

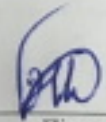
## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

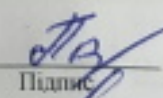
на тему Метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами

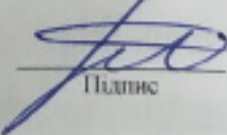
Галузь знань 12 – Інформаційні технології  
Шифр і назва галузі знань

Спеціальність 122 – Комп'ютерні науки  
Шифр і назва спеціальності

Освітня програма Комп'ютерні науки  
Назва освітньої програми

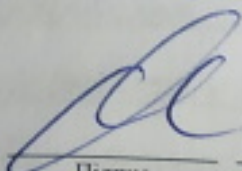
Виконав: студент 4 курсу, група КН-19-1  Д.О. Жайворон  
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: к.п.н., доцент кафедри КН  С.С.Петровський  
Науковий ступінь, посада Підпис Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КН  Р.О. Багрій  
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор

 О.В. Бармак  
Підпис Ініціали, прізвище

05 06 2023 р.

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

Освітня програма освітньо-професійна програма підготовки бакалавра

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

(підпис)

д.т.н., професор О.В. Бармак

«06» 03 2023 року

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами»

2. Завдання видано студенту Жайворону Дмитру Олеговичу  
(прізвище, ім'я, по батькові)

3. Керівник роботи доцент кафедри КН Петровський Сергій Степанович  
(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від «01» 03 2023р. № 5

5. Дата видачі завдання студенту: «03» 03 2023р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Провести аналіз предметної області й існуючих підходів до автоматизованого стенографування аудіозаписів, визначити особливості застосування нейромереж для задач автоматизованого стенографування аудіозаписів.

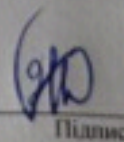
Провести аналіз підходів до класифікації текстів за ключовими словами.

Розробити метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами. Розробити архітектуру нейронної мережі для стенографування аудіозаписів. Спроектувати структуру інформаційної системи класифікації аудіозаписів. Спроектувати та створити інформаційну систему що використовує розроблений метод, виконати її тестування.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником	грудень 2022	Виконав
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	січень 2023	Виконав
3	Робота над розділом 1 – Характеристика предметної області та постановка задачі	січень 2023	Виконав
4	Робота над розділом 2 – Метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами	березень 2023	Виконав
5	Робота над розділом 3 – Програмна реалізація інформаційної системи класифікації аудіозаписів	квітень 2023	Виконав
6	Оформлення пояснювальної записки згідно вимог	травень 2023	Виконав
7	Попередній захист кваліфікаційної роботи бакалавра	травень 2023	Виконав
8	Захист кваліфікаційної роботи бакалавра на засіданні Екзаменаційної комісії	червень 2023	Виконав

Виконавець: студент 4 курсу, група КН-19-1  
Курс, група виконання

  
Підпис

Д.О. Жайворон  
Ініціали, прізвище

Керівник: к.п.н., доцент кафедри КН  
Науковий ступінь, посада

  
Підпис

С.С.Петровський  
Ініціали, прізвище

## Анотація

Тема кваліфікаційної роботи бакалавра: Метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-19-1 Жайворон Дмитро Олегович

Керівник кваліфікаційної роботи бакалавра: к.п.н., доцент кафедри КН Петровський Сергій Степанович

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
62	19	0	27	3

Метою кваліфікаційної роботи бакалавра є розробка та програмна реалізація методу нейромережевого стенографування для класифікації аудіозаписів за ключовими словами.

Результатом виконання кваліфікаційної роботи бакалавра є створений метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами та інформаційна система класифікації аудіозаписів, яка використовує розроблений метод й дозволяє за аудіофайлом автоматизовано визначати його семантичну категорію за допомогою нейронної мережі.

Ключові слова: класифікація аудіозаписів, стенографування, нейронна мережа, аудіозапис, нейромережа, система класифікації аудіозаписів.

Виконавець: студент 4 курсу, група КН-19-1

Курс, група виконавця



Підпис

Д.О.Жайворон

Ініціали, прізвище

## Зміст

Перелік скорочень .....	4
Вступ.....	5
Розділ 1 Характеристика предметної області і постановка задачі.....	7
1.1 Аналіз підходів до автоматизованого стенографування аудіозаписів .....	7
1.2 Особливості застосування нейромереж для задач автоматизованого стенографування аудіозаписів .....	10
1.3 Аналіз існуючих рішень для задач автоматизованого стенографування аудіозаписів.....	12
1.4 Аналіз підходів до класифікації текстів за ключовими словами .....	14
1.5 Аналіз існуючих рішень для задач класифікації текстів .....	16
1.6 Мета, задачі та вимоги до реалізації програмної системи .....	19
Розділ 2 Метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами.....	20
2.1 Опис і етапи виконання методу нейромережевого стенографування для класифікації аудіозаписів за ключовими словами.....	20
2.2 Розробка архітектури нейронної мережі для стенографування аудіозаписів .....	22
2.3 Підготовка робочих даних для системи .....	26
2.4 Проектування структури інформаційної системи класифікації аудіозаписів .....	28
2.5 Даталогічна модель бази даних інформаційної системи класифікації аудіозаписів.....	32
2.6 Вибір засобів розробки інформаційної системи класифікації аудіозаписів .....	35
2.7 Висновки до розділу 2 .....	39
Розділ 3 Програмна реалізація інформаційної системи класифікації аудіозаписів .....	41
3.1 Структура модулів інформаційної системи та їх взаємозв'язок.....	41
3.2 Особливості реалізації інформаційної системи класифікації аудіозаписів .....	44
3.3 Опис функціональних можливостей інформаційної системи класифікації аудіозаписів.....	46
3.4 Тестування системи .....	50
3.4.1 Створення плану тестування системи.....	50

	3
3.4.2 Результати тестування інформаційної системи .....	54
3.5 Висновки до розділу 3 .....	60
Висновки .....	61
Перелік посилань.....	62
Додатки	

## Перелік скорочень

Скорочення, термін, позначення	Пояснення
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
NLP	Natural Language Processing
BERT	Bidirectional Encoder Representations Transformers
GPT	Generative Pre-trained Transformer
AI	Artificial intelligence
QA	Quality Assurance
ПЗ	Програмне забезпечення
НМ	Нейронна мережі
БД	База даних
FK	Foreign Key
SST	Stanford Sentiment Treebank
QQP	Quora Question Pairs

## Вступ

Кваліфікаційна робота бакалавра присвячена розробці та програмній реалізації методу нейромережевого стенографування для класифікації аудіозаписів за ключовими словами, який дозволяє за аудіофайлом автоматизовано визначати його семантичну категорію.

**Актуальність.** Актуальністю дипломної роботи можна вважати те, що на даний момент в комп'ютерних технологіях задачі розпізнавання та розуміння контексту мови є дуже актуальними, тому що це може полегшити спосіб спілкування між людиною та комп'ютером, ці технології використовуються в медичних і військових застосуваннях, системах безпеки, автоматизованих системах розпізнавання та ідентифікації тощо.

**Об'єкт дослідження** – процеси стенографування аудіозаписів та класифікації аудіозаписів за ключовими словами їх вмісту.

**Предмет дослідження** – моделі, методи, алгоритми та засоби для нейромережевого стенографування аудіозаписів та класифікації текстів за ключовими словами.

**Мета кваліфікаційної роботи бакалавра** – розробка та програмна реалізація методу нейромережевого стенографування для класифікації аудіозаписів за ключовими словами.

**Завдання кваліфікаційної роботи бакалавра** – Провести аналіз предметної області й існуючих підходів до автоматизованого стенографування аудіозаписів, визначити особливості застосування нейромереж для задач автоматизованого стенографування аудіозаписів. Виконати аналіз існуючих рішень для задач автоматизованого стенографування аудіозаписів. Провести аналіз підходів до класифікації текстів за ключовими словами та огляд існуючих рішень для задач класифікації текстів. Розробити метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами. Розробити архітектуру нейронної мережі для стенографування аудіозаписів. Спроекувати структуру інформаційної системи класифікації аудіозаписів й структуру

відповідної бази даних. Спроекувати та створити інформаційну систему що використовує розроблений метод, виконати її тестування.

## **Розділ 1 Характеристика предметної області і постановка задачі**

### **1.1 Аналіз підходів до автоматизованого стенографування аудіозаписів**

За допомогою спеціальної програмної технології, що базується на машинному навчанні, можна автоматично перетворювати аудіозаписи на текст. Цей процес корисний для відтворення діалогів, лекцій та інших мовних комунікацій, особливо в сферах відеоконференцій, наукових презентацій та телефонних розмов.

Автоматичне стенографування аудіозаписів знаходить застосування в різних галузях і областях, таких як медицина, бізнес, телебачення та радіо, маркетинг тощо. У медицині його можна використовувати для аналізу консультацій між лікарями та пацієнтами, а також для оцінки результативності лікування. У бізнесі цей процес допомагає записувати конференції, розмови менеджерів та відео-зустрічі, щоб мати можливість відстежувати важливі деталі та рішення. У телебаченні та радіо автоматичне стенографування дозволяє записувати передачі, відео-ролики та інші відеоматеріали, щоб пізніше вирізати непотрібну інформацію та здійснювати відеомонтаж. В маркетингу воно застосовується для запису фокус-груп та досліджень ринку з метою визначення потреб покупців. Існують декілька підходів до автоматичного стенографування аудіозаписів, включаючи:

Чисельно-оптимізаційний підхід: він використовує методи оптимізації і математичні моделі для визначення найкращого розподілу слів в аудіозаписі. Приклади чисельно-оптимізаційного стенографування аудіозаписів:

метод Вітербі [1]: він використовує метод динамічного програмування для визначення найкращого розподілу слів в аудіозаписі.

метод Беллмана-Форда [2]: він використовує метод мінімізації часу відповіді для визначення найкращого розподілу слів в аудіозаписі.

Автоматична стенографія аудіозаписів базується на підходах, що використовують машинне навчання для перетворення аудіоданих у текст. Це може включати в себе використання нейронних мереж, методів інтерлінгвістики

та інших машинно-навчальних методів. У цьому напрямку можна виділити декілька прикладів.

Метод нейронних мереж використовує нейронні мережі для визначення найкращого відтворення аудіозапису у текстовому форматі.

Метод підбору параметрів використовує методи машинного навчання, такі як логістична регресія та дерева рішень, для визначення оптимального відтворення аудіозапису у текстовому форматі.

Метод представлення слова використовує машинне навчання, зокрема векторні представлення слів (word embeddings), для визначення найкращого відтворення аудіозапису у текстовому форматі.

Підходи на основі фонетики та морфології використовують знання про фонетику та морфологію мови для розбиття аудіозапису на слова та фрази.

Використання моделей морфологічного розпізнавання дозволяє використовувати морфологічну інформацію, наприклад, частини мови та форми слів, для визначення найкращого відтворення аудіозапису у текстовому форматі.

Використання алгоритмів машинного навчання, які використовують фонетику та морфологію, таких як нейронні мережі та дерева рішень, дозволяє визначити найкраще відтворення аудіозапису у текстовому форматі.

Процес автоматичної стенографії аудіозаписів на основі нейронних мереж включає такі кроки.

Підготовка даних: аудіозаписи перед обробкою повинні бути попередньо підготовлені і перетворені у відповідний формат для навчання нейронної мережі. Це може включати створення спектрограм або використання частотних кепстральних коефіцієнтів Mel (MFCC).

Розробка моделі: нейронна мережа, наприклад, згортова нейронна мережа (CNN) або рекурентна нейронна мережа (RNN), повинна бути розроблена і оптимізована для класифікації звуку.

Навчання моделі: модель повинна бути навчена на великому анотованому наборі даних аудіозаписів, щоб вивчити закономірності та особливості, які

допоможуть відрізнити різні класи. Це може включати навчання під наглядом, де модель представляє пари аудіозаписів і відповідних міток.

Оцінка та налаштування: навчену модель потрібно оцінити на окремому наборі даних, щоб оцінити її продуктивність та виявити області для поліпшення. Це можна зробити за допомогою перехресної перевірки або набору тестів. У разі потреби модель можна налаштувати, змінивши гіперпараметри або архітектуру.

Розгортання: остаточно навчена модель може бути розгорнута в реальних програмах, таких як системи розпізнавання мовлення, системи транскрипції мовлення у текст або системи ідентифікації мовця. Аудіозаписи можна вводити в модель, яка буде надавати прогнози класу у режимі реального часу.

Автоматична стенографія аудіозаписів є важливим напрямком розвитку, оскільки – є зростаючий попит на точні та ефективні методи обробки звуку, оскільки мовні технології стають все більш популярними. Завдяки розумним динамікам та іншим пристроям з голосовим керуванням, а також застосуванням мовних технологій у різних сферах, таких як охорона здоров'я, фінанси та освіта, зростає потреба у точних та ефективних рішеннях з обробки аудіо.

Ручна стенографія аудіозаписів є трудомісткою та часто супроводжується помилками, особливо при великому обсязі даних або мовах зі складною фонетикою. Автоматизована стенографія може забезпечити більш точні та ефективні рішення для цих завдань, сприяючи підвищенню продуктивності та зниженню помилок.

Розвиток штучного інтелекту та машинного навчання привів до значних досягнень у сфері автоматичної стенографії. Моделі, засновані на нейронних мережах, демонструють великий потенціал у розпізнаванні та транскрипції аудіо. Завдяки постійним удосконаленням алгоритмів та збільшенню обчислювальної потужності, автоматична стенографія стає все точнішою та ефективнішою.

Автоматична стенографія може сприяти конфіденційності та безпеці даних. Шляхом ефективною та безпечною обробки аудіо даних без необхідності ручної транскрипції можна запобігти можливим проблемам з конфіденційністю та безпекою інформації, що міститься в аудіозаписах. Можливість і потенціал

створення нейронної мережі для автоматичної класифікації аудіозаписів на основі ключових слів є активною областю дослідження обробки природної мови та машинного навчання. Завдяки вдосконаленню методів глибокого навчання та більшій доступності великих анотованих наборів даних стало можливо створити високоточні моделі класифікації звуку. Фактична корисність і корисність такої моделі, однак, залежить від конкретного випадку використання та якості навчальних даних. Тому можна з упевненістю сказати, що створення нейронної мережі для автоматичної класифікації аудіозаписів має велику актуальність.

## **1.2 Особливості застосування нейромереж для задач автоматизованого стенографування аудіозаписів**

Згортова нейронна мережа (ЗНМ) є важливою моделлю, яку спочатку запропонував Ле-Кун, а потім популяризував Крижевський з використанням моделі AlexNet. ЗНМ складається зі згорткових шарів та об'єднаних шарів, які поєднуються з повністю зв'язаною нейронною мережею. Завдання згорткових шарів та об'єднаних шарів полягає в навчанні вилучати відповідні локальні особливості з вхідних функцій, а повністю зв'язана нейронна мережа виконує завдання класифікації цих функцій. Наприклад, згорткові шари можуть виокремлювати краї об'єктів на зображеннях.

Конфігурація згорткового шару визначається кількістю використаних фільтрів, шириною та висотою фільтра, а також розміром зсуву, що визначає відстань між сприймаючими полями. Результати обчислень згорткового шару проходять через шар об'єднання, який об'єднує семантично схожі функції та зменшує розмір функціональних карт. Один з типів шарів об'єднання - макс-пулінг, який просто вибирає найбільше значення з локальних регіонів кожної функціональної карти. Розміри регіонів визначаються розміром зсуву, а їх кількість - розміром фільтра об'єднання. Згорткові нейронні мережі є ефективними моделями для автоматичного стенографування аудіозаписів. Вони використовуються для аналізу інформації, що міститься в аудіозаписах та їх

перетворення на текстовий формат. Однією з основних переваг ЗНМ є їх здатність розрізняти особливості аудіосигналів, такі як нотні висоти, інтонація та інші акустичні особливості. Ці характеристики можуть бути використані для розпізнавання мови, що є важливою функцією для стенографування аудіозаписів.

