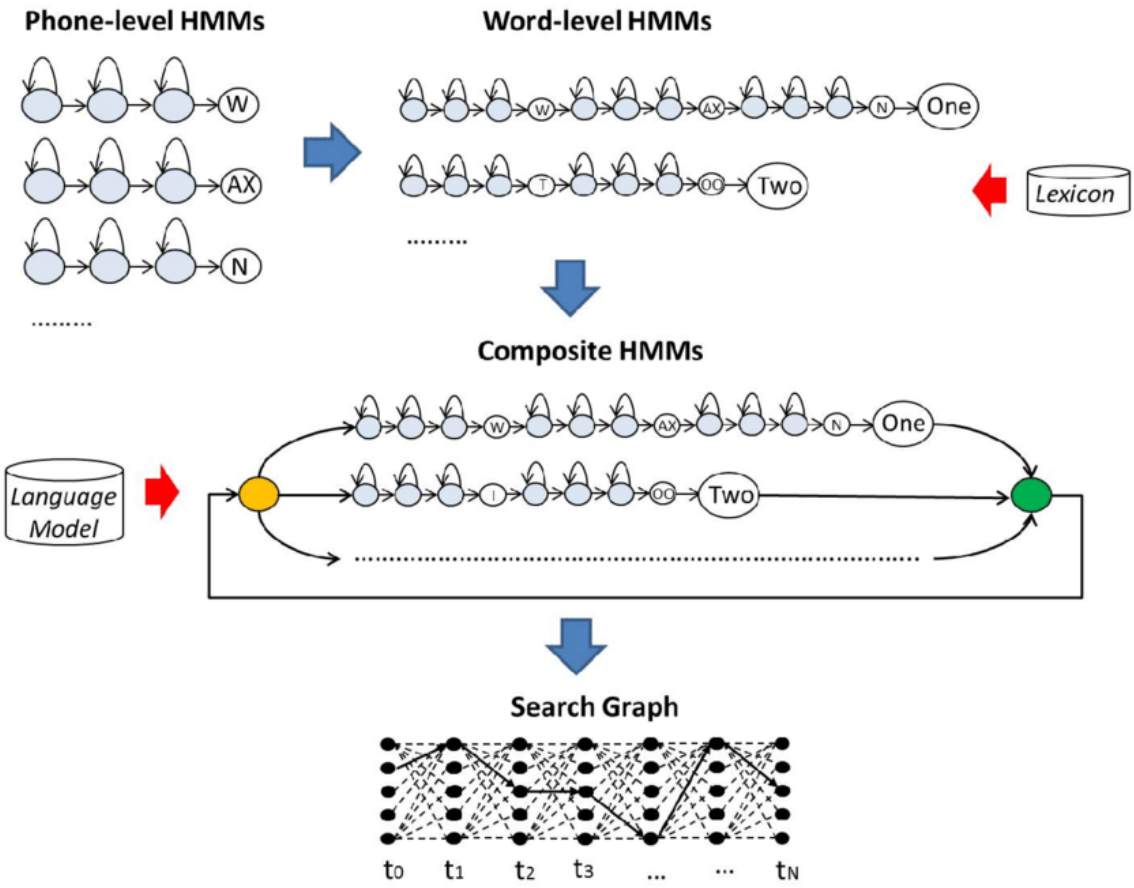

Рисунок 2.1 – Прихована модель Маркова для розпізнавання голосу

2.1.3 Використання глибокої нейронної мережі в процесі розпізнавання мовлення

Більшість сучасних систем розпізнавання мовлення використовують приховані моделі Маркова (HMM), в яких потрібно мати справу з тимчасовою мінливість мови та моделяї суміші Гаусса (GMM), щоб визначити, наскільки добре кожен стан кожного HMM відповідає фрейму або короткому вікну коефіцієнтів фреймів, що представляє акустичний вхід.

Альтернативний шлях оцінки відповідності полягає у використанні нейронної мережі з прямим зв'язком, яка приймає декілька кадрів коефіцієнтів в якості вхідних даних та створює апостеріорні ймовірності щодо станів HMM як вихідні дані. Було досліджено, що глибокі нейронні мережі (DNN), які мають багато прихованих шарів і навчаються за допомогою нових методів, перевершують GMM на різноманітних тестах розпізнавання мовлення, іноді з великим відривом.

DNN – це штучна нейронна мережа з прямим зв'язком, яка має більше ніж один шар прихованих одиниць між входами та виходами. Кожна прихована одиниця, зазвичай використовує логістичну функцію (часто також використовується тісно пов'язаний гіперболічний тангенс і можна використовувати будь-яку функцію з правильною похідною), щоб відобразити свій загальний вхід із нижнього шару у скалярний стан, який він надсилає на шар вище. DNN можна навчати дискримінаційно (DT) шляхом зворотного поширення похідних функції вартості, яка вимірює невідповідність між цільовими результатами та фактичними результатами, виробленими для кожного навчального випадку. При використанні вихідної функції softmax функція натуральних витрат C є перехресною ентропією між цільовими ймовірностями d і результатами softmax, p,

$$C = -\sum_j d_j \log p_j \tag{2.1}$$

де цільові ймовірності, які зазвичай приймають значення одиниці або нуля, є контрольованою інформацією, яка надається для навчання класифікатора DNN.

DNN з багатьма прихованими шарами важко оптимізувати. Градієнтний спуск із випадкової початкової точки поблизу початку координат не є найкращим способом знайти хороший набір ваг, і якщо початкові масштаби ваг не будуть ретельно вибрані, градієнти, поширені назад, матимуть дуже різні величини в різних шарах. На додаток до проблем

оптимізації, DNN можуть погано узагальнювати дані тесту, що зберігаються. DNN з багатьма прихованими шарами і багатьма одиницями на шар є дуже гнучкими моделями з дуже великою кількістю параметрів. Це робить їх здатними моделювати дуже складні та дуже нелінійні зв'язки між входами та виходами. Ця здатність важлива для високоякісного акустичного моделювання, але вона також дозволяє моделювати помилкові закономірності, які є випадковою властивістю конкретних прикладів у навчальному наборі, що може призвести до серйозного переобладнання. Штраф за вагу або раннє зупинення можуть зменшити переобладнання, але лише за рахунок видалення значної частини сили моделювання. Дуже великі навчальні набори можуть зменшити переобладнання, зберігаючи потужність моделювання, але лише за рахунок того, що навчання дуже дорого коштує. Нам потрібен кращий метод використання інформації з навчального набору для створення кількох шарів нелінійних детекторів ознак. На рисунку 2.2 зображений приклад DNN з 5-вимірним введенням, 3-вимірними прихованими шарами та 7-вимірним виведенням. Кожен прихований шар повністю пов'язаний з попереднім і наступним шаром

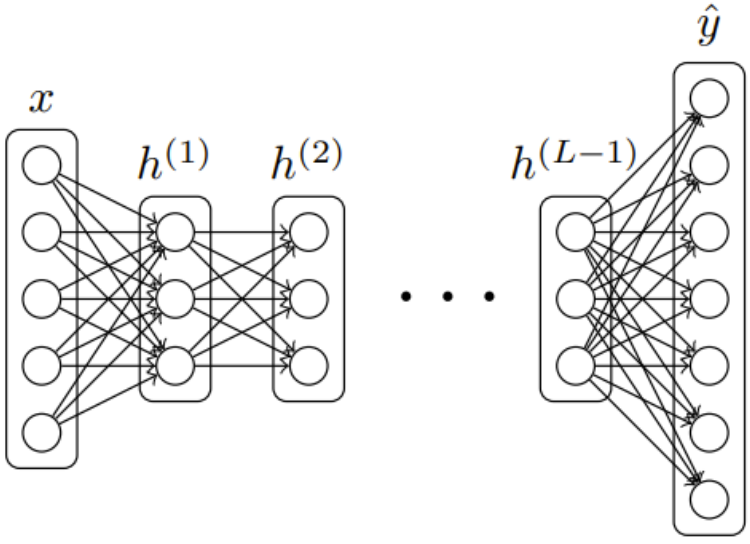

Рисунок 2.2 – Приклад DNN

2.1.4 Наскрізна модель обробки мовлення

Через вищезазначені недоліки моделі на базі HMM на додачу з розвитком технології глибокого навчання, все більше і більше напрацювань почали використовувати наскрізний LVCSR. Наскрізьна модель – це система, яка безпосередньо відображає вхідну звукову послідовність у послідовність слів або інших графем. Його функціональна структура показана на рисунку 2.3.

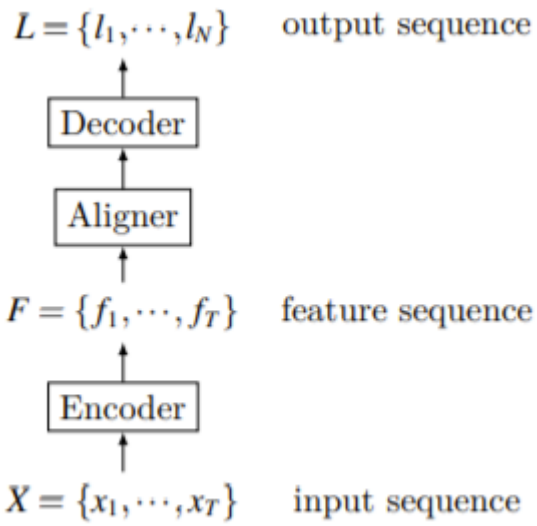

Рисунок 2.3 – Функціональна структура наскрізної моделі

Більшість моделей наскрізного розпізнавання мовлення включають такі частини:

- кодер, який відображає послідовність мовного введення в послідовність ознак;
- вирівнювач, який реалізує вирівнювання між об’єктами послідовність і мова;
- декодер, який декодує кінцевий результат ідентифікації.

Цей поділ існує не завжди, тому що наскрізна модель сама по собі є цілісною структурою, і це зазвичай дуже важко визначити, яка частина виконує яке підзавдання.

На відміну від моделі на основі HMM, яка складається з кількох модулів, наскрізна модель замінює кілька модулів глибокою мережею, реалізуючи пряме відображення акустичних сигналів у послідовності міток без ретельно розроблених проміжних станів. Крім того, немає потреби виконати попередню обробку даних на виході.