ЗНМ також відрізняються високою точністю розпізнавання мови, оскільки вони можуть використовувати глибокі нейронні мережі, що дозволяє їм розуміти складні залежності між вхідними даними та їх відповідями. Крім того, ЗНМ можуть бути навчені на великих наборах даних, що дозволяє їм покращувати свої навички розпізнавання мови з часом.

Ще однією перевагою використання ЗНМ є можливість використання попередньо навчених моделей. Це дозволяє зменшити час і ресурси, необхідні для навчання нової моделі, оскільки можна використовувати попередньо навчені моделі як основу для подальших досліджень.

Незважаючи на всі переваги, ЗНМ також мають свої обмеження. Наприклад, для досягнення високої точності розпізнавання мови за допомогою ЗНМ потрібно мати достатньо великий обсяг тренувальних даних.

Для розв'язання задач автоматизованого стенографування аудіозаписів можна використовувати різні нейромеревеві архітектури. Однією з ефективних архітектур є рекурентна нейронна мережа (RNN), яка здатна аналізувати послідовність вхідних даних та перетворювати їх на текстовий вихід. RNN використовується для створення моделей розпізнавання мови, які здатні розуміти мовлення та перетворювати його на текст.

Іншою ефективною архітектурою є варіаційний автокодер (VAE), який може генерувати нові аудіозаписи, що подібні до вхідних даних. VAE може використовуватись для вирішення завдань стенографування, де потрібно знайти шаблонні звуки та розпізнавати їх у вхідних даних.

Однією з основних переваг використання нейромеревевих архітектур для стенографування аудіозаписів є можливість автоматизувати процес розпізнавання мовлення та зменшити вплив людського фактору на точність результатів. Крім

того, застосування нейромережових архітектур може покращити швидкість та ефективність процесу стенографування.

Висновок, ЗНМ та нейромережові архітектури є потужними інструментами для розв'язання задач стенографування аудіозаписів. Вони дозволяють аналізувати та перетворювати аудіодані на текстовий формат, забезпечуючи автоматизацію процесу та підвищуючи точність результатів. Використання попередньо навчених моделей та різних архітектур дозволяє зменшити час і ресурси, необхідні для навчання нових моделей. Однак, необхідний достатній обсяг тренування для досягнення високої точності розпізнавання мови. Крім того, важливо враховувати особливості завдання та вибирати відповідну архітектуру, яка найкраще відповідає поставленій меті.

### **1.3 Аналіз існуючих рішень для задач автоматизованого стенографування аудіозаписів**

На сьогоднішній день інформаційні системи розпізнавання мови розділяють за ознаками такими як: залежність від диктора, деякі системи підстроюють під одного диктора, що не дозволяє їй на сто відсотків працювати з іншим диктором так як вимова у кожної людини різна – це і є другою ознакою за якою характеризують дані системи, а саме роздільна вимова. Також кожна з цих систем має своє певне призначення, як описувалось вище розпізнавання мови використовують в різних сферах тому системи також характеризують за їх функціональним призначенням.

Роздільність мови характеризується інтервалами між словами, а природна мова зазвичай є зливою. Проаналізувати зливу мову для систем розпізнавання важче, адже потрібно знаходити ледь помітні межі кожного слова, тим більше, що при розмові людина постійно видає звуки, що ще більше ускладнює зчитування.

У системах, які аналізують мову, аналіз відбувається цілісно, тобто при розпізнаванні система враховує всі фізичні характеристики сигналу, проте змістовне навантаження не береться до уваги.

Системи, що дозволяють користувачеві вводити текст за допомогою мікрофона вже аналізують контекст сказаного для більшої точності. Алгоритми, котрі використовують такі системи аналізують контекст кожного елемента мовлення, що і забезпечує точність вводу.

Широке застосування отримали і алгоритми динамічного програмування, котрі використовують для аналізу сказаного. Зазвичай в таких алгоритмах є пропис основних граматичних правил, котрі будуть підходити для більшої частини текстів, котрі можуть пройти через дану систему. Але зі збільшенням кількості прописаних правил, система розпізнаватиме меншу кількість речень, але якість розпізнавання буде найкращою.

Microsoft Speech Recognition [3] перетворює мову в текст. Цей API дає змогу отримувати аудіо з мікрофона чи аудіо файлу та обробляє його. Всі ці дії можливі в режимі реального часу, що дозволяє отримувати результати розпізнавання відразу після передачі звуку на сервер. Microsoft Speech API має кілька рівнів авторизації. Рівень Bing Speech API, який є «Безкоштовним», забезпечує 5000 перетворень на місяць, які є безкоштовними. Bing API використовується для трансфігурації мовлення виразів.

Automatic Speech Recognition [4] – це метод технології, який перетворює усну мову в письмові слова. Коротше кажучи, це перший крок до розробки голосових технологій. За допомогою голосової системи ASR ви можете створювати звуки та розпізнавати їх як слова. ASR є однією з найфундаментальніших лінгвістичних систем.

Google Cloud Speech API [5] дозволяє користувачам використовувати потужні моделі нейронних мереж на зручній у використанні платформі. API розпізнає безліч існуючих мов і діалектів, які допомагають отримувати їм одну з найбільших кількостей користувачів. Може записувати написаний користувачем текст, говорячи в мікрофон із програми, використовуючи голосові команди або записуючи аудіофайли тощо.

Розпізнавати завантажене аудіо, а також поєднувати рідне аудіо з власним сховищем Google [6], використовуючи ту саму технологію, яку Google

використовує для персоналізованої реклами, інформація для якої збирається за допомогою мікрофонів пристроїв користувачів. Збереження вихідних даних в хмарі допомагає користувачам мати швидкий доступ з різних пристроїв до файлів.

Функції Google Voice API: конвертація аудіо в текст, обробка безлічі різних мов й діалектів, можливість збереження готового файлу в хмарному середовищі, видаляє шум з аудозапису, може розпізнавати слова в різних контекстах.

#### **1.4 Аналіз підходів до класифікації текстів за ключовими словами**

Семантичний аналіз тексту стосується процесу розуміння значення тексту шляхом визначення та аналізу зв'язків між словами, фразами та поняттями. Метою семантичного аналізу є вилучення значущої інформації з тексту та представлення її структурованим способом.

Семантичний аналіз тексту включає в себе:

- розпізнавання іменованих об'єктів (NER) [7]: процес ідентифікації іменованих об'єктів, таких як люди, організації, місця та дати, у тексті.
- позначення частин мови (POS): процес визначення граматичних ролей слів у тексті, таких як іменник, дієслово, прикметник і прислівник;
- аналіз настроїв: процес визначення емоцій або думок, виражених у тексті, таких як позитивні, негативні чи нейтральні;
- розв'язання співвідповідності: процес ідентифікації, коли два або більше згадок у тексті стосуються однієї сутності;
- розпізнавання сенсу слова: процес визначення правильного значення слова на основі його контексту;
- моделювання теми: процес ідентифікації тем, присутніх у великій колекції текстових документів, і представлення документів у термінах цих тем.
- класифікація тексту: процес віднесення тексту до однієї чи кількох попередньо визначених категорій на основі його вмісту;

- позначення семантичних ролей (SRL): процес ідентифікації аргументів предиката та позначення їх відповідно до їхніх семантичних ролей;

Загалом, семантичний аналіз тексту передбачає широкий спектр технік і підходів, кожен із яких може дати цінне розуміння сенсу та структури тексту;

Опис необхідності реалізації нейромережевого стенографування для класифікації аудіозаписів за ключовими словами  
Класифікація аудіозаписів на основі ключових слів передбачає призначення набору попередньо визначених категорій аудіозапису на основі ключових слів або фраз, присутніх у його транскрипції. Існують різні підходи та техніки, які можна використовувати для цього завдання, зокрема:

Системи на основі правил: у цьому підході класифікація аудіозапису базується на наборі попередньо визначених правил або умов, які запускаються наявністю певних ключових слів або фраз у стенограмі.

Статистичні моделі: у цьому підході алгоритми машинного навчання, такі як наявний Байєс, дерево рішень і опорні векторні машини (SVM), використовуються для навчання моделі на позначеному наборі даних аудіозаписів і пов'язаних із ними ключових слів або категорій. Потім цю модель можна використовувати для класифікації нових аудіозаписів на основі ключових слів, присутніх у їхніх стенограмах.

Нейронні мережі: у цьому підході такі методи глибокого навчання, як згорткові нейронні мережі (CNN) і рекурентні нейронні мережі (RNN), використовуються для вивчення складних шаблонів в аудіозаписах і їх транскриптах, а також для класифікації нових записів на основі цих шаблонів.

Ці підходи можна використовувати окремо або в комбінації для створення більш складних і точних систем класифікації. Вибір підходу залежить від конкретних вимог і обмежень завдання класифікації, таких як кількість категорій, розмір набору навчальних даних і складність аудіо записів.

Автоматичне стенографування аудіозаписів з використанням методів класифікації за ключовими словами передбачає навчання моделей машинного навчання на анотованих наборах аудіоданих для класифікації записів у

попередньо визначені категорії на основі наявності ключових слів. Методи, які зазвичай використовуються в моделі, включають перетворення мови в текст, акустичну модель, мовну модель, визначення ключових слів тощо. Успіх цих моделей залежить від різних факторів, таких як якість голосу, мінливість динаміків і мовні проблеми. Практична реалізація цього підходу полягає у використанні попередньо навченої моделі перетворення мовлення в текст і точному налаштуванні моделі на тематичні дані для підвищення точності виявлення ключових слів. Інший підхід полягає у використанні алгоритмів неконтрольованого навчання, таких як кластеризація або тематичне моделювання, для класифікації транскрибованого мовлення відповідно до частоти ключових слів.

Класифікація текстів аудіозаписів за ключовими словами - це відома задача машинного навчання та можлива за допомогою різних методів, таких як класифікатори на основі навчальних даних, методи векторного представлення слів та нейромережі. Одним з важливих кроків в цьому процесі є визначення ключових слів в аудіо записах.

## **1. 5 Аналіз існуючих рішень для задач класифікації текстів**

Існує багато різних рішень для задач класифікації текстів, і кожне з них має свої переваги та недоліки, залежно від контексту застосування.

Один із підходів - це використання методів машинного навчання, таких як навчання з учителем на основі алгоритмів класифікації, наприклад, метод опорних векторів (SVM), Баєсів класифікатор[8], логістична регресія, дерева рішень тощо. Ці методи зазвичай використовуються з векторними представленнями тексту, такими як мішок слів (bag-of-words) або TF-IDF.

Ще один підхід - це використання нейронних мереж, а саме згорткових нейронних мереж (CNN). Вони можуть бути ефективними для обробки послідовностей та згенерованих моделей, що здатні адаптуватися до нових даних.

Розглянемо кілька інших методів, які використовуються для класифікації текстів:

Баєсівський класифікатор [8]: цей метод використовує статистичні методи для визначення ймовірності належності документа до певного класу на основі ймовірностей входження кожного слова в документ та ймовірностей наявності цих слів в кожному класі. Хоча цей метод є простим та швидким, він може бути менш точним за інші методи, особливо для складних класифікаційних задач.

Метод опорних векторів (SVM) [9]: цей метод базується на знаходженні оптимальної гіперплощини, яка може розділити дані в просторі вищого виміру на два класи. Він може бути застосований для класифікації текстів шляхом використання векторного представлення слів. Хоча SVM може бути більш точним за Баєсівський класифікатор, він може бути менш ефективним для великих обсягів даних.

Дерева рішень: цей метод використовує деревоподібну структуру для класифікації документів шляхом послідовного порівняння атрибутів документа з пороговими значеннями для кожного атрибута. Дерева рішень можуть бути легкими для розуміння та інтерпретації, але можуть мати проблему перенавчання, коли дерево стає занадто складним та враховує недостатню кількість даних.

Нейронні мережі: цей метод моделює класифікацію документів за допомогою штучної нейронної мережі, яка навчається на вхідних даних. Цей метод може бути більш точним та ефективним, але вимагає більшої кількості даних.

Інший підхід - це використання глибокого навчання і зокрема рекурентних нейронних мереж . РНМ показали вражаючі результати в багатьох завданнях обробки природньої мови, таких як машинний переклад, розпізнавання мови та аналіз тональності текстів. У рекурентних нейронних мережах, які є варіантом звичайних нейронних мереж, є петля зворотного зв'язку, що дозволяє передавати інформацію з одного часового кроку до наступного. Це особливо корисно для обробки текстів, оскільки вони можуть містити довготривалу залежність від контексту.

Одним із найбільш відомих прикладів PHM є Long Short-Term Memory [10] мережа. LSTMs дозволяють моделювати залежності в текстах, зокрема довготривалі залежності, шляхом зберігання внутрішнього стану мережі та його передачі наступному кроці часу.

Інший підхід полягає у використанні згорткових нейронних мереж (CNN) для обробки тексту. Вони зазвичай використовуються в обробці зображень, але також можуть бути застосовані до текстових даних. У ЗНМ текст спочатку перетворюється на матрицю, де кожному слову відповідає вектор. Потім ця матриця згортається за допомогою фільтрів різного розміру, що дозволяє визначити найсильніші зв'язки між словами в реченні. Після цього виконується операція об'єднання для об'єднання отриманих фільтрів у більшу чи меншу кількість блоків інформації.

У загальному, в задачі класифікації текстів використовуються різні методи та алгоритми, включаючи класичні методи машинного навчання, такі як наївний Баєс, метод опорних векторів (SVM) та різноманітні алгоритми глибокого навчання, такі як нейронні мережі. Кожен з цих методів має свої переваги та недоліки, і може бути ефективним в певних ситуаціях. Для досягнення кращих результатів, може бути корисно комбінувати різні методи та алгоритми.

Недавні тенденції у розробці моделей для класифікації текстів включають використання трансформерних архітектур, таких як BERT, GPT та інших, які можуть використовувати контекстуальну інформацію для покращення класифікації. Такі моделі можуть досягати дуже високої точності в класифікації текстів, але вони потребують більшої кількості обчислювальних ресурсів та даних для навчання.

Загалом, вибір підходу до класифікації текстів залежить від конкретної задачі та доступних ресурсів, таких як обчислювальна потужність та кількість даних для навчання.

## 1.6 Мета, задачі та вимоги до реалізації програмної системи

Метою роботи є розробка та програмна реалізація методу неймережевого стенографування для класифікації аудіозаписів за ключовими словами, для чого слід вирішити наступні задачі:

- розробити метод неймережевого стенографування для класифікації аудіозаписів за ключовими словами;
- розробити архітектуру нейронної мережі для стенографування аудіозаписів;
- спроектувати структуру інформаційної системи класифікації аудіозаписів, що використовує метод неймережевого стенографування для класифікації аудіозаписів за ключовими словами, й структуру відповідної БД;
- виконати вибір засобів розробки інформаційної системи ідентифікації особистості;
- виконати програмну реалізацію інформаційної системи класифікації аудіозаписів;
- провести тестування розробленої інформаційної системи.

## **Розділ 2 Метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами**

### **2.1 Опис і етапи виконання методу нейромережевого стенографування для класифікації аудіозаписів за ключовими словами**

Загалом роботу можна поділити на декілька етапів:

Створення бази аудіо файлів, котрі перетворюються у спектрограму, яка відображатиме частоту та часову інформацію звукової хвилі. Можливе використання текстового опису аудіозапису, як додаткові вхідні дані.

Нейромережа буде приймати спектрограми аудіозаписів й текстові описи, як вхідні дані та повертатиме ключові слова вже як вихідні дані. Допустиме використання різних архітектур нейромереж або їх комбінацію. Проводиться навчання на тренувальному наборі даних описаного в 1 графі даної таблиці.

Навчання може проводитись за допомогою декількох методів оптимізації, основні з них це: стохастичний градієнтний спуск, RMSProp, Adam.

Здійснюється за допомогою валідаційного набору даних, в випадку коли їх якість недостатня допускаються зміни в параметрах нейромережі, або перехід до інших методів навчання.

Вже навчена нейромережа використовується для класифікації нових аудіозаписів та отримання з них ключових слів, відповідно за прописаним раніше сценарієм.

Метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами з вхідними та вихідними даними є досить складним процесом, який вимагає певних знань та навичок в області машинного навчання та обробки сигналів.

Однією з головних перешкод при розробці такого методу є відбір відповідних функцій витягування ознак з аудіозаписів, котрі найкраще репрезентують звукові дані та сприяють покращенню якості класифікації. Для цього використовуються декілька методів обробки сигналів, такі як фур'є аналіз [11].



Рисунок 2. 1 – Схема використання Фур'є аналізу [11]

Згідно зі схемою вище, на початку потрібно взяти аудіозапис та виконати попередню обробку, таку як нормалізація, для того, щоб зробити сигнал менш шумним та більш стабільним, після чого переводиться розпізнавання та переведення в текстовий формат.

Потім виконується Фур'є-аналіз, що дозволяє перетворити звуковий сигнал в частотний діапазон.

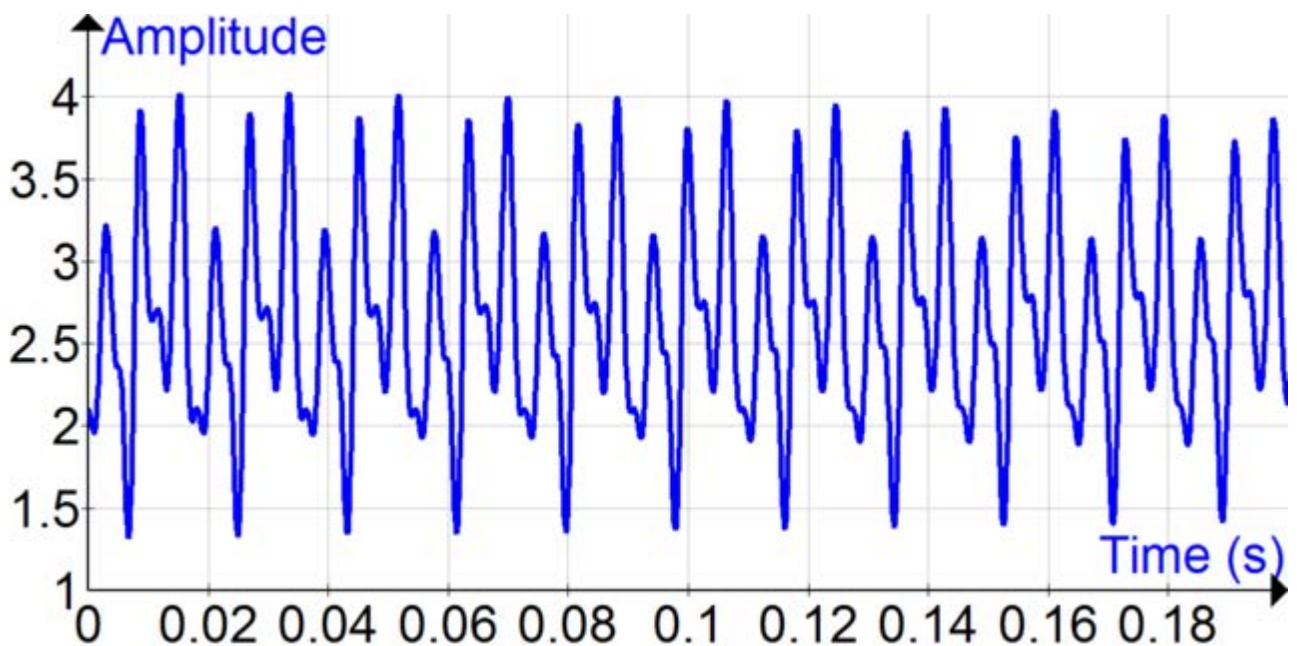


Рисунок 2.2 – Вигляд діаграми Фур'є [11]

Ця діаграма показує, як звуковий сигнал розбивається на його складові частоти. У часовій області сигнал відображається як функція часу, тобто ми бачимо, як змінюється амплітуда сигналу в часі. У частотній області, з іншого боку, сигнал відображається як функція частоти. Це означає, що ми можемо

бачити, які частоти складають сигнал та яка їх амплітуда. Фур'є-аналіз полягає в тому, що звуковий сигнал розбивається на його складові частоти за допомогою математичної операції, відомої як пряме перетворення Фур'є. Ця операція дозволяє розкласти будь-який періодичний сигнал на гармоніки, тобто сигнали з різними частотами, амплітудами та фазами.

Після розбиття сигналу на складові частоти, можна витягнути ознаки, що можуть бути використані для класифікації аудіозаписів за ключовими словами. Наприклад, можна визначити, які частоти присутні в сигналі та які амплітуди вони мають. Ці ознаки потім можуть бути використані як вхідні дані для нейромережі, що виконує класифікацію аудіозаписів за ключовими словами.

Далі, витягуються ознаки, що можуть бути використані для класифікації, такі як амплітуда, частота та енергія.

Після цього, використовуючи нейромережу, виконується класифікація за ключовими словами на основі витягнутих ознак. Ключові слова можуть бути визначені на основі тематики, що обговорюється, звертаючи увагу на основні терміни, які повторюються, або на частість мови (наприклад, іменники, дієслова, прикметники), що зустрічаються найбільше. Також можуть використовуватися певні ключові фрази або вирази, які вказують на конкретні події або дії.

І лише після цього на виході ми отримуємо вихідні мітки, що показують, до якої категорії належить аудіозапис.

## **2.2 Розробка архітектури нейронної мережі для стенографування аудіозаписів**

Для стенографування аудіозаписів можна розробити нейронну мережу з використанням архітектури, яка складається з двох частин: вхідної та вихідної.

Вхідна частина мережі буде відповідати за обробку аудіозапису та виконання попередньої обробки даних. Це можна зробити за допомогою згорткових та рекурентних шарів. Згорткові шари дозволяють розпізнавати різноманітні звуки та інші ознаки в аудіозаписі, тоді як рекурентні шари можуть

використовуватись для розпізнавання повторюваних патернів та структури аудіозапису.

Вихідна частина мережі буде відповідати за стенографування отриманих даних. Для цього можна використовувати послідовні шари, такі як LSTM-шари або GRU-шари, щоб зберігати контекст та уникнути втрати інформації.

Окрім того, можна використовувати механізм уваги (attention mechanism) для того, щоб зосередитись на найважливіших частинах аудіозапису та забезпечити кращу якість стенографування.

Існують різні варіації архітектур нейронних мереж для стенографування аудіозаписів, які можуть відрізнятися за кількістю шарів, кількістю нейронів в кожному шарі, типом функції активації, архітектурою енкодера та декодера та іншими параметрами. Для досягнення оптимальних результатів необхідно провести експерименти з різними архітектурами та параметрами мережі та порівняти їх результати. Одним з можливих напрямів для покращення результатів стенографування є використання комбінації нейронної мережі з іншими технологіями, такими як техніки обробки мовлення, автоматичне розпізнавання мови та інші.

Крім того, існують деякі проблеми, пов'язані з використанням нейронних мереж для стенографування аудіозаписів. Наприклад, нейронні мережі можуть мати обмежену точність для стенографування розмовної мови через складність розпізнавання різних голосів та акцентів. Крім того, стійкість до шуму та інших джерел спотворень може також бути обмеженою.

Опис архітектури:

Енкодер - використовується для перетворення аудіозапису в складний вектор. В якості енкодера можна використовувати рекурентні нейронні мережі, такі як LSTM або GRU. Такі мережі здатні запам'ятовувати інформацію про попередні частини сигналу, що допомагає знизити вплив шуму та втрат даних. Зокрема, енкодер призначений для перетворення аудіозапису на векторну представлення, яке потім буде оброблюватися декодером для відтворення

стенографічного тексту. Декодер натомість відповідає за генерацію вихідного тексту з векторного представлення.

Декодер - використовується для перетворення складного вектору, отриманого від енкодера, в послідовність слів (текст). Для цього можна використовувати рекурентні мережі з функцією активації softmax на кінці, яка допомагає звернути вектор у ймовірності різних символів.

Атеншн - допомагає врахувати важливість різних частин аудіозапису під час перетворення в текст. Використання атеншн допомагає зберегти контекст та зв'язки між різними частинами аудіозапису, що може підвищити точність стенографування.

Архітектура Transformer - можна використовувати замість рекурентних мереж. Ця архітектура здатна враховувати контекст та залежності між різними частинами вхідного сигналу без необхідності у використанні рекурентних мереж. Вона складається з множини однакових блоків, кожен з яких має три складові: механізм уваги, мережу зв'язків між словами та повнозв'язні шари. Модель Transformer використовується для кращої обробки послідовностей даних, зокрема тексту та аудіо. Вона дозволяє ефективно виконувати завдання машинного перекладу, а також підходить для стенографування аудіозаписів, оскільки забезпечує ефективну обробку довгих послідовностей даних.

Функція втрат - використовується для навчання мережі. Для стенографування аудіозаписів можна використовувати крос-ентропійну функцію втрат, яка знаходиться між вихідним текстом та прогнозованим текстом. Функція втрат визначає, наскільки точно мережа відтворює стенографічний текст на основі аудіозапису. Ця функція дозволяє оцінювати рівень точності мережі та підбирати гіперпараметри для досягнення кращих результатів.

Алгоритм зворотнього поширення помилки (backpropagation) - використовується для навчання мережі. Цей алгоритм дозволяє визначити, які параметри мережі потрібно змінити. Він дозволяє коригувати ваги мережі, зменшуючи значення функції втрат на кожному кроці навчання.

Гіперпараметри - потрібно налаштувати декілька гіперпараметрів, таких як кількість шарів, кількість нейронів, кількість проміжних векторів, коефіцієнт випадкового занулення (dropout), швидкість навчання (learning rate) тощо.

Перевірка мережі - після того, як нейронна мережа навчилася, її потрібно перевірити на тестових даних, щоб оцінити її точність та здатність до узагальнення. Для цього можна використовувати метрики, такі як точність, перплексія тощо.

Інтерфейс - для використання мережі для стенографування аудіозаписів потрібен зручний інтерфейс, що дозволяє користувачу завантажувати аудіофайли, отримувати результати та зберігати їх у зручному форматі.

Оптимізація - для покращення продуктивності мережі можна використовувати різні методи оптимізації, такі як стохастичний градієнтний спуск (SGD), адаптивний момент (Adam) та інші. Оптимізація полягає у підборі оптимальних значень гіперпараметрів та ваг мережі, зменшенні часу навчання та збільшенні точності результатів.

Розгортання - після навчання та перевірки мережі, її потрібно розгорнути в практичній додаток. Для цього можна використовувати сервіси хмарних обчислень, наприклад, AWS, Azure або Google Cloud.

Підтримка - для забезпечення гладкої роботи мережі потрібно забезпечувати підтримку та обслуговування. Це може включати в себе виправлення помилок, навчання моделі на нових даних, оновлення гіперпараметрів тощо.

Усі ці компоненти разом створюють архітектуру нейронної мережі для стенографування аудіозаписів. Завдяки цій архітектурі мережа може ефективно виконувати завдання стенографування аудіозаписів та забезпечувати високу точність результатів. Однак, для досягнення кращих результатів, мережу необхідно навчити на достатньо великій кількості різних аудіозаписів та оптимізувати її параметри.

Навчання нейронної мережі для стенографування аудіозаписів може бути виконано за допомогою навчальної вибірки аудіозаписів з відомим текстом та їх

стенографічного еквівалента. В процесі навчання мережі, параметри моделі змінюються таким чином, щоб мінімізувати функцію втрат. Функція втрат визначає, наскільки точні результати стенографування порівняно з відомим текстом. Після навчання мережі, вона може бути використана для стенографування нових аудіозаписів, які не мали раніше відомого тексту. Це досягається шляхом подання нового аудіозапису на вхід мережі, яка генерує стенографічний еквівалент. Для оцінки точності результатів стенографування, можна використовувати метрики, такі як середня або максимальна відносна похибка між стенографічним еквівалентом та відомим текстом.

Узагальнюючи, нейронна мережа для стенографування аудіозаписів - це складна система, яка включає в себе різні компоненти, такі як аудіо-препроцесор, енкодер, декодер, оптимізатор та функцію втрат. Ці компоненти спільно працюють, щоб забезпечити точність та ефективність стенографування аудіозаписів.

### **2.3 Підготовка робочих даних для системи**

Існує багато датасетів аудіозаписів з різними категоріями та типами даних. Для роботи пов'язаної з зчитуванням тексту з аудіофайлів та класифікації їх за ключовими словами, можуть бути корисні такі датасети.

LibriSpeech [12]: це датасет, який містить аудіозаписи з читанням англійських текстів, взятих з проекту LibriVox. Загальна кількість записів становить близько 1000 годин.

VoxCeleb [13]: цей датасет складається з аудіозаписів з відео на YouTube, що містять імена відомих людей. Загальна кількість записів становить більше 1 мільйона.

Common Voice [14]: це датасет, який містить аудіозаписи людей, що читають речення на різних мовах світу. Загальна кількість записів становить більше 9 тисяч годин.

UrbanSound8K [15]: це датасет, який містить аудіозаписи різних звуків, що можуть бути почуті в міському середовищі. Загальна кількість записів становить 8 тисяч.

ESC-50 [16]: цей датасет містить аудіозаписи з 50 різних звуків природи та техніки. Загальна кількість записів становить 2 тисячі.

Щодо категорій, то вони можуть варіюватись в залежності від конкретного датасету. Наприклад, для LibriSpeech[12] можна використовувати категорії "фіксовані фрази", "написані розмовним мовленням", "література тощо". Для VoxCeleb [13] можна використовувати категорії "відомі актори", "відомі музиканти", "відомі політики" тощо. Для Common Voice [14] можна використовувати категорії мов, такі як "англійська", "німецька", "іспанська" тощо.

Для UrbanSound8K та ESC-50 категорії можуть включати "шум дороги", "сигнали автомобілів", "звуки птахів", "звуки сільськогосподарської техніки" тощо.

Для роботи з класифікацією аудіозаписів за ключовими словами, можна використовувати методи машинного навчання, такі як нейронні мережі, різні моделі класифікації та інші. Наприклад, можна використовувати моделі з рекурентними нейронними мережами (RNN) або згортковими нейронними мережами (CNN), які добре показують себе в задачах розпізнавання мови та звуку.

Для побудови класифікатора необхідно підготувати датасет, що містить аудіозаписи та відповідні ключові слова. Зазвичай, ключові слова надаються як мітки (labels) для записів у датасеті. Для підготовки даних можна використовувати різні техніки обробки сигналів, такі як FFT, STFT,

Mel-спектрограма та інші, щоб отримати числові представлення аудіозаписів.

Ще одним прикладом датасету аудіозаписів є Google Speech Commands Dataset, який містить записи коротких команд, таких як "ввімкни світло", "вимкни музику", "відкрий двері" та інші. Цей датасет містить більше 100 тисяч записів, з яких кожен тривалістю близько 1 секунди.