Порівняно з моделлю на основі HMM, зазначені вище відмінності надають наскрізному LVCSR наступні характеристики:

- кілька модулів об'єднані в одну мережу для спільного навчання. Перевага об’єднання кількох модулів полягає в тому, що немає необхідності розробляти багато модулів, щоб реалізувати відображення між різними проміжні стани [14]. Спільне навчання дає змогу наскрізній моделі використовувати функцію, яка є дуже релевантною критеріям остаточної оцінки як глобальній цілі оптимізації;

- він безпосередньо зіставляє вхідну послідовність акустичного підпису з послідовністю результатів тексту і не вимагає подальшої обробки для досягнення справжньої транскрипції або покращення ефективності розпізнавання [15], тоді як у моделях на основі HMM зазвичай є внутрішнє представлення для відтворення ланцюга символів;

- ці переваги наскрізної моделі LVCSR дозволяють значно спростити конструкцію та навчання моделей розпізнавання мовлення. У різному ступені завдання послідовності стикаються з проблемами вирівнювання даних, особливо для розпізнавання мови.

У наскрізній моделі використовується “м’який підхід” до вирівнювання. Кожен аудіокадр відповідає всім можливим станам з певним розподілом ймовірності, що не потребує вимушеної, явної кореспонденції. Наскрізну модель можна розділити на три різні категорії залежно від них реалізації м'якого вирівнювання:

- на основі CTC: CTC спочатку перераховує всі можливі жорсткі вирівнювання (представлені шляхом концепції), потім він досягає м'якого вирівнювання шляхом об'єднання жорстких вирівнювань. CTC припускає

вихід мітки, які не залежать одна від одної під час перерахування жорстких вирівнювань;

- RNN-перетворювач: він також перераховує всі можливі жорсткі вирівнювання, а потім агрегує їх для м'яке вирівнювання. Але на відміну від CTC, RNN-перетворювач не робить самостійних припущень про мітки під час перерахування жорстких вирівнювань, тому він відрізняється від CTC з точки зору визначення шляху і розрахуноку ймовірності;

- на основі уваги: цей метод не перераховує всі можливі жорсткі вирівнювання, а використовує механізм уваги для безпосереднього обчислення інформації м’якого вирівнювання між вхідними даними і вихідна обгортками.

2.2 Підходи для сумаризації тексту

Попередня обробка вхідних даних є важливим і необхідним кроком у завданнях обробки природної мови, нарівні з класифікацією та розпізнаванням. Метод повторної обробки часто використовується для вилучення інформації з неструктурованих текстових даних. У текстах часто зустрічаються дати, числівники та слова без семантичного змісту, такі як прийменники, артиклі та займенники. Токенізація є першим етапом обробки, це процес розбиття тексту на більш дрібні фрагменти, або маркери. Слова та розділові знаки використовуються як лексеми.

Нормалізація є наступним кроком. Нормалізація – це процес перетворення тексту в єдину канонічну форму, який раніше був неможливим. Для процесу нормалізації необхідно зрозуміти, який текст буде оброблятися наступним, а який нормалізуватися.

Тексти часто містять кілька граматичних версій одного слова, а також односкладові терміни. Метою лематизації та стеммування є об’єднання всіх форм слова в єдину стандартну словникову форму. Стемінг – це грубий евристичний процес, який відрізає «зайве» від кореня і виділяє основну

частину слова, часто це призводить до втрати словотворчих суфіксів так як основа не обов'язково збігається з морфологічним коренем слова. Реферування або підсумовування – це процес скорочення набору даних для створення підмножини (резюме), що представляє найважливішу або релевантну інформацію в межах оригінального вмісту. Існує два загальні підходи до автоматичного реферування: вилучення (екстрактивний підхід) та абстрагування (абстрактивний підхід).

При вилученні вміст витягується з вихідних даних, але витягнутий вміст жодним чином не змінюється. Приклади вилученого вмісту включають ключові фрази, які можна використовувати для "тегування" або індексації текстового документа, або ключових позицій (включаючи заголовки), які в сукупності містять абстрактні, і репрезентативні зображення або відео сегменти. Методи даного підходу характеризують наявність оціночної функції важливості інформаційного блоку. Як правило, важливість речення визначається важливістю слів у ньому. Ранжуючи ці блоки за ступенем важливості і вибираючи необхідну їй кількість, ми формуємо підсумкове резюме тексту. Екстрактивні методи збирають резюме виключно з уривків (зазвичай цілих речень), узятих безпосередньо з вихідного тексту, в той час як абстрактні методи можуть генерувати нові слова та фрази, які не фігурують у вихідному тексті – як зазвичай робиться анотація, написана людиною. Екстрактивний підхід легший, оскільки копіювання великих шматків тексту з вихідного документа забезпечує базові рівні граматичності та точності. З іншого боку, витончені підходи, які є вирішальними для високоякісного узагальнення, такі як перефразування, узагальнення або включення реальних знань, можливі лише в абстрактній структурі.

Абстрактивні методи будують внутрішнє смислове подання оригінального тексту, а потім використовують це уявлення для створення реферату, ближчого до того, що може виразити людина. Якщо абстрактивні методи застосовуються для узагальнення тексту в проблемах глибокого навчання, вони може подолати граматичні невідповідності екстрактивного

методу. Алгоритми абстрактивного узагальнення тексту створюють нові фрази та речення, які передають найкориснішу інформацію з оригінального тексту – як і люди.

Тому абстракція працює краще, ніж вилучення. Однак алгоритми узагальнення тексту, необхідні для абстрагування, надзвичайно складні у розробці, тому використання вилучення все ще популярне. Основними методами вилучення є ранжування і відсікання. Зазвичай застосовують один з двох підходів – або використання будь-яких евристичних формул, які дозволяють визначити, чи є слово ключовим, або використання методів машинного навчання. Варто зазначити, що для машинного навчання з учителем необхідний попередньо розмічений корпус документів з виділеними ключовими словами. Спочатку застосування машинного навчання для виділення ключових термів зводилося до вирішення завдання бінарної класифікації шляхом різних підходів до навчання класифікатора. Використовувалися наївні класифікатори Баєса, дерева прийняття рішень, бустінг. Однак такий підхід не дозволяє порівнювати знайдені терми між собою і вибирати кращі з них. Тому, згодом почали застосовуватися алгоритми, що дозволяють ранжувати терми попарно (наприклад, алгоритм KEA, TextRank).

2.2.1 Статистичні підходи

Використання статистичних методів для узагальнення тексту є ефективним підходом, який використовувався у багатьох статтях. У статистичних методах важливі речення вибираються на основі частоти слів, індикаторних фраз та інших ознак незалежно від значення слів, згаданих у попередніх розділах. Існує кілька методів визначення ключових речень, таких як метод заголовка , метод розташування, Метод подібності агрегації [16], метод частоти, метод запиту на основі TF, та латентно-семантичний аналіз. Але найпоширенішими методами є байєсівський класифікатор

(Bayesian Classifier) і підходи зв'язків концепції зв'язку. Одним із найвідоміших статистичних методів узагальнення тексту є метод LSA [17]. LSA – це алгебраїко-статистичний метод, який виділяє приховані структури значення слів і речень [18]. Цей підхід є методом без нагляду, який витягує текстові структури лише на основі інформації про слова в реченні без необхідності будь-яких інших знань. Ідея в LSV полягає в тому, що слова, спільні між різними реченнями, є причиною значеньної залежності. LSA також може показувати значення слів і лексики одночасно. Він використовує метод алгебраїчної декомпозиції сингулярних значень (SVD), щоб визначити зв’язок між реченнями та словами. На додаток до можливості моделювати зв’язок між словами та реченням, SVD також може зменшити шум і підвищити точність.

Алгоритм узагальнення на основі методу LSA складається з трьох кроків: створення вхідної матриці, застосування методу SVD до створеної матриці та вилучення речення. LSA також має деякі обмеження. Найважливіші з них такі:

- алгоритм не використовує інформацію про розташування слів у реченнях, граматику та морфологічні співвідношення. Однак ця інформація може бути корисною для кращого розуміння слів і речень.;
- алгоритм не використовує знання слів і базу даних слів;
- збільшуючи кількість різних слів і різнорідних даних, продуктивність алгоритму значно знижується. Зниження продуктивності пов'язано з часом і складністю пам'яті методу SVD.

2.2.2 Підхід на основі лексичного ланцюжка

Лексичний ланцюжок створює подання суміжних структур тексту. Даний алгоритм використовує слово net database для визначення зв’язку між термінами, а потім створює континуум між цими термінами. Оцінки, які враховуються для термінів, залежать від типу та кількості зв’язків набору

ланцюжків. Остаточне резюме включає речення з найсильнішим ланцюжковим зв’язком. Лексичні ланцюжки можна обчислити семантично відношення слів. Він також може ідентифікувати синоніми та гіпоніми, щоб розмістити їх у групі в одному лексичному ланцюжку. Лексичний ланцюжок також використовується для пошуку інформації та виправлення граматичних помилок. У підходах до лексичного ланцюга є два недоліки. Перше – багатозначність у ланцюжку слів. Якщо деякі слова мають семантичну неоднозначність, створений ланцюжок також матиме семантичну неоднозначність. Другим недоліком є відношення створеного ланцюга до основної теми. Усі ланцюжки не пов’язані з основною темою.

2.2.3 Морфологічні операції

Морфологічна обробка зображення – це сукупність нелінійних операцій, пов’язаних із формою чи морфологією об’єктів зображення. Морфологічні операції покладаються лише на відносне впорядкування значень пікселів, а не на їх числові значення, і тому особливо підходять для обробки двійкових зображень. Морфологічні операції також можна застосувати до зображень у відтінках сірого, так що їх функції передачі світла невідомі, і тому їх абсолютні значення пікселів не представляють інтересу або не мають значення[19].

Морфологічні методи досліджують зображення за допомогою невеликої форми або шаблону, який називається структурним елементом. Елемент структуризації розташовується в усіх можливих місцях на зображенні і порівнюється з відповідним оточенням пікселів. Деякі операції перевіряють, чи елемент «вписується» в околиці, тоді як інші перевіряють, чи він «входить» чи перетинає околиці. Морфологічна операція над двійковим зображенням створює нове двійкове зображення, в якому піксель має ненульове значення, тільки якщо тест успішно пройшов у цьому місці вхідного зображення.

Елементом структуризації є невелике двійкове зображення, тобто невелика матриця пікселів, кожен з яких має значення нуль або одиницю:

- розміри матриці визначають розмір елемента структури;
- шаблон одиниць і нулів задає форму структурного елемента;
- початком координат елемента структуризації зазвичай є один з його пікселів, хоча загалом початок може бути поза структурним елементом [13].