Категорії в цьому датасеті включають різні голосові команди та слова, що використовуються в повсякденному житті, такі як "так", "ні", "почекай", "привіт", "допоможи" та інші. Цей датасет може бути використаний для розпізнавання коротких голосових команд, які використовуються в різних додатках, таких як голосові помічники, домашні асистенти та інші.

Для роботи з цим датасетом можна використовувати такі методи машинного навчання, як згорткові нейронні мережі (CNN), які можуть добре виконувати завдання класифікації голосових команд.

В цілому, датасети аудіозаписів можуть бути корисними для різних завдань, пов'язаних з обробкою мови та звуку, включаючи розпізнавання мови, класифікацію звуків, розпізнавання емоцій та інші. Однак, для успішної роботи з датасетами необхідно враховувати особливості технологій обробки звуку та методів машинного навчання, що використовуються в даній області.

## **2.4 Проєктування структури інформаційної системи класифікації аудіозаписів**

Структура інформаційної системи класифікації аудіозаписів може включати наступні компоненти:

база даних - це центральний елемент інформаційної системи, де зберігається вся необхідна інформація про аудіозаписи, включаючи метадані (назва, виконавець, тривалість тощо), а також самі аудіозаписи у вигляді файлів;

модуль класифікації - це програмний модуль, який автоматично класифікує аудіозаписи за певними критеріями, такими як жанр, настрій, тематика тощо. Для цього можуть використовуватися методи машинного навчання, аналізу спектрограм, алгоритми обробки сигналів тощо;

модуль пошуку - це програмний модуль, який дозволяє шукати аудіозаписи за різними критеріями, такими як назва, виконавець, жанр, настрій, тематика тощо. Для цього можуть використовуватися індексація бази даних та алгоритми пошуку;

модуль програвання - це програмний модуль, який дозволяє програвати аудіозаписи безпосередньо в інформаційній системі. Він може мати різні функції, такі як пауза, перемотування, налаштування гучності тощо;

модуль аналітики - це програмний модуль, який дозволяє аналізувати статистику використання інформаційної системи, таку як кількість програних аудіозаписів, тривалість використання тощо. Цей модуль може допомогти зрозуміти, як користувачі використовують інформаційну систему та як її можна поліпшити;

модуль аутентифікації та авторизації - це програмний модуль, який дозволяє забезпечити безпеку доступу до інформаційної системи. Він може включати в себе функції аутентифікації користувачів, контролю доступу до функціоналу системи, захисту від несанкціонованого доступу та зберігання історії дій користувачів;

модуль імпорту та експорту даних - це програмний модуль, який дозволяє імпортувати аудіозаписи з різних джерел, таких як комп'ютер, мобільний пристрій, інтернет тощо, а також експортувати аудіозаписи для використання в інших програмах чи пристроях;

модуль анотування даних - це програмний модуль, який дозволяє додавати до бази даних додаткові метадані про аудіозаписи, які можуть допомогти в подальшому пошуку та класифікації. Наприклад, це можуть бути описи контенту, теги, або коментарі користувачів.

Ці компоненти можуть бути інтегровані в один програмний продукт або мати окрему реалізацію в рамках різних систем. Залежно від потреб користувачів та розміру системи, можуть додаватися додаткові компоненти або функціонал.

Модуль класифікації є одним з найважливіших компонентів інформаційної системи класифікації аудіозаписів. Основна мета цього модулю полягає в автоматичному визначенні категорії аудіозапису на основі аналізу його звукових характеристик. Для досягнення цієї мети модуль класифікації може використовувати різноманітні методи машинного навчання, такі як нейронні мережі, дерева рішень, байєсовські класифікатори, метод опорних векторів тощо.

При цьому, для підготовки навчальної вибірки можуть використовуватися як аудіозаписи з відомою категорією, так і експертні оцінки.

Основні етапи роботи модуля класифікації виглядають наступним чином:

Підготовка даних - на цьому етапі аудіозаписи підлягають попередній обробці для отримання векторів ознак, що можуть включати в себе інформацію про спектральний склад, енергетичні характеристики, часові параметри тощо;

навчання моделі - на цьому етапі використовуються методи машинного навчання для побудови моделі класифікації на основі навчальної вибірки аудіозаписів з відомими категоріями;

класифікація нових даних - на цьому етапі використовується побудована модель для визначення категорії нових аудіозаписів на основі їх векторів ознак;

оцінка результатів - на цьому етапі проводиться оцінка точності класифікації за допомогою метрик, таких як точність, чутливість, специфічність тощо.

Оскільки робота модуля класифікації може залежати від різних факторів, таких як якість аудіозаписів, рівень шуму, кількість категорій тощо, досвідченість користувачів може бути великим чинником в успішності класифікації.

Для покращення роботи модуля можуть бути застосовані наступні підходи:

попередня обробка аудіозаписів - якість аудіозаписів може впливати на точність класифікації. Тому можуть використовуватися методи зменшення шуму, фільтрації, нормалізації та інші;

розширення навчальної вибірки - для покращення точності класифікації може бути використана додаткова навчальна вибірка аудіозаписів, які містять додаткові категорії або мають відмінності від наявної навчальної вибірки;

використання ансамблю моделей - можуть бути застосовані різні моделі машинного навчання, і результати можуть бути об'єднані для покращення точності класифікації;

використання активного навчання - в цьому підході система може звертатися до користувача для отримання додаткової інформації про категорії аудіозаписів, що дозволяє покращити точність класифікації.

У результаті використання цих підходів можна досягнути покращення точності класифікації, що дозволяє досягти кращих результатів у роботі інформаційної системи класифікації аудіозаписів.

Модуль анотування даних для інформаційної системи класифікації аудіозаписів - це програмне забезпечення, що дозволяє користувачам відмічати та класифікувати аудіозаписи відповідно до встановлених категорій. Основна мета модуля анотування даних - допомогти користувачам зібрати та організувати навчальні дані для модуля класифікації аудіозаписів. Основні функції модуля анотування даних включають наступне:

Додавання нових категорій - користувачі можуть додавати нові категорії аудіозаписів для класифікації;

анотування аудіозаписів - користувачі можуть відмічати категорії для кожного аудіозапису;

класифікація даних - модуль може використовувати алгоритми класифікації для автоматичного визначення категорій аудіозаписів;

фільтрація та пошук даних - користувачі можуть фільтрувати та шукати аудіозаписи за певними параметрами, такими як назва, категорія, дата додавання тощо;

експорт даних - модуль може дозволяти користувачам експортувати анотовані дані у вигляді таблиць або файлів для використання в інших програмах.

Для досягнення найкращих результатів, модуль анотування даних може бути інтегрований з іншими модулями інформаційної системи класифікації аудіозаписів, такими як модуль збору даних та модуль класифікації. Це дозволяє забезпечити стійкість та точність в роботі інформаційної системи класифікації аудіозаписів. Для досягнення найкращих результатів, модуль анотування даних може мати такі особливості:

Масштабованість - модуль повинен мати можливість обробки великих обсягів даних, що дозволить користувачам працювати з великою кількістю аудіозаписів та категоріями.

Доступ до даних - модуль повинен мати механізм захисту даних, щоб запобігти несанкціонованому доступу до анотованих даних.

Гнучкість - модуль повинен бути гнучким та налаштовуватись під потреби користувачів. Це дозволить збирати дані для різних типів аналізу, таких як відновлення мовлення або розпізнавання голосу.

Інтерфейс користувача - модуль повинен мати простий та інтуїтивний інтерфейс користувача, що дозволяє користувачам швидко та легко анотувати аудіозаписи.

Звітність - модуль повинен мати звітність, що дозволяє користувачам отримувати статистику про кількість аудіозаписів та їх категорій, що допомагає в організації роботи та прийнятті рішень.

Всі ці особливості дозволяють модулю анотування даних стати ефективним інструментом для збору та організації даних для модуля класифікації аудіозаписів.

## 2.5 Даталогічна модель бази даних інформаційної системи класифікації аудіозаписів

Для інформаційної системи класифікації аудіозаписів можна розробити даталогічну модель бази даних, яка включатиме наступні сутності.



Рисунок 2.3 – Схематичне зображення етапів роботи програми

Вхідні дані: аудіозапис: це вхідний сигнал звуку, який буде класифікуватися програмою.

Передобробка: завантаження аудіозапису: аудіозапис завантажується в програму для подальшої обробки. Попередній аналіз: виконується попередній аналіз аудіозапису, такий як нормалізація гучності, фільтрація шуму або видалення зайвих частот.

Екстракція ознак: витягнення аудіоознак: з аудіозапису витягуються різні ознаки, такі як мел-частотні кепстральні коефіцієнти (MFCC), спектральні функції, енергія сигналу тощо. Ці ознаки використовуються для подальшого аналізу та класифікації.

Класифікація: модель машинного навчання: використовується побудована модель машинного навчання, така як нейронна мережа, дерево рішень або метод опорних векторів, для класифікації аудіозапису на певні категорії або класи. Тренування моделі: модель машинного навчання може бути навчена на попередньо підготовлених наборах даних, які містять аудіозаписи разом з відповідними мітками класів.

Класифікація аудіозапису: отримані ознаки аудіозапису подаються на вхід моделі, яка виробляє прогноз щодо класу, до якого належить аудіозапис. Наприклад, класифікація може вказувати на те, чи є аудіозапис музичним, голосовим, шумовим тощо.

Вихідні дані: класифікація аудіозапису: після проходження через модель машинного навчання отримуємо прогнозований клас, до якого належить аудіозапис. Наприклад, може бути повернута мітка або назва класу, який вказує на жанр музики або тип звукового вмісту. Ймовірності класів: окрім самої класифікації, модель може повертати ймовірності для кожного класу. Це може бути корисно для подальшого аналізу і прийняття рішень, особливо коли є декілька можливих класів аудіозаписів.

Сутності (Entities): аудіозапис: основна сутність, що представляє сам аудіозапис. Може містити поля, такі як ідентифікатор, назва, тривалість, шлях до файлу і т.д. Категорія: сутність, що описує категорії аудіозаписів, до яких вони

можуть бути класифіковані. Може містити поля, такі як ідентифікатор, назва, опис і т.д.

**Зв'язки (Relationships):** аудіозапис може належати до однієї або багатьох категорій. Це багато-до-багатьох відношення, оскільки одна категорія може мати багато аудіозаписів, і один аудіозапис може належати до багатьох категорій. Цей зв'язок можна моделювати за допомогою додаткової таблиці, яка містить ідентифікатори аудіозаписів і категорій. Категорія може мати багато аудіозаписів, але кожен аудіозапис може належати лише до однієї категорії. Це один-до-багатьох відношення, і його можна моделювати, додавши зовнішній ключ "категорія" до таблиці аудіозаписів.

**Атрибути (Attributes):** аудіозапис може мати додаткові атрибути, такі як дата створення, автор, рейтинг, коментарі тощо. Категорія може мати додаткові атрибути, такі як батьківська категорія (якщо є ієрархічна структура категорій), інформація про користувача, який створив категорію, тощо.

**Запити (Queries):** запити можуть включати пошук аудіозаписів за категорією, додавання нових аудіозаписів, оновлення атрибутів аудіозаписів, видалення аудіозаписів і категорій тощо.

**Пошук за категорією:** можна виконати запит для отримання всіх аудіозаписів, які належать до певної категорії або її підкатегорій. Це може бути корисно для навігації або фільтрації аудіозаписів за їх класифікацією.

**Додавання нових аудіозаписів:** можна виконати запит для додавання нових аудіозаписів до бази даних. Це включатиме вказання необхідних полів, таких як назва, тривалість, шлях до файлу, а також можливість призначити їх категорію.

**Оновлення атрибутів аудіозаписів:** можна виконати запит для зміни атрибутів існуючих аудіозаписів, таких як назва, автор, рейтинг, коментарі тощо. Це дозволяє актуалізувати інформацію про аудіозаписи з часом або відповідно до потреб користувачів.

**Видалення аудіозаписів і категорій:** можна виконати запит для видалення аудіозаписів або категорій з бази даних. Це може бути корисно для управління контентом і видалення застарілих або непотрібних елементів.

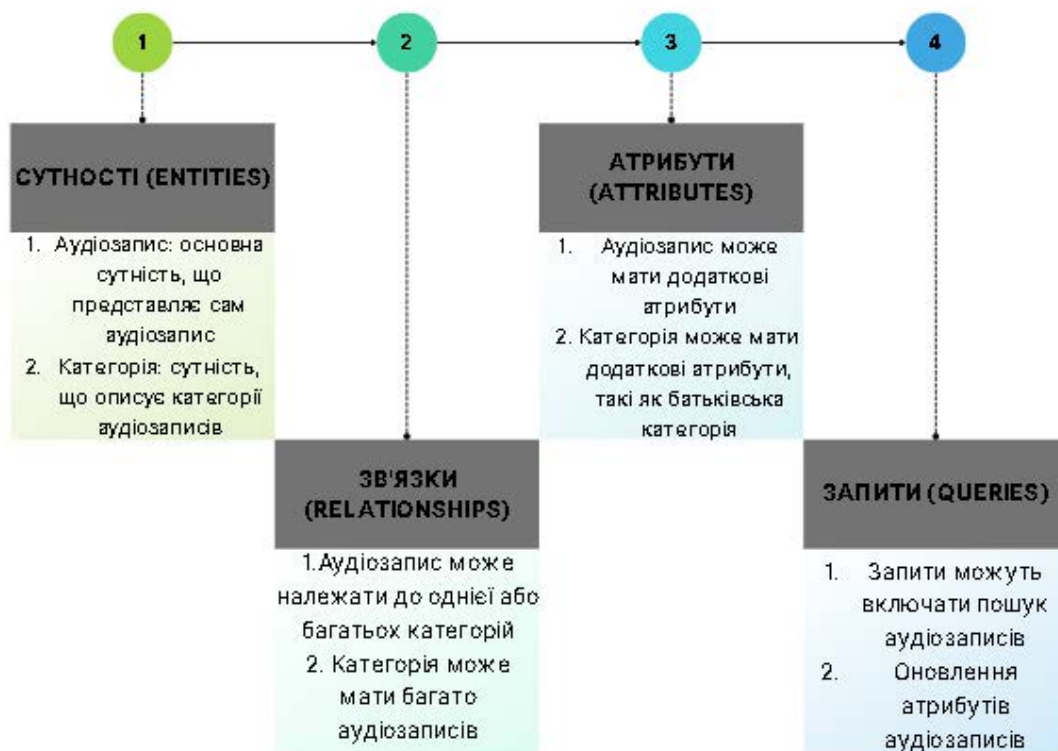


Рисунок 2. – Схематичне зображення даталогічної моделі

Ця даталогічна модель бази даних відображає основні сутності, зв'язки та атрибути, які можуть бути використані в інформаційній системі класифікації аудіозаписів.

## 2.6 Вибір засобів розробки інформаційної системи класифікації аудіозаписів

Вибір засобів розробки інформаційної системи класифікації аудіозаписів залежить від ряду факторів, таких як обсяг та складність проекту, потреби відповідної функціональності, доступність ресурсів для розробки та підтримки системи, рівень експертизи команди розробників.

Основні засоби розробки інформаційних систем включають в себе мови програмування, бази даних та фреймворки. Нижче перераховано деякі з них, які

можуть бути використані для розробки інформаційної системи класифікації аудіозаписів:

Python [17] – це одна з найпопулярніших мов програмування для розробки інформаційних систем. Python має велику кількість бібліотек обробки звуку та машинного навчання, які дозволяють ефективно обробляти аудіозаписи.

TensorFlow [18] – це програмне забезпечення з відкритим кодом для розробки та навчання моделей машинного навчання, які можна використовувати для класифікації аудіозаписів.

Keras [19] – це фреймворк для розробки нейронних мереж, який також можна використовувати для класифікації аудіозаписів.

Librosa [20] – це бібліотека для аналізу звукових сигналів, яка дозволяє виконувати ряд операцій з обробки аудіозаписів, таких як визначення тембру, темпу та ін.

PyAudio [21] – це бібліотека для роботи з аудіозаписами у реальному часі, яка дозволяє записувати, відтворювати та обробляти аудіо.

MongoDB [22] – це документ-орієнтована база даних, яка може бути використана для зберігання та керування аудіозаписами та додатковою інформацією про них.

MySQL [23] – це реляційна база даних, яка може бути використана для зберігання та керування аудіозаписами та пов'язаною з ними інформацією.

Django [24] – це фреймворк для розробки веб-додатків на мові Python, який може бути використаний для розробки інтерфейсу користувача та керування базою даних.

Flask - це мінімалістичний фреймворк для розробки веб-додатків на мові Python, який також може бути використаний для розробки інтерфейсу користувача та керування базою даних.

Вибір конкретних засобів залежить від специфіки проекту та вимог до функціональності, тому важливо ретельно проаналізувати всі можливі варіанти та вибрати найбільш підходящі для проекту.

Python [17] – це високорівнева інтерпретована об'єктно-орієнтована мова програмування з лаконічним і зрозумілим синтаксисом. Python має багато переваг для розробки інформаційних систем, таких як класифікація записів.

Простота і лаконічність коду. Python має дуже читабельний і легкий для розуміння синтаксис, який дозволяє швидко розробляти, тестувати та підтримувати код. Це зменшує витрати на розробку, забезпечує швидший процес розробки та дозволяє зосередитися на більш важливих аспектах вашого проекту.

Безліч бібліотек і фреймворків. Python має багато стандартних бібліотек і фреймворків, які дозволяють швидко розробляти якісні програми. Такі бібліотеки, як NumPy, Pandas, Scikit-learn, Tensorflow, PyTorch, librosa тощо, можна використовувати для розробки інформаційних систем класифікації записів для обробки та аналізу аудіоданих.

Переносимість коду. Python - це крос-платформна мова програмування, що дозволяє запускати програми на різних операційних системах без необхідності виконувати додаткові налаштування.

Зручність тестування. Python має вбудовану підтримку модульного тестування, що дозволяє швидко та ефективно тестувати програму на наявність помилок.

Велика спільнота розробників. Python має велику та активну спільноту розробників, яка постійно розробляє нові бібліотеки та фреймворки.

Легкість використання. Python є однією з найбільш легких мов програмування для вивчення та використання. Це забезпечує широку доступність для розробників з різними рівнями досвіду.

Швидкість розробки. Python дозволяє розробляти інформаційні системи класифікації аудіозаписів значно швидше, ніж більшість інших мов програмування. Це забезпечує більш ефективний процес розробки та можливість швидко змінювати функціональність системи за потреби.

Підтримка паралельної обробки даних. Python має різні бібліотеки, такі як NumPy та Pandas, що дозволяють ефективно працювати з багатовимірними

масивами даних. Крім того, Python також підтримує паралельну обробку даних, що дозволяє зменшити час обробки даних.

Підтримка машинного навчання. Python має різні бібліотеки, такі як Scikit-learn, TensorFlow та PyTorch, що дозволяють розробляти моделі машинного навчання. Це дозволяє здійснювати класифікацію аудіозаписів за допомогою навчання моделей з використанням алгоритмів машинного навчання.

Наявність великої кількості документації та підтримки. Python має велику кількість документації та підтримки, що дозволяє розробникам швидко знайти відповіді на свої запитання та розв'язати проблеми.

Отже, Python - це один з найкращих варіантів для розробки інформаційних систем класифікації аудіозаписів. Він має багато переваг, таких як простота та лаконічність коду, багато бібліотек для роботи з даними, підтримку машинного навчання та паралельної обробки даних. Python також має велику спільноту розробників, що регулярно розробляють нові бібліотеки та модулі для полегшення роботи з даними та розробки інформаційних систем.

Одним з найбільш відомих бібліотек для роботи з аудіо є librosa. Ця бібліотека надає зручний інтерфейс для аналізу аудіозаписів, включаючи функції для зчитування та запису аудіофайлів, візуалізації звуку, екстракції характеристик звуку та т.д. Іншими бібліотеками, які можна використовувати для розробки інформаційної системи класифікації аудіозаписів, є PyAudio, SpeechRecognition [25] та аудіо-процесор.

Speech recognizer - це технологія розпізнавання мовлення, яка перетворює аудіозапис мовлення на текстову форму. Вона використовується для автоматичного стенографування аудіозаписів і забезпечує швидку та точну обробку великої кількості даних.

Speech recognizer використовується в інформаційних системах для стенографування аудіозаписів, оскільки вона дозволяє замінити ручний процес стенографування, що може займати багато часу і бути неточним. Замість цього, система може автоматично розпізнавати мовлення на аудіозаписах і перетворювати його на текст, що значно збільшує швидкість та точність процесу.

Одна з переваг Speech recognizer полягає в тому, що вона може працювати з великою кількістю різноманітних мов і діалектів, що дозволяє застосовувати її у різних країнах та для різних типів мовлення.

Крім того, Speech recognizer може бути дуже корисною для розробки інформаційних систем, які використовують аналіз тексту. Наприклад, вона може допомогти в розумінні відгуків користувачів на додатки, ефективно класифікувати текст на основі ключових слів або виконувати автоматичну індексацію документів для пошуку.

Python також має деякі інструменти для роботи з нейромережами. Наприклад, бібліотека TensorFlow надає можливість створювати та тренувати нейромережі, які можуть бути використані для класифікації аудіозаписів. До інших бібліотек, які можна використовувати для роботи з нейромережами в Python, належать Keras, Theano та PyTorch.

Загалом, використання Python для розробки інформаційної системи класифікації аудіозаписів забезпечує необхідну швидкість розробки та можливість розширення функціоналу системи в майбутньому.

## **2.7 Висновки до розділу 2**

У даному розділі було описано метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами. Було розглянуто етапи виконання цього методу, від підготовки даних до проєктування структури інформаційної системи та даталогічної моделі бази даних.

Також було описано розробку архітектури нейронної мережі для стенографування аудіозаписів та вибір засобів розробки інформаційної системи класифікації аудіозаписів.

Метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами є досить ефективним, оскільки дозволяє автоматично розпізнавати тематику аудіозапису та класифікувати його відповідно до цієї

тематики. Розробка архітектури нейронної мережі виконувалась з урахуванням специфіки задачі класифікації аудіозаписів.

Для успішної реалізації даної інформаційної системи важливо правильно підготувати робочі дані та проектувати структуру системи, враховуючи потреби користувачів. Для цього було описано даталогічну модель бази даних інформаційної системи класифікації аудіозаписів.

Вибір засобів розробки інформаційної системи є важливою складовою успішної реалізації проекту. У даному розділі було описано вибір Python як основної мови програмування для розробки інформаційної системи. Python має декілька переваг, серед яких простота вивчення, широкий спектр бібліотек для наукових обчислень та зручність у використанні

## **Розділ 3 Програмна реалізація інформаційної системи класифікації аудіозаписів**

### **3.1 Структура модулів інформаційної системи та їх взаємозв'язок**

Структура модулів інформаційної системи класифікації аудіозаписів створеної в рамках роботи над кваліфікаційною роботою є доволі простою, проте завдяки цій структурі система функціонує без багів та виконує поставлені перед нею задачі.

Головний модуль: цей модуль відповідає за запуск інформаційної системи та взаємодію з користувачем. Включає в себе віджети графічного інтерфейсу користувача (GUI), такі як кнопки, поле виведення тексту, поле виведення класифікації і т. д. Містить функції, які відповідають на події взаємодії користувача, наприклад, натискання кнопки "Відкрити файл". Викликає функції з модуля класифікатора для аналізу та класифікації аудіозапису. Цей модуль є основою всієї системи і керує виконанням інших модулів.

Модуль класифікатора (TextClassifier): цей модуль містить клас TextClassifier, який відповідає за класифікацію тексту згідно зазначеними ключовими словами. Містить списки ключових слів для класифікації з різних тем, таких як будівництво, навчання, медицина. Містить метод classify\_text, який приймає текст і тему, та повертає булеве значення, що вказує, чи містить текст ключові слова з відповідної теми. Цей модуль дозволяє визначати тему аудіозапису з використанням ключових слів.

Модуль розпізнавання мови (speech\_recognition): цей модуль залежить від сторонньої бібліотеки speech\_recognition, яка використовується для розпізнавання мовлення з аудіо-файлу. Використовується для створення об'єкту Recognizer і AudioFile, а також для запису аудіо-файлу та його розпізнавання.

Модуль графічного інтерфейсу (tkinter): цей модуль залежить від стандартної бібліотеки tkinter, яка використовується для створення графічного інтерфейсу користувача. Містить класи і віджети для побудови графічного інтерфейсу, такі як вікна, кнопки, текстові поля. Цей модуль дозволяє

користувачеві взаємодіяти з системою шляхом натискання кнопок, введення тексту тощо.

Включає функції, які обробляють події користувача, такі як натискання кнопок і введення тексту.

Модуль діалогових вікон (filedialog, messagebox): цей модуль залежить від стандартної бібліотеки tkinter і містить функції для відкриття діалогових вікон, таких як вікно вибору файлу. Використовується для отримання шляху до аудіо-файлу, який обирає користувач, та для відображення повідомлень про помилки.

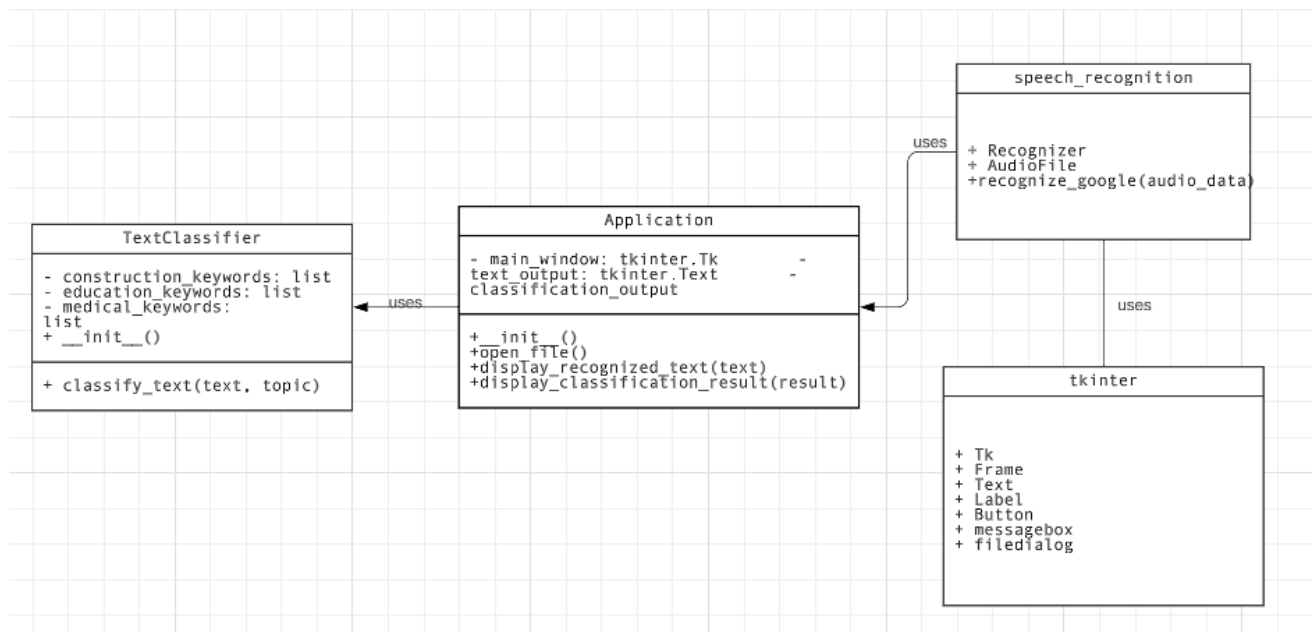


Рисунок 3.1– Діаграма класів

Ця діаграма показує основні класи і взаємодію між ними. **TextClassifier** відповідає за класифікацію тексту за ключовими словами. **Application** відповідає за графічний інтерфейс програми та включає методи для відображення розпізнаного тексту, класифікації та повідомлень. **speech\_recognition** використовується для розпізнавання мовлення, а **tkinter** - для створення графічного інтерфейсу користувача.

Класи, які наявні в проекті:

- Клас **TextClassifier** - відповідає за класифікацію тексту за ключовими словами. Включає наступні методи:
  - `__init__()`: ініціалізує список ключових слів для кожної теми.

- `classify_text(text, topic)`: класифікує заданий текст за ключовими словами відповідно до теми.

- Клас `Application` - відповідає за графічний інтерфейс програми. Включає наступні методи:

- `__init__()`: ініціалізує головне вікно та компоненти інтерфейсу.

- `open_file()`: відкриває діалогове вікно для вибору аудіо-файлу та запускає процес розпізнавання та класифікації.

- Інші необхідні класи:

- `speech_recognition`: використовується для розпізнавання мовлення з аудіо-файлу.

- `tkinter`: використовується для створення графічного інтерфейсу.

Ці класи взаємодіють між собою наступним чином: об'єкт `Application` створюється і запускає головне вікно програми.

Користувач може натиснути кнопку "Відкрити файл", що викликає метод `open_file()` у класі `Application`.

У методі `open_file()`, використовуючи модуль `filedialog`, обирається аудіо-файл для обробки.

Після вибору аудіо-файлу, відбувається розпізнавання мовлення. Використовується об'єкт `Recognizer` з модуля `speech_recognition` для читання аудіо-файлу та отримання об'єкту `AudioData`.

Отриманий об'єкт `AudioData` передається до методу `recognize_google()` об'єкта `Recognizer` для розпізнавання мовлення з використанням `Google Web Speech API`.

Розпізнаний текст відображається у віджеті `text_output`.

Об'єкт `TextClassifier` створюється для класифікації розпізнаного тексту.

Метод `classify_text()` об'єкта `TextClassifier` викликається для класифікації тексту за ключовими словами відповідно до теми.

Результати класифікації відображаються у віджеті `classification_output`.

## 3.2 Особливості реалізації інформаційної системи класифікації аудіозаписів

Особливості реалізації інформаційної системи класифікації аудіозаписів на основі ключових слів включають наступні аспекти.

Використання бібліотеки `speech_recognition`: система використовує бібліотеку `speech_recognition` для розпізнавання мови з аудіозапису. Ця бібліотека надає доступ до різних голосових розпізнавачів, включаючи Google Web Speech API.

Класифікація тексту за ключовими словами: для класифікації тексту система використовує підхід заснований на ключових словах. Кожна тема має свій набір ключових слів, і система перевіряє наявність цих слів у тексті для визначення теми. Це дозволяє швидко та просто класифікувати текст.

Можливість редагування ключових слів: система дозволяє редагувати набори ключових слів для кожної теми. Це дозволяє користувачу встановлювати власні ключові слова або редагувати наявні для відповідності їх власним потребам або специфічним вимогам.

Гнучкий інтерфейс користувача: система має інтерфейс користувача, побудований на основі бібліотеки Tkinter, що надає можливість відкривати аудіозаписи, переглядати розпізнаний текст, відображати результати класифікації та повідомлення про помилки. Користувач може взаємодіяти з системою за допомогою кнопок та віджетів.

Підтримка української мови: система розпізнавання мови та класифікації підтримує українську мову. Це дозволяє користувачам використовувати українські аудіозаписи.

Обробка аудіозаписів: система включає функціональність для обробки аудіозаписів. Вона може приймати аудіофайли різних форматів, таких як MP3, WAV тощо. Перед класифікацією аудіозапису система може провести попередню обробку, таку як підвищення якості звуку, зменшення шуму або нормалізацію амплітуди, для отримання кращих результатів.