2.2.4 Підхід на основі графів

Підхід на основі графів передбачає методи узагальнення тексту з використанням теорії графів. Після звичайних кроків попередньої обробки, таких як виділення коренів і видалення стоп-слова, пропозиції в документах представлені у вигляді вузлів орієнтованого графа. Речення з'єднуються між собою ребрами відповідно до речення. Основна ідея підходів на основі графіків – це щось на зразок голосування. У результаті, коли ребро з'єднало вузол з іншим вузлом, це означає, що голоси за нього, і незалежно від того, наскільки високим є вхідний ступінь вузла, воно має вищий пріоритет. У згаданому способі також є оцінка голосування. Якщо вузол має більший вихідний ступінь, його важливість зростає. Ступінь важливості кожного вузла розраховується за допомогою формули (2) для всіх вузлів у графі рекурсивно. Процес триває до тих пір, поки всі варіанти не буде охоплено і $S(V_a)$ не зміниться.

$$S(V_a) = (1-d) + d * \sum V_b \in \ln(V_a) \frac{S(V_b)}{out(V_b)}, \quad (2.2)$$

Алгоритм також можна застосувати до неорієнтованого графа, але підсумок результату відрізняється більшою складністю часу. Ефективним запропонованим алгоритмом зведення на основі графів є алгоритм рангу тексту. Він використовує метод без нагляду для виділення ключових слів і речень із оцінкою вузлів на основі вищезгаданих показників подібності. Алгоритм запускається з необов'язкових значень вузла і рекурсивно повторюється, поки охоплення не досягне попередньо визначеного порогу.

Необов’язкове значення вузлів не впливає на підсумкові бали.

2.2.5 Кластерні підходи

Деякі автоматичні системи підсумовування використовують кластери для створення значущих підсумків. У цьому підході застосовуються різні алгоритми кластеризації для поділу тексту на лексеми, такі як слова, фрази, речення і навіть абзаци. Підхід, заснований на кластері, являє собою екстрактивне узагальнення, і найбільш схожі речення, засновані на згаданих показниках подібності, поміщаються в кластери. Найбільший кластер вибирається як основна тема, а його речення виділяються та розміщуються в підсумковому виведенні. Для визначення подібності та несхожості кластерів використовуються евклідова відстань, декартова подібність, косинус та деякі інші міри подібності. Документи представлені за допомогою терміна частотно-інверсної частоти документа (TF-IDF). У контексті частота термінів (TF) – це середня кількість випадків (за документами) у кластері. Тема представлена словами, значення TF-IDF яких вище в кластері. Вибір важливих речень ґрунтується на мірі подібності речень із темою скупчення. В їх алгоритмі запропоновано дві ідеї оптимізації. Перший – це метод обчислення оцінки речення, а другий – спосіб отримати оптимальну кількість кластерів. Для обчислення оцінки пропозиції алгоритм використовує TF-IDF і довжину пропозиції (X).

$$Score(X) = \frac{\sum_t tf-idft}{|x|}, \tag{2.3}$$

Одним з найважливіших питань у кластеризації k-середніх є визначення оптимальної кількості кластерів. Враховуючи, що в цьому методі найбільший кластер розглядається як основна тема, розмір введеного тексту має безпосередній вплив на визначення кількості кластерів. Наприклад, велике k призводить до невеликих кластерів і, як наслідок, невелике та розсіяне резюме з дуже низькою кореляцією, а мале k призводить до великих

і щільних кластерів і, як результат, низького стиснення тексту.

2.2.6 Підходи, засновані на нечіткій логіці

Підходи на основі нечіткої логіки розглядають кожну характеристику тексту як вхід нечіткої системи. Ці методи використовували лише нечітку логіку для виявлення та вилучення важливих речень. Методи узагальнення нечіткого тексту відрізняються за відмінностями у виділенні ознак, нечіткими правилами, лінгвістичними змінними, функціями належності, методами нечіткості та дефазфікації. Деякі зміни в числових значеннях призводять до отримання кращих лінгвістичних змінних Ф. Кьомарсі, Х. Хосраві запропонували метод, який узагальнює текст у два етапи. Перший етап – попередня обробка, а другий – нечіткий аналіз тексту. Ознаками, які вони виділили, є: кількість загальних слів, включаючи речення та заголовок, кількість слів у реченні, подібність речення з реченням теми абзацу та подібність речення та першого речення в абзаці.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЕКСПЕРЕМЕНТУ

3.1 Огляд запропонованої схеми комплексу

В даній дипломній роботі досліджується використанням декількох моделей для перетворення голосу в текст, а також проведення сумаризації отриманого текстового контенту. Для виконання поставленої задачі було прийнято розробити модульну мікросервісну архітектуру, яка б забезпечувала можливість зміни семантичних моделей без значного впливу на весь комплекс [20].

Модуль Spech to text на вхід приймає звуковий запис в форматі WAV з частотою 16 kHz, оскільки це є обмеженням роботи перетворення звукового ряду, оскільки він вимагає високої якості вхідних даних. В деяких моделях для обробки звуковго ряду гостро стоїть задача, щодо отримання аудіозапису без сторонього шуму [21], яка не є імплементованою в деякі з рішень. Так само і у вибраній моделі для дослідженя дана функціональсті є відсутньою. Тому було прийнято використовувати підхід для очищення звукового ряду за допомогою глибокої нейронної мережі, який описаний у дослідженні проведеним Yong Xu, Jun Du, Li-Rong Dai, Chin-Hui Lee і має назву “A Regression Approach to Speech Enhancement Based on Deep Neural Networks”.

Модуль сумаризації тексту отримує на вхід результат від роботи модуля розпізнавання мовлення (ASR) у вигляді JSON об’єкту. Також в даному досліджені запропонована ідея щодо створення сервіса для фільтрації вхідного тексту за допомогою використання слів маркерів. Фільтрація тексту відбувається на етапі переходу JSON об’єкту від модуля ASR до модуля сумаризації тексту.

На рисунку 3.1 зображена схема запропонованої системи. Дана схема складається з вищезазначених модулів: speech-to-text і summarization module.

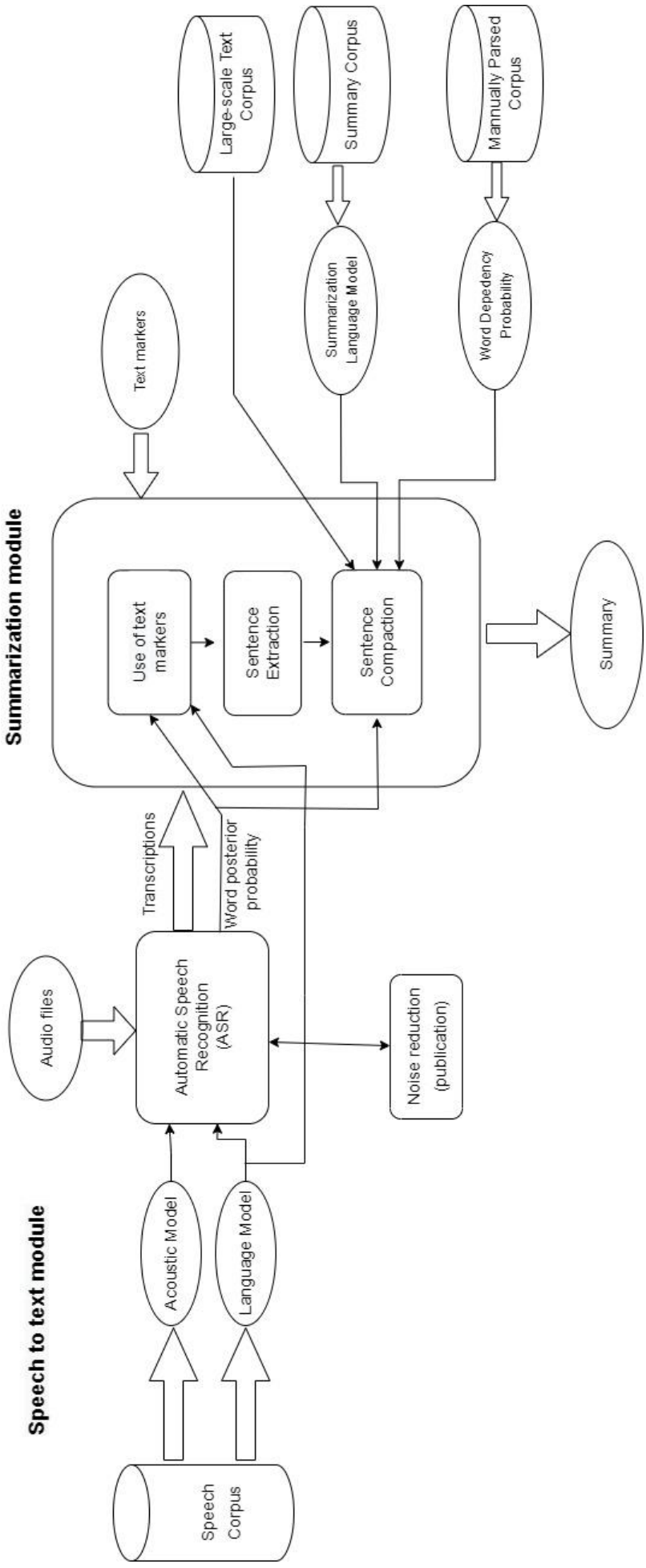

Рисунок 3.10 – Запровонована архітектура комплексу

3.2 Модуль Speech to text

3.2.1 Підмодуль ASR

Для даного модуля тестувалося 2 моделі. Для англійської мови це wav2vec2-xls-r-1bз, яка є опенсорсною моделю від компанії Facebook і доступна в загальному доступі. А для української мови було обрано Ukrainian STT model (wav2vec2-xls-r-1b-uk-with-lm with with Language Model), яка є доопрацьованою версією рішення від Facebook.

Традиційні моделі розпізнавання мовлення в першу чергу тренуються на аудіо з анотованим мовленням із транскрипцією. Хороші системи вимагають великих обсягів анотованих даних, які доступні лише для невеликої кількості мов. Самоувага надає спосіб використовувати неанотовані дані для створення кращих систем.