Автоматична сегментація аудіозаписів: деякі аудіозаписи можуть мати довжину більше однієї фрази або секції. Система може автоматично розділити аудіозапис на відповідні сегменти для окремої класифікації. Це забезпечує більш точну класифікацію та можливість відстеження тематики по всьому аудіозапису.

Масштабованість: система розроблена з урахуванням масштабованості, що дозволяє їй працювати з великою кількістю аудіозаписів та обробляти їх ефективно. Застосування оптимізаційних технік та використання потокової обробки дозволяє підвищити продуктивність та швидкість роботи системи.

Візуалізація результатів: система може відображати результати класифікації у зручному для користувача форматі. Це може бути представлення у вигляді текстового виводу, графіків або діаграм, що дозволяє швидко сприймати та аналізувати результати класифікації.

Підтримка розширення: система розроблена з урахуванням можливості розширення функціональності. Наприклад, ви можете додати нові ключові слова, створити нові теми.

Машинне навчання: Система може використовувати методи машинного навчання для покращення точності класифікації. Вона може навчитися розпізнавати нові ключові слова або використовувати алгоритми класифікації, такі як наївний Байєс або метод опорних векторів, для покращення результатів.

Адаптація до вхідних даних: система може адаптуватися до різних типів вхідних даних. Вона може працювати з різними форматами аудіозаписів і мовами, забезпечуючи гнучкість та універсальність.

Відстеження стану системи: Система може вести журнал дій та помилок, що дозволяє відстежувати стан та продуктивність системи. Це полегшує виявлення проблем та вдосконалення роботи системи.

Застосування в реальному часі: система може працювати в режимі реального часу, що означає, що вона може обробляти аудіозаписи негайно під час їхнього надходження. Це особливо корисно для використання в системах спілкування в реальному часі, таких як чат-боти або голосові асистенти.

Безпека і конфіденційність: система дотримується вимог безпеки та конфіденційності. Вона може захищати оброблювані дані та забезпечувати їх конфіденційність, щоб забезпечити надійність та довіру користувачів.

Оптимізація продуктивності: система має оптимізований алгоритм обробки аудіозаписів та класифікації, що дозволяє досягати високої швидкості роботи та оптимального використання ресурсів обчислювальної системи. Це особливо важливо при роботі з великим обсягом даних.

Розширення до інших мов та діалектів: система може бути розширена для підтримки інших мов або діалектів. Шляхом навчання моделі на додаткових даних та використання відповідних ключових слів, система може бути адаптована для розпізнавання та класифікації тексту в різних мовах.

Інтеграція з іншими системами: система може бути легко інтегрована з іншими інформаційними системами або додатками. Наприклад, вона може надавати API для обміну даними та інтеграції з іншими сервісами..

Дослідження та вдосконалення: реалізація системи класифікації аудіозаписів може бути постійною роботою над вдосконаленням та дослідженням. Це може включати оптимізацію алгоритмів, вдосконалення точності класифікації, розширення функціональності та впровадження нових технологій.

Ці особливості реалізації дозволяють інформаційній системі класифікації аудіозаписів ефективно та точно визначати теми за ключовими словами, працювати з різними мовами та аудіозаписами у разі необхідності розширення даного проекту.

### **3.3 Опис функціональних можливостей інформаційної системи класифікації аудіозаписів**

Інформаційна система класифікації аудіозаписів, реалізована в ході виконання кваліфікаційної роботи бакалавра – має наступні функціональні можливості.

Візуальний інтерфейс: Інформаційна система має графічний інтерфейс, реалізований за допомогою бібліотеки tkinter.

Покращений візуальний інтерфейс: Відображення тексту та результатів класифікації відбувається в зручних та зрозумілих віджетах. Текстові поля `text_output` та `classification_output` мають відповідні розміри та стилі шрифту для зручного відображення тексту.

Користувацька зручність: Інтерфейс системи максимально простий та зрозумілий для користувача. Кнопка "Відкрити файл" надає зручний спосіб вибору аудіо-файлу, а текстові поля `text_output` та `classification_output` чітко відображають результати розпізнавання та класифікації.



Рисунок 3.2 – Загальний вигляд інтерфейсу програми

Відкриття аудіо-файлу: Користувач може обрати аудіо-файл для подальшого аналізу та класифікації. Відбувається за допомогою кнопки "Відкрити файл", яка викликає діалогове вікно вибору файлу.

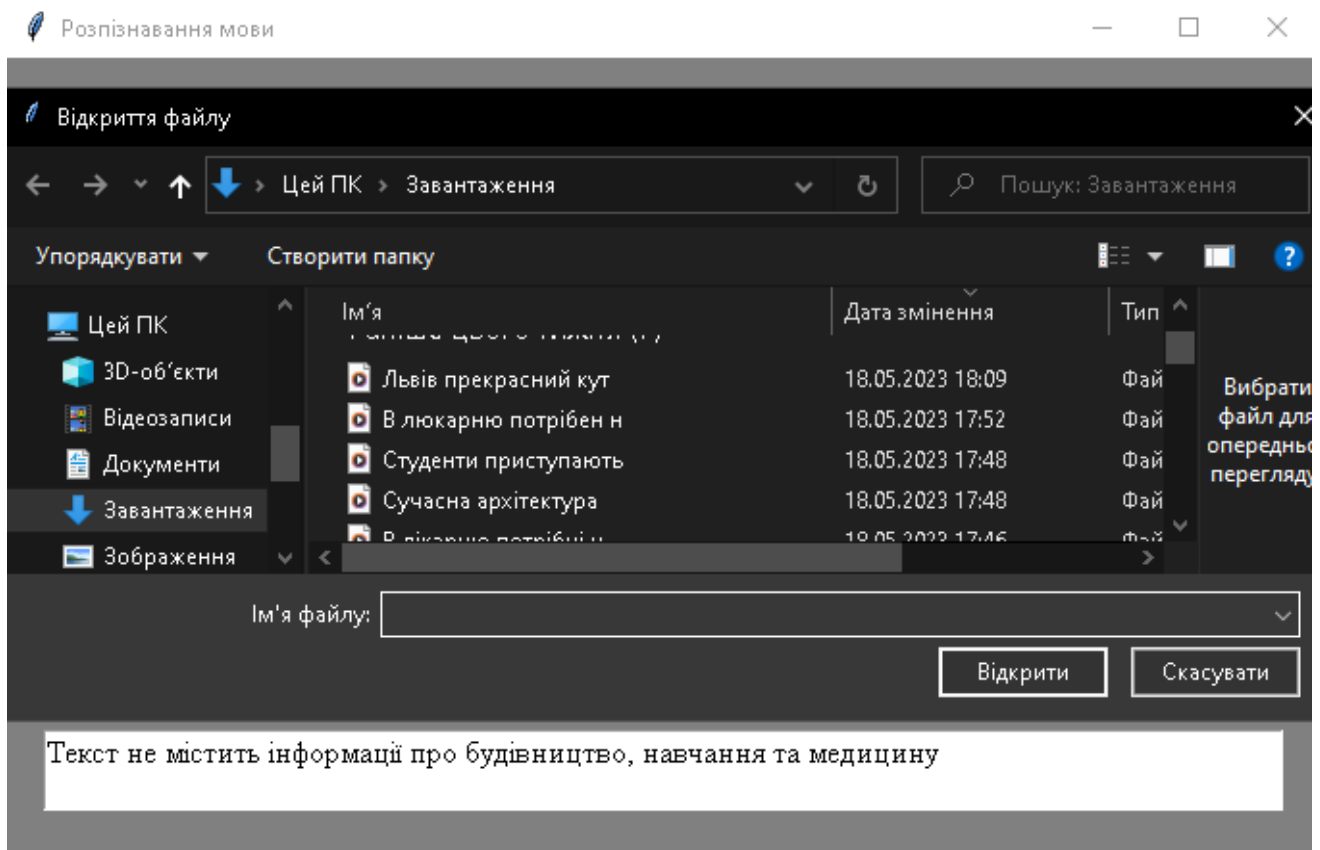


Рисунок 3.3 – Результат на тискання на кнопку вибору файлу

Розпізнавання мови: Вибраний аудіо-файл програвается, і система використовує модуль `speech_recognition` для розпізнавання мови в аудіо-записі. Використовується `Google Web Speech API` для отримання текстового представлення мовлення.

Вивід розпізаного тексту: Результат розпізнавання мови виводиться в текстове поле `text_output`. Розпізнаний текст відображається в цьому полі.

Класифікація тексту: Розпізнаний текст піддається класифікації за допомогою класу `TextClassifier`. За ключовими словами, визначеними для кожної теми (будівництво, навчання, медицина), текст класифікується відповідно до тематики.

Вивід результату класифікації: Результат класифікації виводиться в текстове поле `classification_output`. Залежно від теми, до якої належить текст, виводиться відповідне повідомлення про наявність або відсутність інформації щодо будівництва, навчання або медицини.

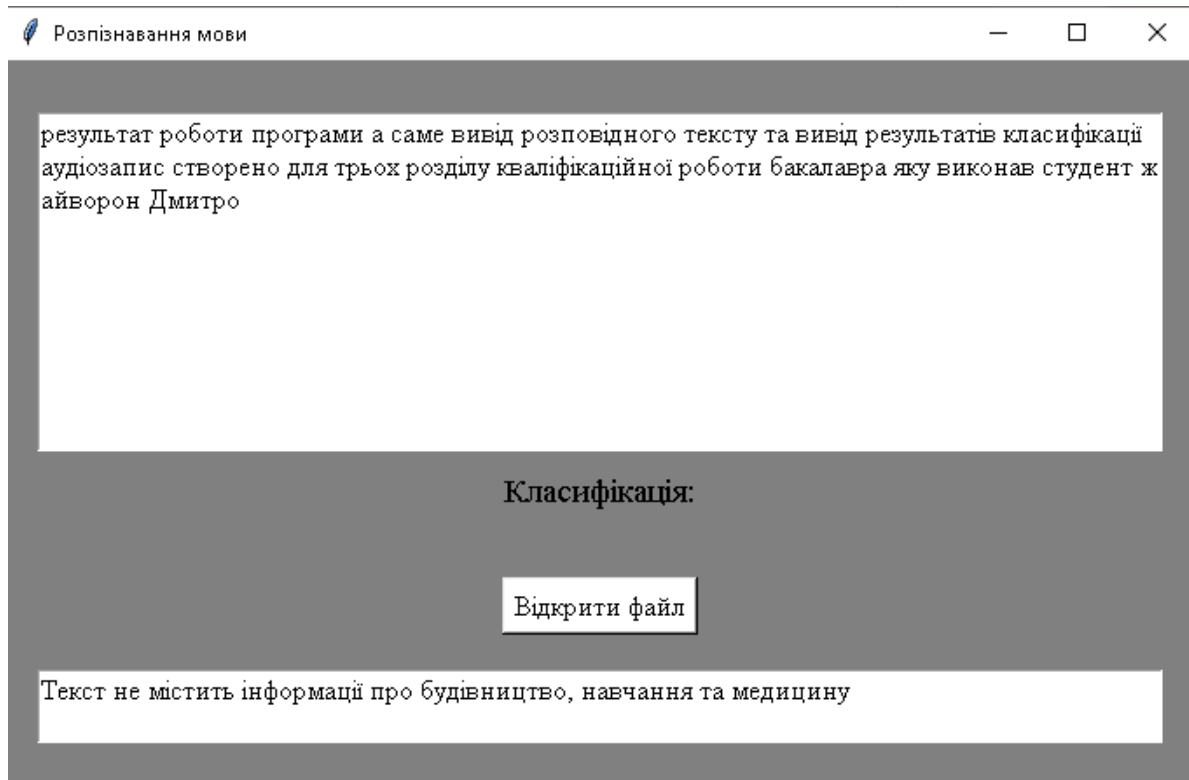


Рисунок 3.4 – Виводи розпізнаного тексту та результат класифікації

Обробка помилок: У разі виникнення помилки під час розпізнавання мови, відображається повідомлення про помилку.

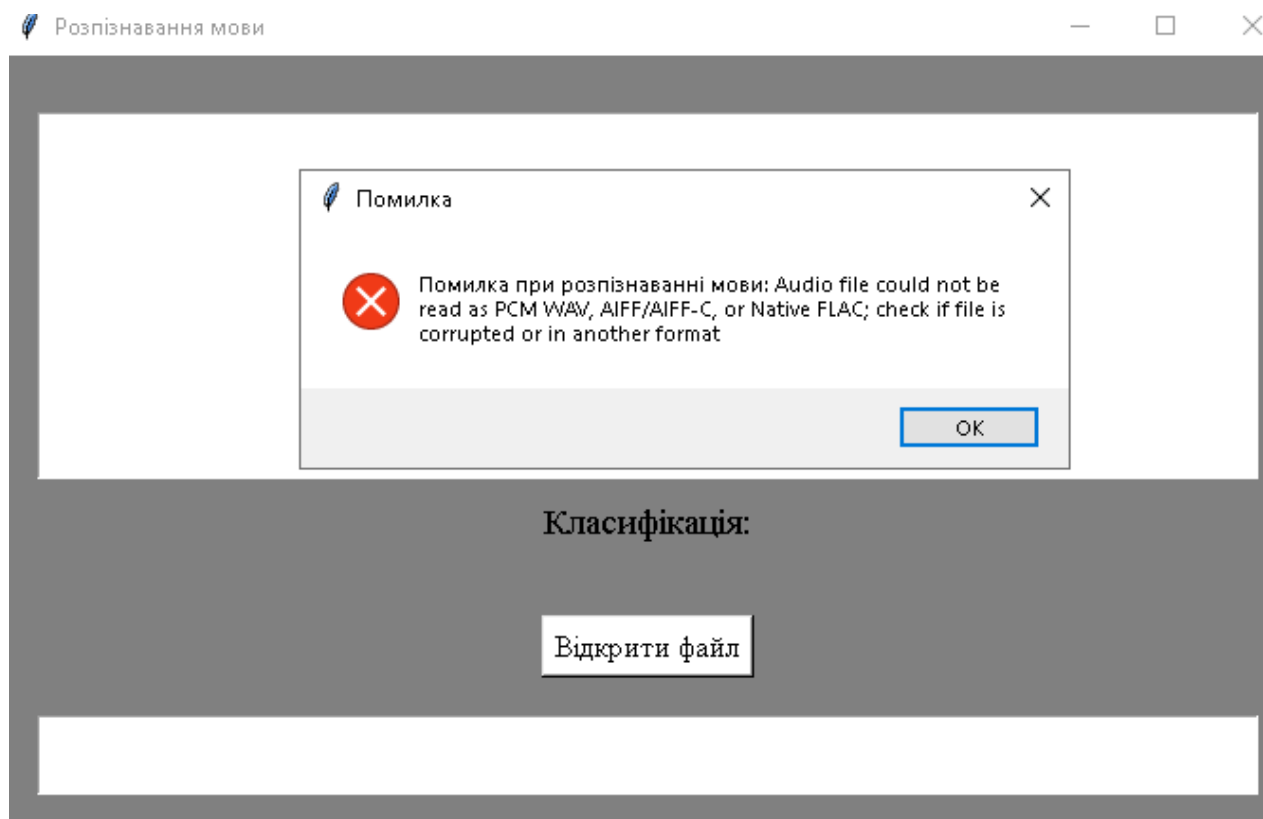


Рисунок 3.5 – Вивід вікна помилки

Можливість розширення: Система може бути розширена шляхом додавання нових ключових слів та тем для класифікації тексту. Клас TextClassifier може бути розширений для визначення нових тем та відповідних ключових слів для класифікації.

Підтримка української мови: Система використовує Google Web Speech API для розпізнавання мови в українському варіанті (language='uk-UA'). Це дозволяє користувачам класифікувати аудіозаписи, що містять українськомовний текст.

Обробка помилок: У випадку помилки під час розпізнавання мови або відкриття файлу, система виводить повідомлення про помилку за допомогою діалогового вікна.

Загальною функціональністю інформаційної системи є відкриття, розпізнавання та класифікація аудіозаписів за ключовими словами, зручний візуальний інтерфейс та можливість розширення для підтримки нових тем та мов.

### **3.4 Тестування системи**

#### **3.4.1 Створення плану тестування системи**

Метою плану тестування є перевірка функціональності, коректності та продуктивності інформаційної системи класифікації аудіозаписів. План передбачає проведення різних тестів, які охоплюють різні аспекти системи, включаючи відкриття файлів, розпізнавання мовлення, класифікацію тексту та користувацький інтерфейс.

Тестування розпочинається з підготовки тестових аудіо-файлів, що містять мовлення з різними темами. Це дозволяє перевірити роботу системи на різних вхідних даних. Тестування включає перевірку головного вікна програми, відкриття файлів, відображення розпізнаного тексту та правильність класифікації. Також проводиться тестування на різних сценаріях помилок, а також на відповідність дизайну та взаємодії з користувачем.

Тестування точності класифікації включає перевірку правильності класифікації тексту за використанням відомих аудіо-файлів з відомим мовним

вмістом, а також перевірку з використанням нових ключових слів. Проводиться також тестування на помилкових випадках, таких як аудіо-файли зі спотвореним мовленням або шумом, а також з мовленням іншою мовою.

Тестування продуктивності включає вимірювання часу, необхідного для розпізнавання та класифікації тексту з різних аудіо-файлів. Це дозволяє оцінити продуктивність системи при обробці великих обсягів даних.

В цілому, план тестування має на меті забезпечити перевірку функціональності, надійності та коректності роботи інформаційної системи класифікації аудіозаписів. Цей план охоплює широкий спектр сценаріїв тестування, що дозволяє перевірити різні аспекти системи та виявити можливі проблеми.

Загальний підхід до плану тестування полягає в створенні тестових аудіо-файлів з різними мовними вмістом та тривалістю. Це дозволяє перевірити, як система працює з різними типами вхідних даних. Тестування включає такі етапи, як перевірка відображення головного вікна програми, відкриття файлів, розпізнавання мовлення, класифікацію тексту та коректність повідомлень про помилки.

Додатково, в плані передбачено перевірку коректності класифікації тексту з використанням відомих аудіо-файлів з відомим мовним вмістом. Також, шляхом введення нових ключових слів для теми, можна перевірити, як система впорається з класифікацією тексту на основі цих нових ключових слів.

План тестування також включає перевірку відповідності дизайну та взаємодії програми очікуванням користувача. Проводиться перевірка відображення назв віджетів та кнопок, а також коректність повідомлень про помилки.

Окремий аспект плану тестування становить оцінка продуктивності системи шляхом вимірювання часу, необхідного для розпізнавання та класифікації тексту з різних аудіо-файлів. Це дозволяє переконатися, що система працює ефективно та забезпечує задовільний час виконання завдань.

Загальна перевірка функціональності включає в себе виконання різних комбінацій вибору аудіо-файлів, перегляду розпізнаного тексту та класифікації, щоб переконатися, що програма працює стабільно і коректно в усіх сценаріях.

План тестування також передбачає випробування системи на помилкових випадках. Наприклад, створення аудіо-файлу зі спотвореним мовленням або шумом для перевірки, як програма впорається з розпізнаванням такого мовлення. Також, створення аудіо-файлу з мовленням іншою мовою, а не українською, для перевірки правильності розпізнавання тексту.

Останній етап плану тестування полягає в загальній перевірці функціональності та взаємодії програми. Проводяться різні взаємодії з програмою, такі як натискання кнопок, вибір файлів і перегляд результатів, щоб переконатися, що взаємодія є зручною та інтуїтивно зрозумілою.

Далі будуть описані задачі поставлені в ході створення плану тестування інформаційної системи яка створена для стенографування та класифікації аудіозаписів за ключовими словами.

– Задача 1: Підготовка тестових аудіо-файлів

Створити кілька аудіо-файлів з різними темами, такими як будівництво, навчання та медицина.

Забезпечити, щоб аудіо-файли мали різні мовні вміст та тривалість.

– Задача 2: Виконання тестів

Відкрити програму та перевірити, чи відображається головне вікно з відповідними віджетами.

Натиснути кнопку "Відкрити файл" та переконатися, що відкривається діалогове вікно для вибору аудіо-файлу.

Вибрати тестовий аудіо-файл та переконатися, що розпізнаний текст правильно відображається у віджеті "Розпізнаний текст".

Перевірити, чи правильно класифікується розпізнаний текст за ключовими словами та відображається відповідна класифікація у віджеті

"Класифікація" Повторити попередні кроки для різних тестових аудіо-файлів з різним мовним вмістом.

Перевірити, чи правильно відображається повідомлення про помилку у разі виникнення помилки при розпізнаванні мовлення.

- Задача 3: Перевірка коректності класифікації

Перевірити, чи правильно класифікується текст з використанням відомих аудіо-файлів з відомим мовним вмістом.

Ввести нові ключові слова для теми та перевірити, чи класифікується текст правильно з використанням цих нових ключових слів.

- Задача 4: Тестування користувацького інтерфейсу

Перевірити, чи правильно відображаються назви віджетів та кнопок.

Натиснути кнопку "Відкрити файл" без вибору аудіо-файлу та переконатися, що з'являється повідомлення про помилку.

Натиснути кнопку "Відкрити файл" та вибрати аудіо-файл, який не підтримується програмою (наприклад, з іншим форматом), і переконатися, що з'являється повідомлення про помилку.

Перевірити, чи відображаються розпізнаний текст та класифікація у відповідних віджетах після вибору тестового аудіо-файлу.

- Задача 5: Тестування точності класифікації

Запустити програму з різними тестовими аудіо-файлами з відомим мовним вмістом і перевірити, чи правильно класифікується текст відповідно до теми.

Створити власні тестові аудіо-файли з різним мовним вмістом, які відповідають або не відповідають темам класифікації, і перевірити, чи правильно класифікується текст.

- Задача 6: Тестування помилкових випадків

Створити аудіо-файл зі спотвореним мовленням або шумом і перевірити, як програма впорається з розпізнаванням такого мовлення.

Створити аудіо-файл з мовленням іншою мовою, а не українською, і перевірити, чи програма правильно розпізнає текст.

- Задача 7: Загальна перевірка функціональності Виконати різні комбінації вибору аудіо-файлів, перегляду розпізнаного тексту та

класифікації, щоб переконатися, що програма працює стабільно і коректно в усіх сценаріях.

– Задача 8: Тестування продуктивності

Виконати тестування з великими аудіо-файлами для перевірки продуктивності програми.

Виміряти час, необхідний для розпізнавання і класифікації тексту з різних аудіо-файлів, і переконатися, що час виконання є прийнятним.

Цей план тестування допоможе виявити можливі проблеми та недоліки в роботі системи класифікації аудіозаписів. Важливо забезпечити, щоб усі аспекти функціональності, надійності, коректності та продуктивності були враховані під час тестування, щоб забезпечити якісну та надійну роботу програми.

### 3.4.2 Результати тестування інформаційної системи

Під час тестування згідно «Задачі 1» було створено 3 аудіозаписи за допомогою сервісу онлайн диктора «Narakeet [26]».

The image shows three instances of the Narakeet online dictation interface. Each instance consists of the following elements:

- ГОЛОС:** A dropdown menu for selecting a voice. The first instance is set to 'Oleksandr', the second to 'Olena', and the third to 'Svitlana'.
- ФОРМАТ:** A dropdown menu for selecting an audio format, all set to 'wav'.
- ТЕКСТ:** A text input area containing the text to be recorded. The first instance contains 'Сучасна архітектура дуже сильно відрізняється від архітектури 19 століття.', the second contains 'Студенти приступають до задачі екзаменів.', and the third contains 'В лікарню потрібні нові лікарі'.
- БІЛЬШЕ ОПЦІЙ:** A button to expand the interface for more settings.
- СТВОРИТИ АУДІО:** A button to generate the audio file from the provided text and settings.

Рисунок 3.6 – Процес створення аудіозаписів

Після того, як були створені аудіозаписи, перевіряємо чи відкривається програма, запустивши проект ми бачимо, що відкривається вікно програми, тому переходимо до виконання тестів пов'язаних з пунктом «Задача 2».

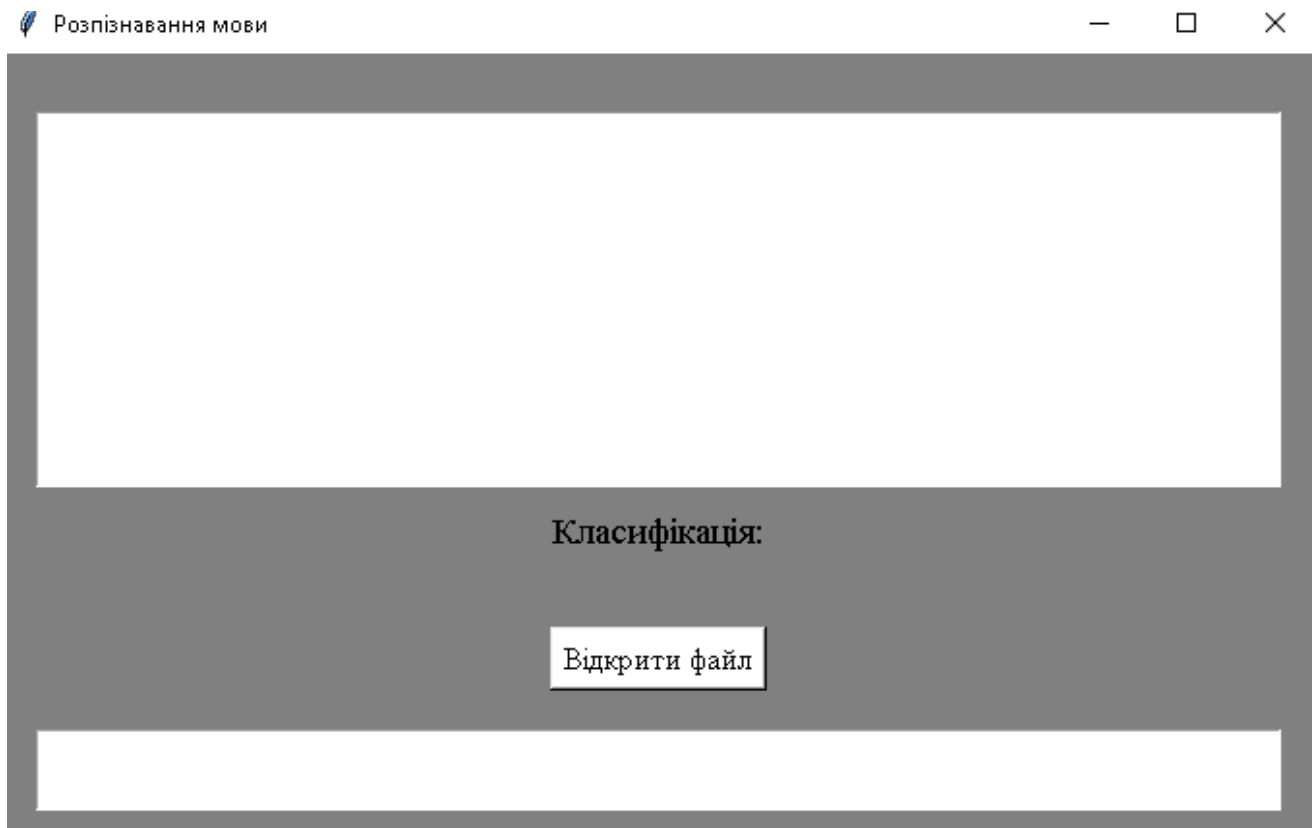


Рисунок 3.7 – Вигляд інтерфейсу програми.

Натискаємо кнопку «Відкрити файл» – перед нами з'являється меню вибору файлу.

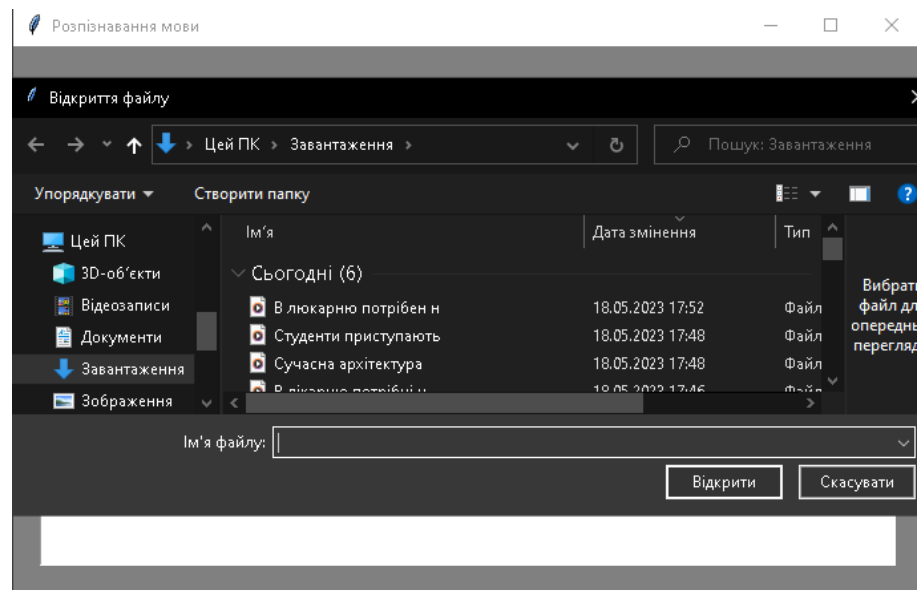


Рисунок 3.8 – Результат роботи кнопки «Відкрити файл»

Обираємо файли, які ми записали раніше за допомогою онлайн диктора.