Інші самоконтрольовані підходи до мовлення намагаються відновити аудіосигнал, що вимагає від моделі захоплення кожного аспекту мовлення, включаючи середовище запису, шум каналу та особливості мовця. Інший поширений підхід полягає в тому, щоб навчити модель, попросивши її передбачити, що доповідач сказав далі, порівнявши кілька варіантів. Даний підхід вивчає набір мовних одиниць, які коротші за фонеми, для опису звукової послідовності мовлення. Оскільки цей набір скінченний, модель не може представляти всі варіації, наприклад фоновий шум. Натомість одиниці спонукають модель зосередитися на найважливіших факторах для представлення звуку мовлення.

Модель wav2vec2 спочатку обробляє необроблену форму сигналу мовного аудіо за допомогою багатошарової згорткової нейронної мережі, щоб отримати приховані звукові представлення тривалістю 25 мс кожне. Ця модель вивчає основні мовні одиниці, які використовуються для вирішення завдання, яке контролюється самоконтролем. Модель навчена передбачати правильну мовну одиницю для замаскованих частин аудіо, в той же час

вивчаючи, якими мають бути мовні одиниці. Завдяки лише 10 хвилинам транскрибованого мовлення та 53 тисячам годин мовлення без міток, wav2vec 2.0 дає змогу моделювати розпізнавання мовлення з коефіцієнтом помилок слів (WER) 8,6 відсотка для шумної мови та 5,2 відсотка для чистої мови за стандартним тестом LibriSpeech. На рисунку 3.2 зображене графічне представлення принципу роботи моделі.

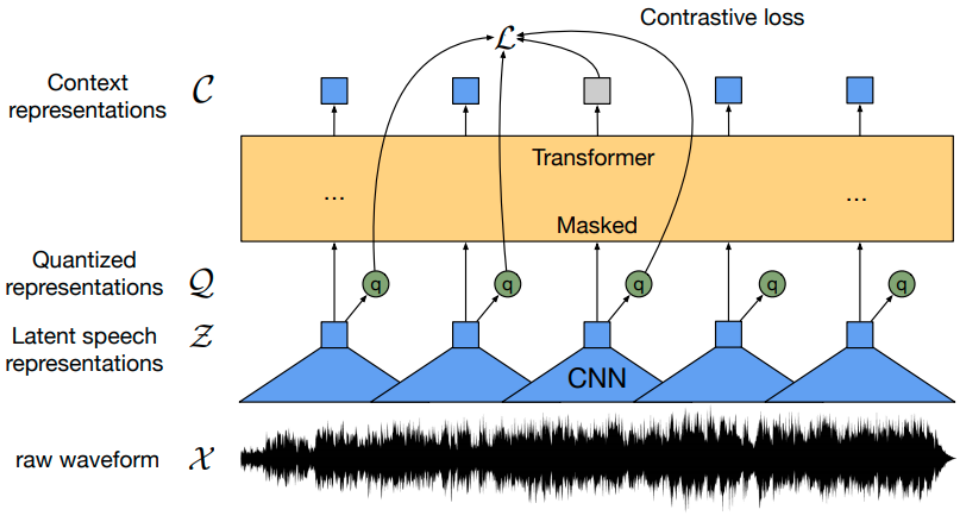

Рисунок 3.2 – Графічне представлення принципу роботи wav2vec2

Для wav2vec створена архітектура, що складається з двох багатошарових згорткових нейронних мереж, накладених одна на одну. Мережа кодера відображає необроблений аудіо вхід у представлення, де кожен вектор охоплює приблизно 30 мілісекунд (мс) мовлення. Контекстна мережа використовує ці вектори для створення власних уявлень, які охоплюють більший проміжок до секунди.

Кількість нейроних шарів в модулі екстаркторі – 7. Код наведений в лістингі 3.1.

Лістинг 3.1 – Приклад нейроного шарів в екстракторі

```
(0): ConvLayerBlock(
        (layer_norm): LayerNorm((512,), eps=1e-05,
```

```
elementwise_affine=True)
        (conv): Conv1d(1, 512, kernel_size=(10,), stride=(5,))
      )
(1): ConvLayerBlock(
        (layer_norm): LayerNorm((512,), eps=1e-05,
elementwise_affine=True)
        (conv): Conv1d(1, 512, kernel_size=(10,), stride=(5,))
      )
```

Кількість нейроних шарів для модуля Енкодера – 48, які ще мають здатність до самоуваги при навчані. Код для EncoderLayer наведений в лістингі 3.2.

Лістинг 3.2 – Приклад нейроного шару для EncoderLayer

```
 (47): EncoderLayer(
          (attention): SelfAttention(
            (k_proj): Linear(in_features=1280,
out_features=1280, bias=True)
            (v_proj): Linear(in_features=1280,
out_features=1280, bias=True)
            (q_proj): Linear(in_features=1280,
out_features=1280, bias=True)
            (out_proj): Linear(in_features=1280,
out_features=1280, bias=True)
          )
```

В якості дата сету для тренування моделі було використано mozilla-foundation/common_voice_7_0. Набір даних Common Voice складається з унікальних wav записів і відповідного текстового файлу. Багато з 13905 записаних годин у наборі даних також містять демографічні метадані, такі як вік, стать та акцент, які можуть допомогти підвищити точність механізмів розпізнавання мовлення.

Набір даних наразі складається з 11192 перевірених годин на 76 мовах, але завжди додається більше голосів і мов. Для моделі, яка має має підтримку української мови було використано додаткову лінгвістичну модель.

Результати оцінки "тесту" Common Voice 7 (WER) без додаткової української лінгвістичної моделі і з нею показано в таблиці 3.1.

Таблиця 3.1 – Результати WER для моделі з підтримкою української мовою

З лінгвістичною моделю	Без лінгвістичної моделі
14.62	21.52

На основі отриманих результатів, які зображені в таблиці 3.1, можна зробити висновок, щодо зменшення показника рівня помилок слів, при використані додаткової української лінгвістичної моделі у випадку розпізнавання аудіофайлів, які записані українською мовою. Що впливає на якість розпізнавання слів з аудіо і потенційно може призвести до отримання більш якісних текстових даних на виході.

3.2.2 Параметри та результати тренування модулю Speech to text

Під час навчання були використані наступні гіперпараметри:

- learning_rate: 5e-05;
- train_batch_size: 8;
- eval_batch_size: 8;
- seed: 42;
- gradient_accumulation_steps: 20;
- total_train_batch_size: 160;
- optimizer: Adam with betas=(0.9,0.999) and epsilon=1e-08;
- lr_scheduler_type: linear;
- lr_scheduler_warmup_steps: 500;
- num_epochs: 100.0;
- mixed_precision_training: Native AMP.

Під час підготовки до тестування даної моделі були проведені додаткові тренування Ukrainian STT model, результати тренувань представлені нижче в таблиці 3.2.

Таблиця 3.2 – Результати тренувань

Training Loss	Epoch	Step	Validation Loss	Wer	Cer
1.2815	7.93	500	0.3536	0.4753	0.1009
1.0869	15.86	1000	0.2317	0.3111	0.0614
0.9984	23.8	1500	0.2022	0.2676	0.0521
0.975	31.74	2000	0.1948	0.2469	0.0487
0.8868	47.61	3000	0.1903	0.2257	0.0439
0.8424	55.55	3500	0.1786	0.2206	0.0423
0.8126	63.49	4000	0.1849	0.2160	0.0416
0.7901	71.42	4500	0.1869	0.2138	0.0413
0.7671	79.36	5000	0.1855	0.2075	0.0394
0.7467	87.3	5500	0.1884	0.2049	0.0389
0.731	95.24	6000	0.1877	0.2060	0.0387

На відміну від звичайних методів зменшення шуму на основі мінімальної середньоквадратичної помилки (MMSE), запропонований контрольований метод покращення мови за допомогою пошуку функції відображення між шумними та чистими мовними сигналами на основі глибоких нейронних мереж (DNN). Щоб мати можливість працювати з широким спектром адитивних шумів у реальних ситуаціях, спочатку було розроблено великий навчальний набір, який охоплює багато можливих

комбінацій мовлення та типів шуму.

Архітектура DNN використовується як функція для нелінійної регресії, щоб забезпечити потужні можливості моделювання. Також дане рішення має кілька методів для покращення системи мовлення на основі DNN, включаючи вирівнювання глобальної дисперсії, щоб пом’якшити проблему надмірного згладжування моделі регресії, а також стратегії навчання з відключенням та усвідомленням шуму для подальшого покращення здатності DNN до узагальнення.

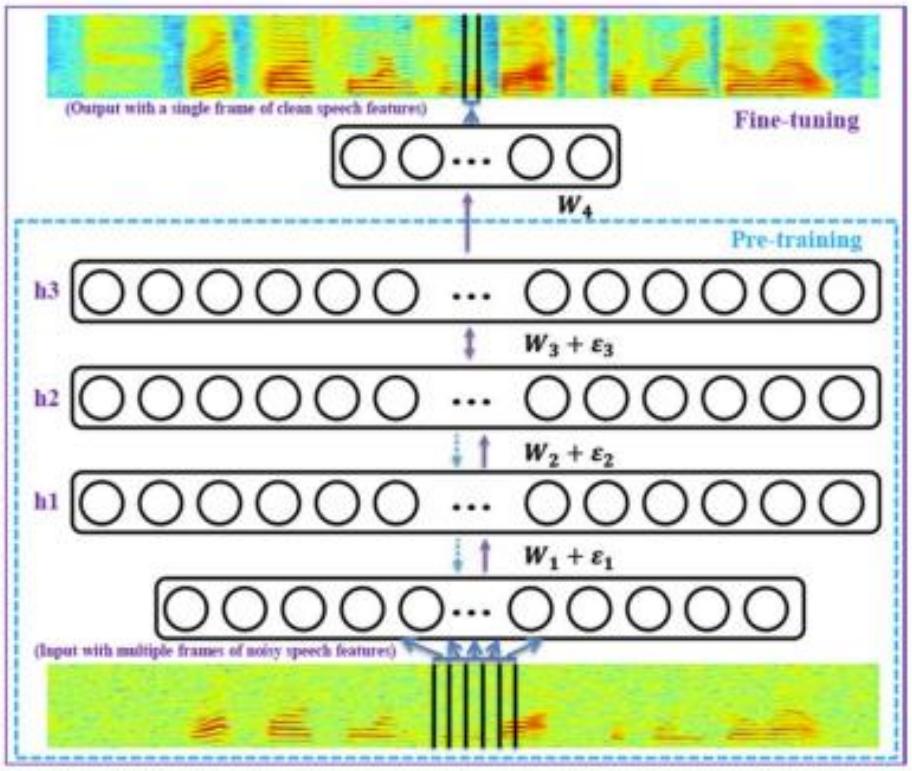

Рисунок 3.3 – Ілюстрація базової процедури навчання DNN

Дана DNN модель спочатку була навчена на 100 годинах шумових мовних даних з ста чотирма типами шуму. Щоб покращити здатність DNN до узагальнення в умовах невідповідності шуму, використовується 3 прихованих шари та 2048 прихованих одиниць для кожного прихованого шару.

3.3 Модуль сумаризації тексту

3.3.1 Модель mT5_multilingual_XLSum

Ранні результати навчання для NLP використовували рекуррентні нейронні мережі, але останнім часом стали більш поширеними використовувати моделі, засновані на архітектурі «трансформера». Спочатку було показано, що Transformer ефективний для машинного перекладу, але згодом він був використаний у широкому діапазоні параметрів NLP.. Основним будівельним блоком Transformer є самоувага. Самоувага – це варіант уваги, який обробляє послідовність, замінюючи кожен елемент на середнє зважене значення решти послідовності. Оригінальний Transformer складався з архітектури кодер-декодер і був призначений для виконання завдань від послідовності до послідовності. Останнім часом також стало звичайним використання моделей, що складаються з одного стека шарів.

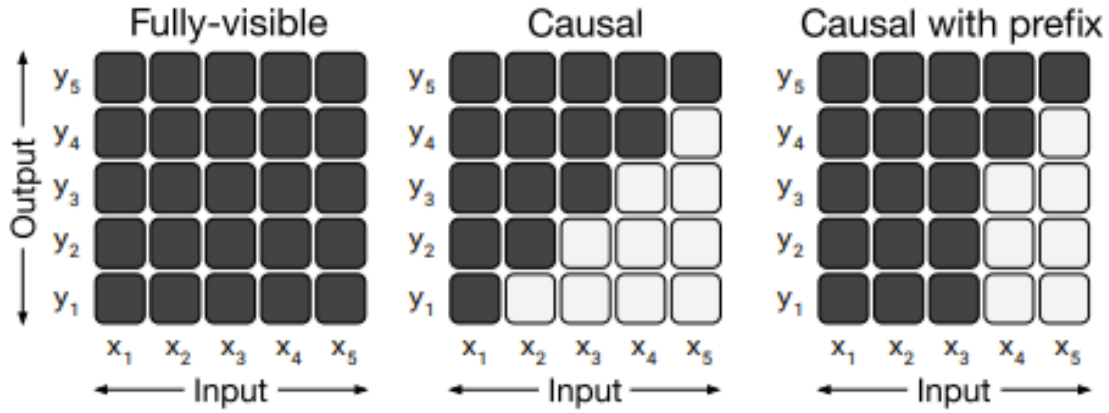

Рисунок 3.4 – Матриці, що представляють різні шаблони масок уваги

Transformer, з різними формами самоуваги, які використовуються для створення архітектур, відповідних для мовного моделювання або завдання щодо класифікації та прогнозування діапазону. Спочатку вхідна послідовність токенів відображається на послідовність вбудов, яка потім

передається в кодер. Кодер складається з стеку «блоків», кожен з яких складається з двох підкомпонентів: шар самоуваги, за яким слідує невелика мережа прямої подачі. Нормалізація шару застосовується до входу кожного підкомпонента. В даному рішені використовується спрощена версію нормалізації шару, де активації лише змінюються, і не застосовується адитивне зміщення. Після нормалізації шару залишкове з'єднання пропуску додає вхідні дані кожного підкомпонента до свого виходу. Dropout застосовується в мережі прямої подачі, на з'єднанні пропуску, на вагових коефіцієнтах уваги, а також на вході та виході всього стека.

Декодер за структурою схожий на кодер, за винятком того, що він включає стандартний механізм уваги після кожного рівня самоуваги, який обслуговує вихід кодера. Механізм самоуваги в декодері також використовує форму авторегресії або причинно-наслідкової самоуваги, що дозволяє моделі враховувати лише минулі результати. Вихід кінцевого блоку декодера подається в щільний шар з виходом softmax, ваги якого спільно з вхідною матрицею вбудовування.

Усі механізми уваги в Transformer розділені на незалежні «голови», вихідні дані яких об'єднуються перед подальшою обробкою. Оскільки самоувага не залежить від порядку (тобто це операція над множинами), звичайним є надання явного сигналу положення до Трансформатора. У той час як оригінальний Transformer використовував синусоїдний сигнал позиції або вбудоване положення, останнім часом стало більш поширеним використання вбудовування відносного положення. Замість використання фіксованого вбудовування для кожної позиції, вбудовування відносних позицій створюють відповідно до зміщення між «ключом» і «запитом», які порівнюються в механізмі самоуваги. Кожне «вбудовування» є просто скаляром, який додається до відповідного логіта, який використовується для обчислення ваг уваги.

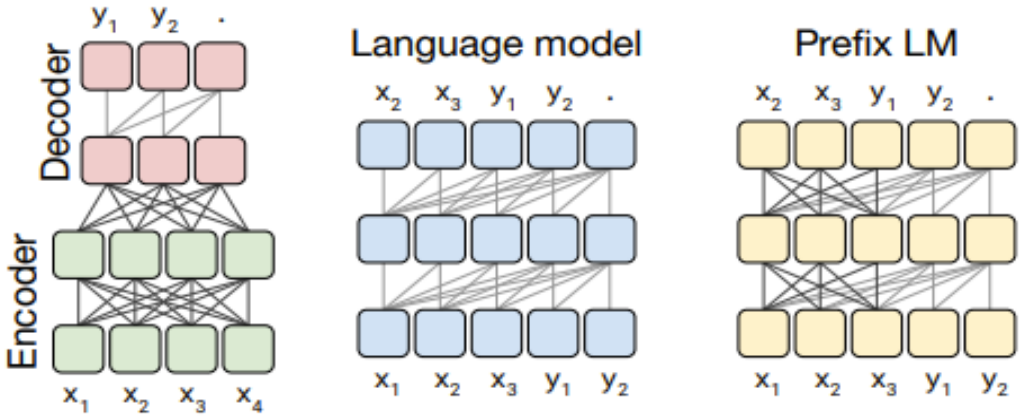

Рисунок 3.5 – Архітектура Transformer

Для ефективності поділяються параметри вбудовування позиції для всіх шарів моделі, хоча в межах даного шару кожна голова уваги використовує інше впроваджене положення. Як правило, вивчається фіксована кількість вкладень, кожне з яких відповідає діапазону можливих зміщень ключового запиту. У цьому рішені використовується 32 вбудовування для всіх наших підмоделей з діапазонами, розмір яких логарифмічно збільшується до зміщення 128, за межами якого призначаються всі відносні позиції тому самому вбудованню.

Даний рівень нечутливий до відносного положення за межами 128 маркерів, але наступні шари можуть побудувати чутливість до більших зміщень, комбінуючи локальну інформацію з попередніх шарів. Модель mT5 – попередньо навчений мультилінгвальний Transformer для 101 мов. mT5 є розширенням моделі Text-to-Text Transfer Transformer (T5).

Дане рішення було навчене на корпусі веб-сторінок з Common Crawl 101 мовами – mC4. Архітектура моделі та процедура навчання mT5 схожа з архітектурою та навчанням класичної T5 моделі. mT5 ґрунтується на версії “T5.1.1” моделі T5, в якій використовують GeGLU нелінійність та попередньо навчають на нерозмічених даних без використання dropout. Дані різними мовами семплювали так, щоб можна було регулювати баланс між рідкісними та популярними мовами веб-сторінок. Для цього обчислювали

ймовірність семплінг тексту певною мовою. На даной моделю проведений fine-tuning за допомогою XL-SUM датасету.

Даний датасет має різноманітний набір даних, що містить 1,35 мільйона професійно анотованих пар стаття-резюме від BBC, витягнутих за допомогою набору ретельно розроблених евристик. Набір даних охоплює 45 мов, від низьких до високоресурсних, для багатьох із яких наразі немає загальнодоступного набору даних. XL-Sum дуже абстрактний, стислий і високоякісний, як вказує людська і внутрішня оцінка.

В даному дипломному проекті розлянуті варіанти для української та англійської мови. В таблиці 3.3 можна ознайомитися з результатами бенчмаркінгу оцінки тестових наборів XL-SUM за метрикою ROUGE.

Таблиця 3.3 – Бенчмаркінг тестових наборів XL-SUM

Мова	ROUGE-1	ROUGE-2	ROUGE-3
Англійська	37.601	15.1536	29.8817
Українська	23.9908	10.1431	20.9199

3.3.2 Підмодуль роботи з маркерами тексту

Основна ідея роботи цього підмодуля полягає в розбиті вхідного тексту на частинки. Тобто відбувається створення окремого json файлу з “нарізаним” текстом між двома словами маркерами. Даний функціонал створює можливість реферувати окремі частини тексту не змішуючи не пов’язану між собою інформацію, яка може призвести до втрали сенсу інформації. Після етапу реферування відбувається “склеювання” отриманих даних в один документ, наступного виду: “Слово маркер: контент”. Згідно проведених досліджень дане покращення направлене на підвищення логічного зв’язку вхідної інформації між собою.

На рисунку 3.7 зображена схема роботи даного рішення.

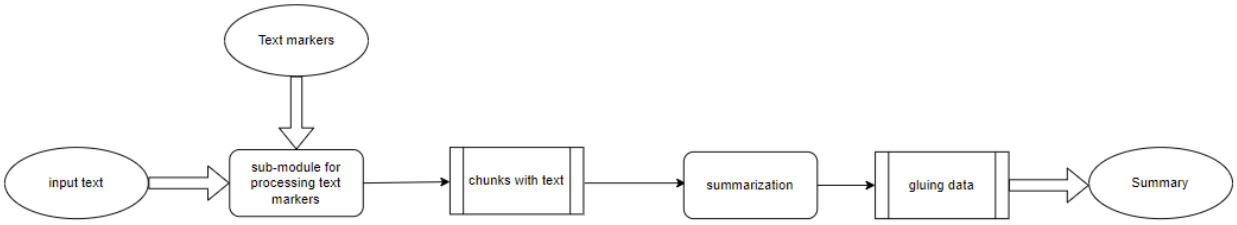

Рисунок 3.7 – Схема роботи підмодулю, який опрацьовує маркери

Опис роботи схеми:

1) На вхід подається текст і текстові маркери.
2) Відбувається парсинг і нарізання тексту на “шматки”.
3) Запуск модуля Speech to text.
4) Склеювання даних до єдиного документу.

3.4 Використання хмарних рішень

Під час розробки комплексу була розроблена модель використання даного рішення в хмарних сервісах. Перенесення даного рішення в хмару дозволить запобігти втрати даних під час непередбачуваних ситуацій, серед яких слід виділити військові конфлікти, природні або техногенні катастрофи. А також, що не менш важливо, дозволить використовувати більш потужні обчислювальні можливості.

Модель хмарного рішення була розроблена для використання Amazon Web Services, так як їх рішення є лідерами на ринку і забезпечують надійність роботи з даними. Також слід відмітити можливість до моніторингу використання ресурсів і автоматичного збільшення потужності сервера в залежності від заданого алгоритму реакцій на події.

В даному рішені використані наступні технології AWS:

- S3 bucket;
- AWS Lambda;

- Amazon EC2;
- S3 Glacier.

Amazon S3 або Amazon Simple Storage Service – це служба, яку пропонує Amazon Web Services (AWS), яка забезпечує зберігання об’єктів через інтерфейс веб-служби. Amazon S3 використовує ту саму масштабовану інфраструктуру зберігання, що й Amazon.com для роботи своєї мережі електронної комерції. Amazon S3 може зберігати будь-який тип об’єктів, що дозволяє використовувати, як-от сховище для Інтернет-додатків, резервне копіювання, аварійне відновлення, архіви даних, озера даних для аналітики та гібридне хмарне сховище.

Lambda – це обчислювальний сервіс, який дозволяє запускати код без надання або керування серверами. Lambda запускає код на високодоступній обчислювальній інфраструктурі та виконує все адміністрування обчислювальних ресурсів, включаючи обслуговування сервера та операційної системи, надання потужності та автоматичне масштабування, моніторинг коду та ведення журналів. За допомогою Lambda існує можливість запускати код практично для будь-якого типу програми або серверної служби.

Amazon Elastic Compute Cloud (Amazon EC2) – є одним із сервісів Amazon Web Services, що дозволяє користувачеві орендувати віртуальний сервер, які називаються інстанс. Для запуску віртуальних серверів використовуються попередньо налаштовані образи, що скорочує час завантаження нового сервера. EC2 пропонує найширшу та найглибшу обчислювальну платформу з більш ніж 500 інстансами та набором новітніх процесорів, сховищ, мереж, операційних систем та моделей покупок, забезпечуючи належну відповідність потреб конкретного робочого навантаження. На AWS виконується більше робочих навантажень SAP, високопродуктивних обчислень (HPC), машинного навчання та Windows, ніж у будь-якій іншій хмарі.

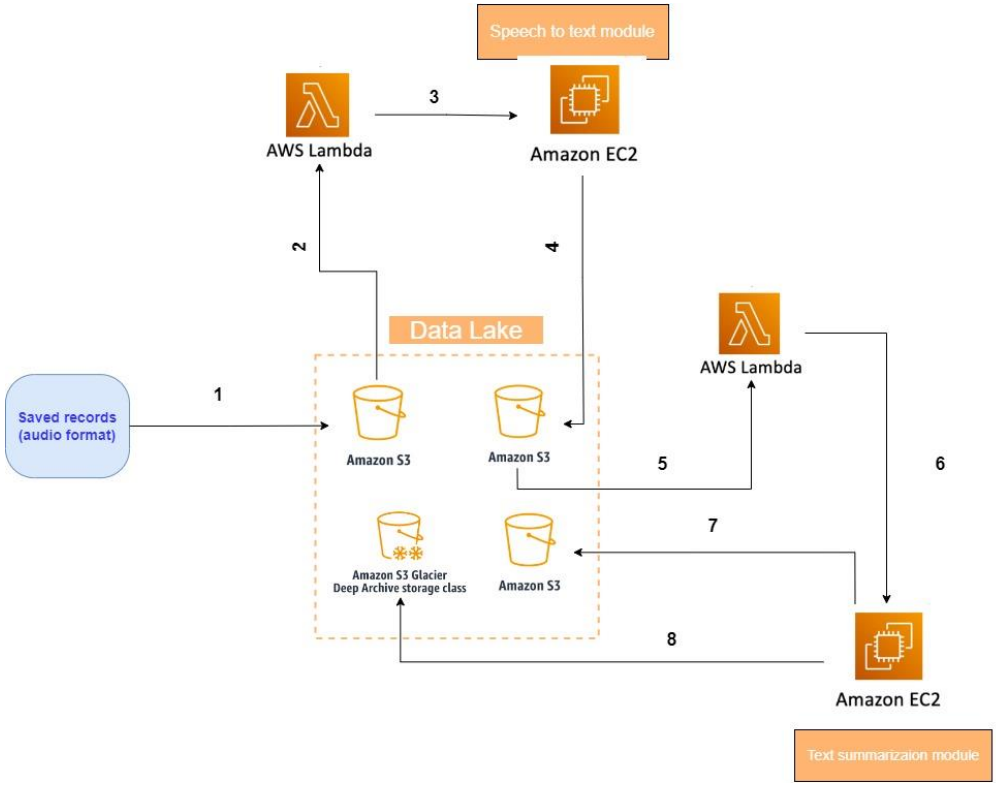

Рисунок 3.7 – Схема для AWS

Класи сховища Amazon S3 Glacier спеціально створені для архівування даних, забезпечуючи найвищу продуктивність, гнучкість пошуку та найнижчу вартість архівного сховища в хмарі. Усі класи зберігання даних S3 Glacier забезпечують практично необмежену масштабованість і розраховані на 99,999999999% довговічності даних. Класи сховища S3 Glacier пропонують варіанти якнайшвидшого доступу до архівних даних і найдешевшого архівного сховища в хмарі.

Опис роботи схеми:

1) завантаження аудіофайлів в сегмент S3;
2) виклик лямбда для запуску екземпляра з модулем Speech to text;
3) запуск модуля Speech to text;
4) збереження даних у S3;
5) виклик лямбда для запуску екземпляра модуля сумаризації тексту;
6) запуск модуля сумаризації тексту;
7) збереження даних у S3 Glacier (архівування даних).

4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Дослідження проводилося на локальному компьютері, а також на відаленому хмарному рішені з характеристиками, які зазначені у таблиці 4.1.

Таблиця 4.1 – Використане апаратне забезпечення для тестування системи

CPU	Intel Core i7-9750H (2.6 - 4.5 ГГц)	Intel Xeon 2.30GHz
GPU	NVIDIA GeForce GTX 1650 Mobile	NVIDIA Tesla T4
GPU (характеристики)	896 ядра CUDA, 1665 МГц, 4 ГБ DDR5	2560 ядра CUDA, 320 ядра Tensor, 1250 МГц, 16 ГБ DDR6
RAM	16 Gb	12 Gb
OS	Linux (Ubuntu 20.04)	Linux

Час, необхідний для виконання завдання на одному процесорі, порівняно з часом, необхідним для виконання тієї ж діяльності на паралельних процесорах, зазвичай використовується для визначення потенційної вигоди від паралельних обчислень.

$$Speedup = \frac{T_s}{T_p} \tag{4.1}$$

де T_s – час виконання послідовного алгоритму;

T_p – час виконання паралельного алгоритму.

4.1 Оцінка роботи під модуля очищення звукового ряду

Результати експерименту демонструють, що запропонована структура

може досягти значних покращень як в об’єктивних, так і в суб’єктивних показниках порівняно зі звичайною технікою на основі MMSE. Також цікаво помітити, що запропонований підхід DNN може добре вилучити дуже нестаціонарний шум, з яким загалом важко впоратися. Крім того, отримана модель DNN, навчена штучно синтезованими даними, також ефективна для роботи з шумними мовними даними, записаними в реальних сценаріях, без створення дратівливого музичного артефакту, який зазвичай спостерігається у звичайних методах покращення. В таблиці 4.2 можна побачити порівняння за метрикою WER, щодо ефективності роботи за округленими в більшу сторону середніми показниками з трьох аудіодоріжок формату wav.

Таблиця 4.2– Ефективність очищення від стороннього шуму

Мова	Українська	Англійська
Аудіодоріжка очищена від шуму, WER	29	3.9
Аудіодоріжка не очищена від шуму, WER	38.7	5.2
Різниця, %	25%	21%

У випадку використання української мови дана різниця є важливою, так як WER, що знаходиться в нормальному діапазоні при розпізнані мовлення з лекції є 20-30.

4.2 Оцінка роботи модуля Speech to text

Було проаналізовано 3 аудіозаписи англійською та українською мовами з різною довжиною. Класифікація записів: короткий (5с), середній(~ 60с), довгий (~ 240с). Результати витраченого часу на обробку аудіосигналу українською і англійською мовами, а також наведено у таблиці нижче.

Таблиця 4.3 – Результати витрати часу на обробку аудіо

Довжина аудіозапису	Час витрачений на персональному комп’ютері (українська), сек	Час витрачений на хмарному рішенні (українська), сек	Час витрачений на персональному комп’ютері (англійська), сек	Час витрачений на хмарному рішенні (англійська), сек
Короткий	0.8	0.6	0.6	0.48
Середній	10.8	3.6	9.4	3.14
Довгий	222.7	84.2	217.1	70.1

Рисунок 4.1 – Гістограма витраченого часу

Згідно результатам, зображених на гістограмі (рисунок 4.1), видно тенденцію прискорення для більш потужної відеокарти. Розпаралелений алгоритм на декілька потоків має перевагу при більшому часі аудіозапису.

Також проведений замір метрики рівня помилок слів (WER) для очищеного звукового ряду, який для обох відеокарт є однаковим, але для мов – різним. З результатами замірів, можна ознайомитися в таблиці 4.4

Таблиця 4.4 – Результати заміру WER

Довжина аудіозапису	Українська мова, WER	Англійська мова, WER
Короткий	24.6	3.4
Середній	28.4	3.7
Довгий	33.9	4.5

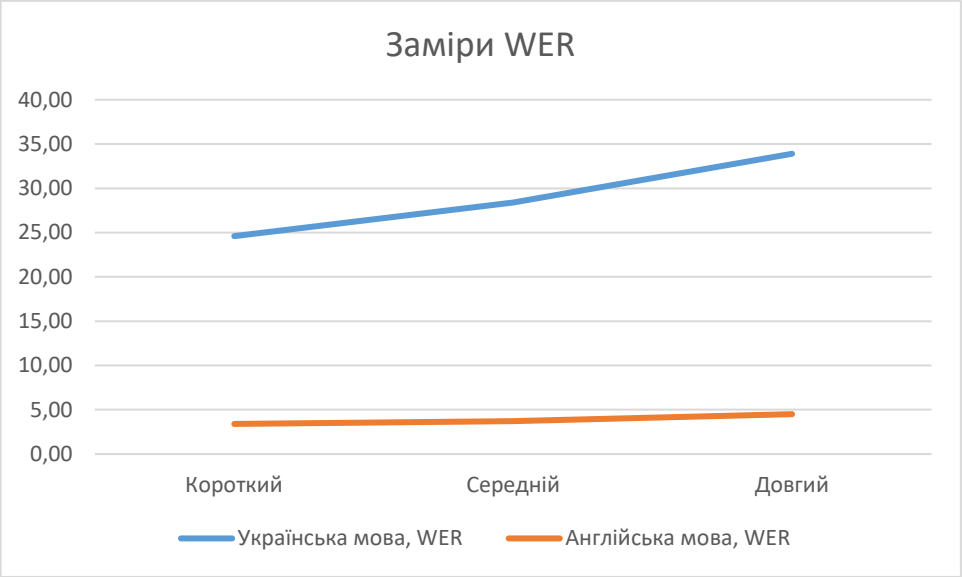

Рисунок 4.2 – Заміри WER метрики

Згідно результатам, зображених на гістограмі (рисунок 4.2), видно тенденцію збільшення рівня помилок в словах при для більшій кількості даних. Це може бути пов’язано з недостатньо навченою моделю для української мови модуля і призвести до втрати логічного сенсу запису. Нажче наведено приклад роботи даного.

Приклад 1 (українська мова):

“серце чи дивовижний порятунок мільйони людей фактично в прямому ефірі вже три доби спостерігають за спробами мароканських рятувальників дісттисколодя за пятирічне хлопя досі незрозуміло чи вдастся дістати його з тридцятиметрового провалля живим про надзвичайно складну операцію що триває в цю мить я не слесарчукодязв який провалився пятирічний раян ледь

помітна діра в землі менше тридцяти сантиметрів у діаметрі але в глиб вона тягнеться на тридцять два метри батьки шукали сина кілька годин перед тим як зрозуміли він під землею а коли він зник я молилася богу просила аби алаксбирігмо сина і його дістали сководізяживим господи хай йому там менше болить в тій тіріятак сподіваючись що у рятувальники все вийде його неможливо витягти просто так розуміють рятувальників занадто вузько а розширяти діру не можна вона просто завалиться тому вони три добою розкопують амнокдалік і поки працює технік”

Приклад 2 (англійська мова):

“before he had time to answer a much encumbered vera burst into the room with the question i say can i leave these here these were a small black pig and a lusty specimen of black red game cock”

4.3 Оцінка роботи модуля Summarization

Даний модуль показав досить високі результати для задачі реферування тексту, для української та англійської мов. Даний підхід до сумаризації використовує абстрактивний метод з використанням розпаралелених обчислень.

Результати порівняння вхідного і вихідного тексту зображено в таблиці 4.5. Для порівняння використовується текст отриманий з модуля Speech to text.

Таблиця 4.5 – Порівняння зжаття тексту

Мова	Початкова кількість слів	Кінцева кількість символів	Зжаття тексту, %
Українська	136	14	89.7
Англійська	188	11	94.15

Нижче наведені приклади по яких порівнюється оцінка роботи модуля сумаризації в таблиці вище.

Приклад 1 (українська мова):

Вхідний текст – “серце чи дивовижний порятунок мільйони людей фактично в прямому ефірі вже три доби спостерігають за спробами мароканських рятувальників дісттисколодя за пятирічне хлопя досі незрозуміло чи вдастся дістати його з тридцятиметрового провалля живим про надзвичайно складну операцію що триває в цю мить я не слесарчукодязв який провалився пятирічний раян ледь помітна діра в землі менше тридцяти сантиметрів у діаметрі але в глиб вона тягнеться на тридцять два метри батьки шукали сина кілька годин перед тим як зрозуміли він під землею а коли він зник я молилася богу просила аби алаксбирігмо сина і його дістали сководізяживим господи хай йому там менше болить в тій тіріятак сподіваючись що у рятувальники все вийде його неможливо витягти просто так розуміють рятувальників занадто вузько а розширяти діру не можна вона просто завалиться тому вони три добою розкопують амнокдалік і поки працює технік”

Сумаризований текст – “Вже три доби спостерігають за спробами мароканських рятувальників дісттисколодя за п'ятирічне хлопя з провалля.”

Приклад 2 (англійська мова):

Вхідний текст – “videos that say approved vaccines are dangerous and cause autism cancer or infertility are among those that will be taken down the company said the policy includes the termination of accounts of antivaccine influencers tech giants have been criticised for not doing more to counter false health information on their sites in july us president joe biden said social media platforms were largely responsible for people scepticism in getting vaccinated by spreading misinformation and appealed for them to address the issue youtube which is owned by google said videos were removed from its platform since last year when it implemented a ban on content spreading misinformation about covid vaccines in a blog post the company said it had seen false claims about covid jabs spill over into misinformation about vaccines in general the new policy covers longapproved vaccines such as those against measles or hepatitis b we're

expanding our medical misinformation policies on youtube with new guidelines on currently administered vaccines that are approved and confirmed to be safe and effective by local health authorities and the who the post said referring to the world health organization.”

Сумаризований текст – “YouTube has banned thousands of videos spreading misinformation about Covid vaccines.”

Загалом потрібно відзначити досить високу якість реферування тексту навіть українською мовою. Слід звернути увагу, що дослідження NLP напрямку для сумаризації тексту проводяться відносно давно для багатьох мов, але лідером по високим показаникам є англійська мова.

Нижче в таблиці 4.6 зображено витрачений час даним модулем для обробки вхідних даних українською мовою.

Таблиця 4.6 – Порівняння часу витраченого на сумаризацію тексту українською мовою

Кількість слів	Час витрачений на персональному комп’ютері (українська), сек	Час витрачений на хмарному рішенні (українська), сек
25	0.4	0.37
136	3.2	0.92
725	13.6	5.13

Згідно результатам, зображених на точковій діаграмі (рисунок 4.3), видно тенденцію збільшення часу обробки даних, що і є очікуваним результатом. Також на даній діаграмі можна помітити зменшення часу обробки даних для кращої відеокарти.

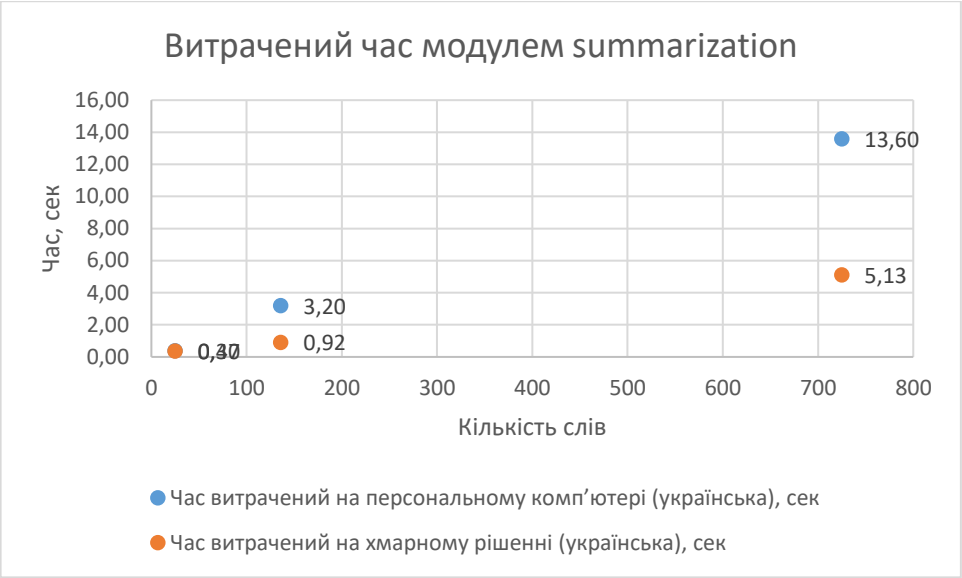

Рисунок 4.3 – Діаграма витраченого часу

Також були проведені заміри для англійського тексту. Результати зображені в таблиці 4.7

Таблиця 4.7 – Порівняння часу витраченого на сумаризацію тексту англійською мовою

Кількість слів	Час витрачений на персональному комп'ютері (англійська), сек	Час витрачений на хмарному рішенні (англійська), сек
32	0.38	0.34
187	3.3	0.94
713	12.7	4.98

Згідно отриманих результатів було побудовано діаграму, на якій можна помітити зменшення часу обробки даних відносно української мови. Так само, як і у випадку наведеному вище з використанням більш потужної відеокарти можна побачити значний приріст обробки даних для більш великого текстового контенту. З діаграмою можна ознайомитися на рисунку 4.4.

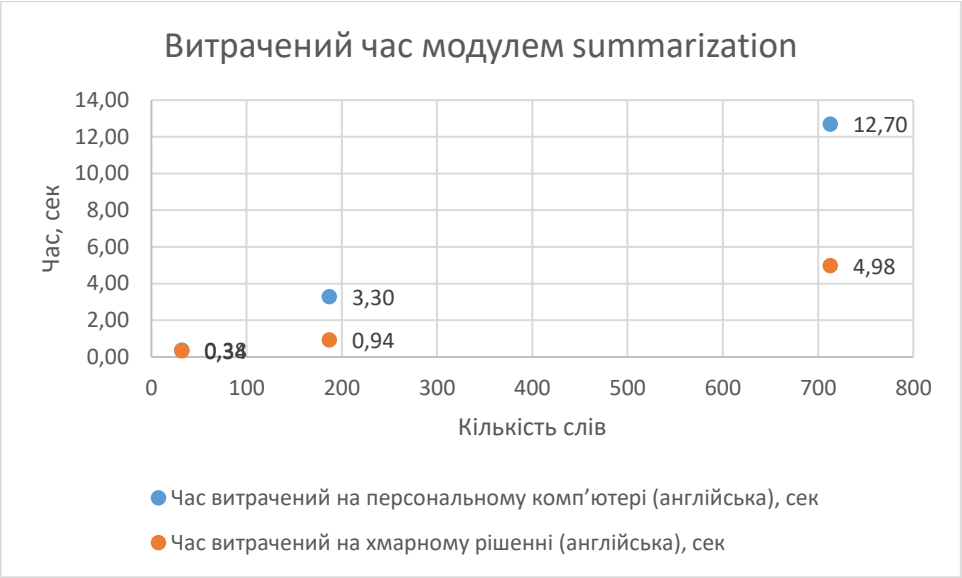

Рисунок 4.4 – Діаграма витрати часу для обробки тексту англійською мовою

Отже можна зробити висновок щодо ефективності пришвидшення роботи обробки даних в цьому модулі з більш потужною відекартою. Для української мови відбувається значне пришвидшення з більшою кількістю вхідних даних, а саме якщо взяти до уваги результати тестування 136 і 725 вхідних слів, середнє прискорення буде близько 34%. Виходячи з результатів щодо 725 вхідних слів, чим більше вхідних слів – тим вище прискорення від більш потужної відеокарти.

Щодо обробки тексту англійською мовою, результат є трішки швидшим і так само спостерігається прискорення, для більш потужної відеокарти. Середній результ прискорення для вхідних 187 і 713 є так само 34%.

4.3 Оцінка роботи підмодуля текстових маркерів

Згідно проведених вище досліджень, розроблене рішення компенсує проблему втрати контексту документу, а також з використанням паралельних обчислень критично не навантажує систему. Однак створює обмеження в зв'язку з необхідністю доповідачу вживати слова маркери. З результатами роботи даного рішення можна ознайомитися в прикладах наведених нижче.

Приклад 1 (англійська мова):

Слова маркери: “Introduction, introdaction, INTRODUCTION, Conclusion, conclusion, CONCLUSION”.

Текст отриманий при роботі модуля speech-to-text – “introdaction the period of the pandemic has significantly creased the relevance of the development and expansion of the functionality of various digital educational platforms these information spaces are in demand at different levels of education in countries around the world from primary schools to higher education institutions well as training courses in different areas of business allowing students to be provided with teaching materials communicating with teachers and also providing opportunity for remote control of the level of knowledge this approach is not a new one that is why now we have not the breakthrough but the further fast development this the virtual interaction of the distributed user community is organizethe number of interactive possibilities of digital educational platforms is constantly expanding new options are added to the basic functionality access to training materials on cloud storage services remote interaction with teacher for example testing block knowledge control access level management of training content depending on the users role analysis of statistical data attendance and some other special functionality like it conclusion this paper discussed the main applications of audio analytics such speech recognition and speech text transformation many of them use noise reduction as a basis for improving the signal level which was demonstrated in the paper as a proposed speechtext module for converting a verbal lecture or a lesson into written text form digital educational platforms the main methods and algorithms of noise reduction were considered with the presentation of examples of processing results their computational complexity and the possibility of parallelization”

Сумаризований текст з використанням слів маркерів після етапу склейки – “Introdaction: The period of the pandemic has significantly creased the relevance of the development and expansion of various digital educational

platforms. These information spaces are in demand at different levels of education in countries around the world from primary schools to higher education institutions. The number of interactive possibilities of digital educational platforms is constantly expanding new options are added. Conclusion: The main methods and algorithms of noise reduction were considered with the presentation of examples of processing results their complexity and the possibility of parallelization. A proposed speechtext module for converting a verbal lecture or a lesson into written text form digital educational platforms was presented as an example of how this could be achieved."

Приклад 2 (українська мова)

Слова маркери: "Вступ, вступ, ВСТУП, Висновок, виснок, ВИСНОВОК".

Текст отриманий при роботі модуля speech-to-text – "вступ військовослужбовці окрема мотострілковой бригади імені кошового атамана костя гордієнка показала як українські бійці знищують планктонні переправа через річку сіверський донецьк у луганській області українська розвідка повідомляє що росія вже вичерпала резервна боєздатна батальйонних тактичних груп за рахунок прихованої мобілізації білгородський та ростовській області з представників і мобілізаційних резервів формують нові підрозділи який направляють в україну втім ефективність яких підрозділів низка зазначають у розвідці висновок російські найманці які приїхали вбивати українці на донбасі відмовилися воювати їхні відеозвернення опубліковано журналіст денис казанський дата зйомки невідома окупанти відкривається що коли прибули до луганської області місцева командування використання їх не за призначенням та відправило на забій наразі вони повернулися до росії але підтримку з боку своєї держави також не отримали"

Сумаризований текст з використанням слів маркерів після етапу склейки – "Вступ: Українські військовослужбовці демонструють переправу через річку сіверський донецьк у луганській області. Українська розвідка повідомляє що росія вже вичерпала свої резервні батальйонні тактичні групи

але в білгороді та ростовсько-ростовській області таємно мобілізують нові підрозділи. Висновок: Російські найманці які прийшли вбивати українців на донбасі відмовилися воювати. Тепер вони повернулися до росії але не отримали підтримки від своєї держави. Окупанти виявили що прибувши на луганщину місцеве командування використовувало їх не за призначенням.”

ВИСНОВКИ

Розпізнавання мовлення являється придатним і відносно надійним інструментом для розпізнавання голосу з аудіофайлів. Але для того, щоб підвищити якість і точність потрібна попередня обробка аудіозапису, а також натренована модель на лінгвістичному корпусі.

Сумаризація тексту являється хорошим інструментом для узагальнення інформації. Але має обмеження щодо втрати контексту, якщо реферується великий текст. Запропонована ідея використання слів маркерів направлена на часткове вирішення цієї проблеми.

В процесі виконання кваліфікаційної роботи було проведено дослідження параметрів, які повинна задовольняти система розпізнавання голосу та система сумаризації тексту. Також були розглянуті існуючі методи обробки NLP у випадку з перетворенням мовлення в текст і його сумаризації.

На підставі проведеного дослідження був запропонований та розроблений комплекс обробки вхідних аудіозаписів та їх сумаризації. В процесі розробки системи було проведено порівняння розпізнавання мовлення англійською і українською мовами за допомогою використання моделі, яка розроблена з використанням згорткової нейроної мережі. А також проведено сумаризації текстів на вище вказаних мовах, з використанням новітньої моделі від компанії Google.

В ході дослідження були проведені оцінка таймінгу роботи розробленої системи та аналіз отриманих результатів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Barkovska, Olesia, Viktor Khomych, and Oleksandr Nastenko. "RESEARCH OF THE TEXT PROCESSING METHODS IN ORGANIZATION OF ELECTRONIC STORAGES OF INFORMATION OBJECTS." Innovative Technologies and Scientific Solutions for Industries 1 (19) (2022).

2. Juang, Biing-Hwang, and Lawrence R. Rabiner. "Automatic speech recognition–a brief history of the technology development." Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara 1 (2005): 67.

3. Wang, Dong, Xiaodong Wang, and Shaohe Lv. "An overview of end-to-end automatic speech recognition." Symmetry 11.8 (2019): 1018.

4. Ramezani, Majid, and Mohammad-Reza Feizi-Derakhshi. "Automated text summarization: An overview." Applied Artificial Intelligence 28.2 (2014): 178-215.

5. Allahyari, Mehdi, et al. "Text summarization techniques: a brief survey." arXiv preprint arXiv:1707.02268 (2017).

6. Villamizar, Mario, et al. "Infrastructure cost comparison of running web applications in the cloud using AWS lambda and monolithic and microservice architectures." 2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). IEEE, 2016.

7. Ding Jr, Ing, and Yen-Ming Hsu. "An HMM-like dynamic time warping scheme for automatic speech recognition." Mathematical Problems in Engineering 2014 (2014).

8. Wielgat, Robert. "Improving speech recognition accuracy using HMM and DTW methods." w Proc. ICSES. 2004.

9. Chuctaya, Hernan Faustino Chacca, Rolfy Nixon Montufar Mercado, and Jeyson Jesus Gonzales Gaona. "Isolated automatic speech recognition of Quechua numbers using MFCC, DTW and KNN." Int. J. Adv. Comput. Sci. Appl 9.10 (2018): 24-29.

10. P. G. N. Priyadarshani, N. G. J. Dias, and A. Punchihewa,“Dynamic time warping based speech recognition for isolatedSinhala words,” in Proceedings of the 55th IEEE InternationalMidwest Symposium on Circuits and Systems (MWSCAS ’12),pp.892–895, August 2012.

11. Singh, Charu, et al. "A real-time DSP-based system for voice activity detection and background noise reduction." *Intelligent Speech Signal Processing*. Academic Press, 2019. 39-54.

12. J. Bergstra and Y. Bengio. "Random search for hyper-parameter optimization. Journal of Machine Learning Research", 13:281–305, 2012.

13. Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013.

14. Kim, S.; Hori, T.; Watanabe, S. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 4835–4839

15. Graves, A.; Jaitly, N. Towards end-to-end speech recognition with recurrent neural networks. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21 June–26 June 2014; pp. 1764–1772.

16. W. al-sanie, “Towards an infrastructure for Arabic text summarization using rhetorical structure theory”, M.S Thesis”, Department of computer science. King Saud university, Riyadh, Kingdom of Saudi Arabia, 2005.

17. H. Luhn, “The automatic creation of literature abstraction”, IBM Journal of research and development, Vol 2, pp.159-165, 1958.

18. G.O. Makbule, “Text Summarization using Latent Semantic Analysis”, M.S thesis, Middle East Technical University, 2011.

19. A. Agrawal, U. Gupta, “Extraction based approach for text summarization using k-means clustering”, International Journal of Scientific and Research Publications, Vol 4, Issue 11, 2014.

20. Барковська О. Ю., Литвиненко В. С. Дослідження продуктивності

нейромережевих моделей при семантичному аналізі / Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління. Тези доповідей 12-ої МНТК (online-режим). – 2022. – Т.1: секція 3. – с.136.

21. Barkovska, Olesia, Vladyslav Kholiev, and Vladyslav Lytvynenko. "STUDY OF NOISE REDUCTION METHODS IN THE SOUND SEQUENCE WHEN SOLVING THE SPEECH-TO-TEXT PROBLEM." Advanced Information Systems 6.1 (2022): 48-54.