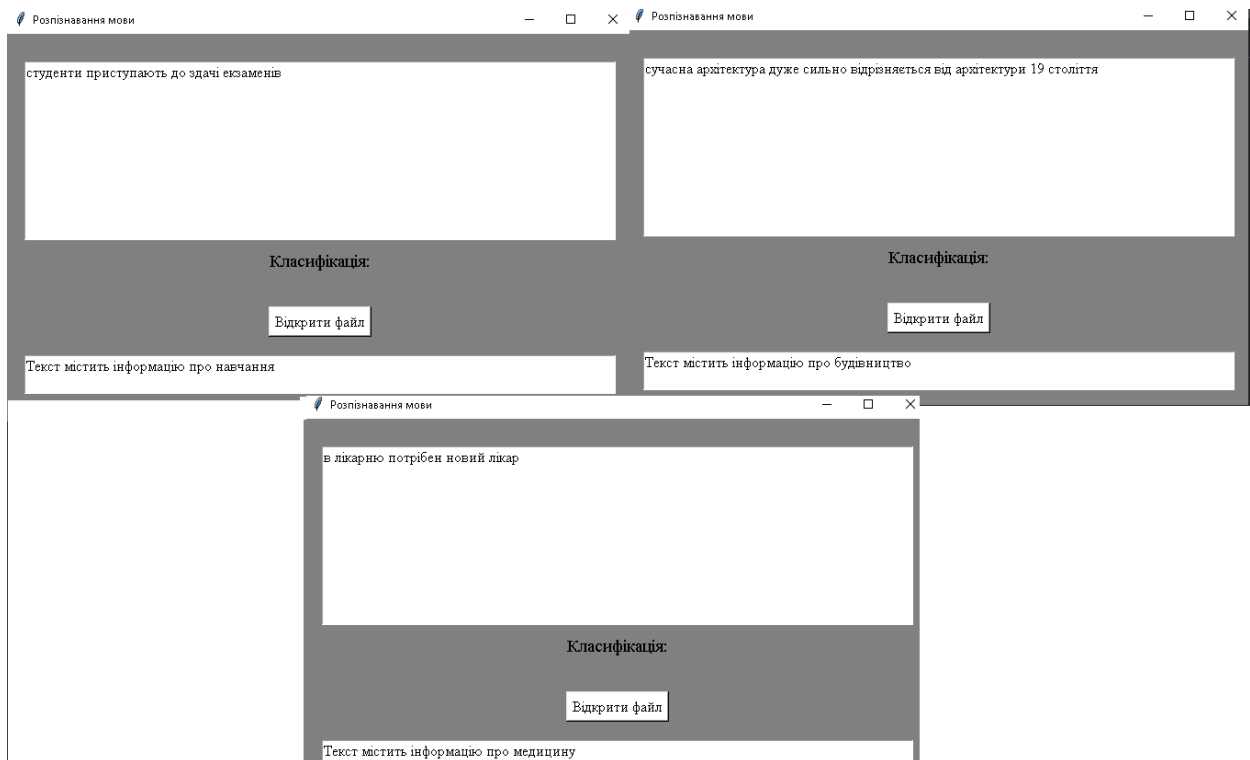


Рисунок 3.9 – Результат роботи програми з файлами

Згідно результатів стенографування та класифікації, можна зробити висновок, що програма поставлену задачу виконала – текст зчитався в повному обсязі, ключові слова з тексту були вибрані та тексту присвоєно відповідну категорію.

Виконуємо перевірку коректності відображення тексту помилки, який з'являється тоді коли файл не було обрано або обрано неправильний формат файлу.

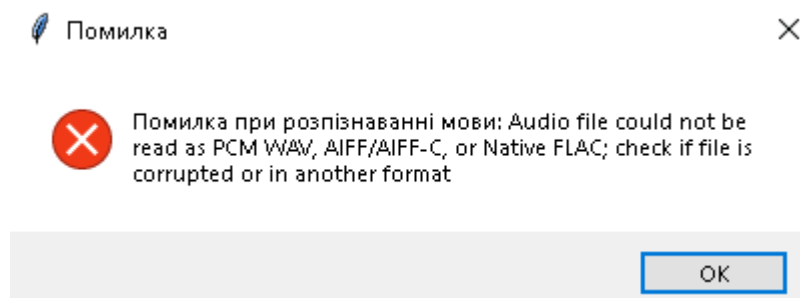


Рисунок 3.10 – Вигляд вікна помилки

В результаті маємо вікно помилки, яке вказує користувачеві, що формат файлу був обраний неправильно.

Створюємо файл англійською мовою за допомогою сервісу «Zvukogram [27]» і перевіряємо результат роботи програми, а саме можливі помилки в роботі чи повну відмову програми.

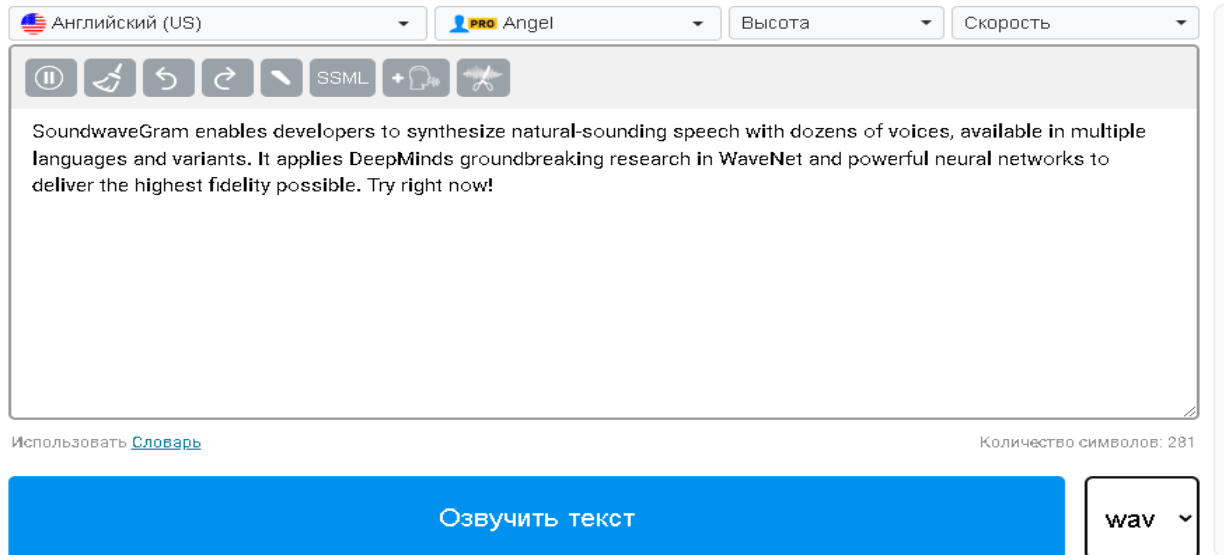


Рисунок 3.11 – Створення аудіо файлу з текстом на англійській мові

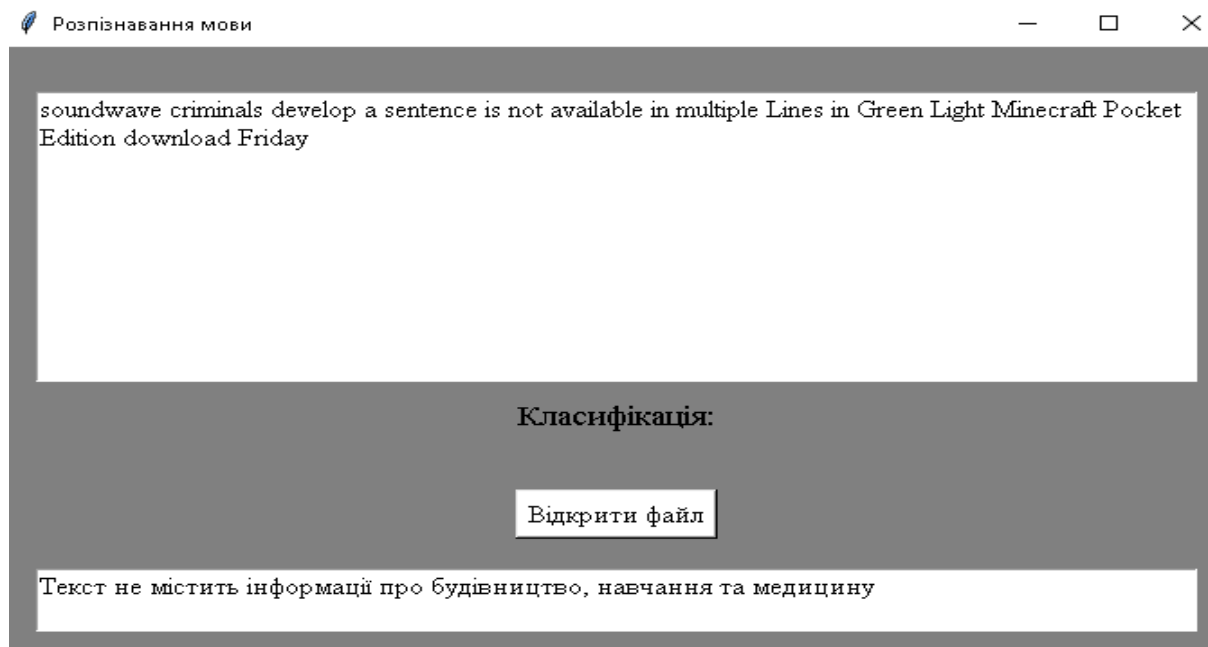


Рисунок 3.12 – Результат роботи програми з англійським аудіо

В результаті програма не розпізнала текст англійською мовою, проте це можливо реалізувати додавши мовний пакет англійської мови.

Для тесту на продуктивність обираємо відрізок тексту, довжина аудіозапису 1 хвилина.

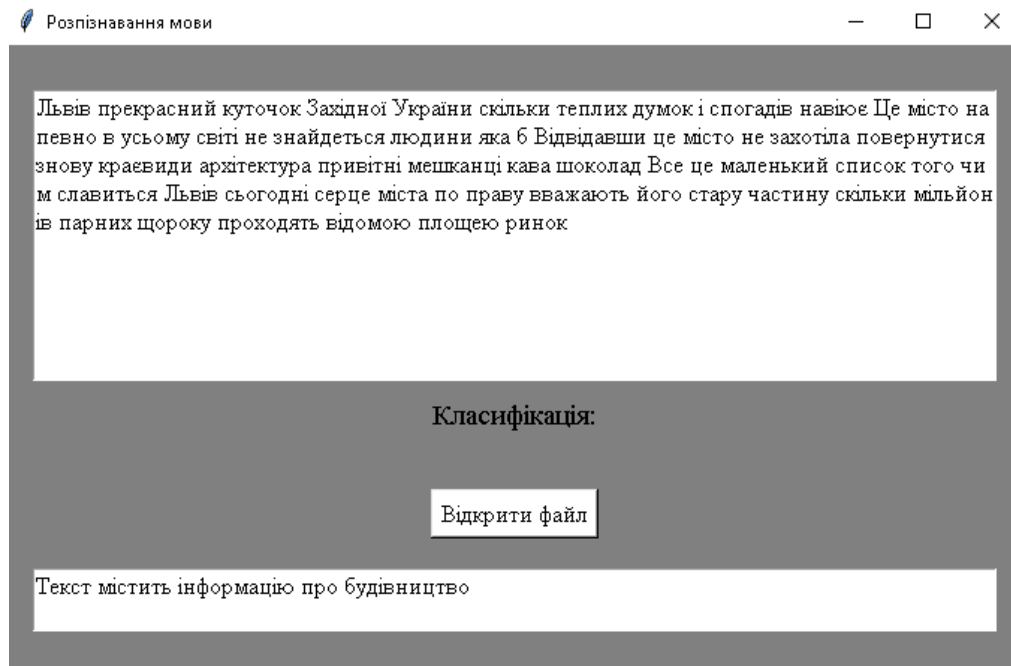


Рисунок 3.13 – Результат роботи з довгим текстом

Програма виконала свої функції та зчитала даний текст за 13 секунд.

Для наступного тесту ми додаємо шуми та змінюємо висоту голосу з такими налаштуваннями

Якісна озвучка тексту онлайн українською мовою

1 Оберіть голос для озвучки 2 Оберіть швидкість мови 3 Виберіть висоту голосу

Володимир 0.5 +1.8

Очистити Додати паузу Додати наголос

Результат роботи з зниженою швидкістю мови та шумами.

▶ КОНВЕРТУВАТИ ▶ ВІДТВОРИТИ ⬇️ ЗАВАНТАЖИТИ 📁 ЗБЕРЕГТИ

00:06

Рисунок 3.14 – Створення аудіозапису з шумами

В результаті маємо помилку в роботі програми, коли аудіозапис має дуже сильні шуми й одночасно дуже повільне мовлення.

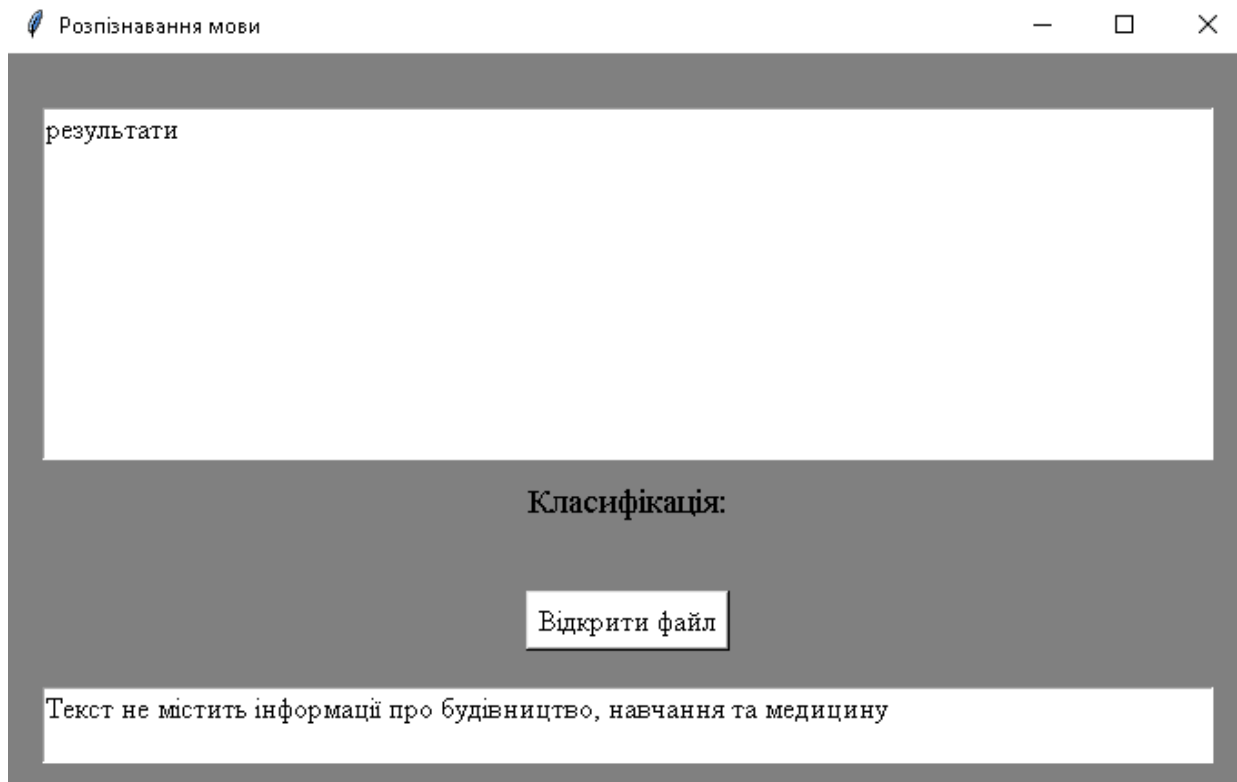


Рисунок 3.15 – Результат роботи програми з шумами

Програмний код виконує функцію розпізнавання мови з подальшою класифікацією тексту за ключовими словами. Також, під час тестування не було виявлено помилок у функціональності програми. Вона успішно розпізнає мову з аудіо-файлів та класифікує текст за ключовими словами, що дозволяє ефективно визначати тему тексту з питань будівництва, навчання або медицини.

На основі цих даних та результатів тестування можна зробити висновок, що програма виконує свою функцію розпізнавання мови та класифікації тексту з надійністю та точністю. Вона може бути корисним інструментом для автоматичної обробки та аналізу великих обсягів текстової інформації, зокрема для визначення тематики текстів з різних галузей.

Програмний код демонструє добру роботу та може бути використаний як основа для подальшого розвитку та вдосконалення програми.

### 3.5 Висновки до розділу 3

У розділі 3 було розглянуто програмну реалізацію інформаційної системи класифікації аудіозаписів. Була описана структура модулів системи та їх взаємозв'язок.

Особливості реалізації інформаційної системи були описані, включаючи використання бібліотеки SpeechRecognition для розпізнавання мови, використання tkinter для створення графічного інтерфейсу користувача, а також реалізацію класу TextClassifier для класифікації тексту за ключовими словами.

Були описані функціональні можливості інформаційної системи, включаючи відкриття аудіо-файлу, розпізнавання мови за допомогою Google Web Speech API, вивід розпізнаного тексту, класифікацію тексту за ключовими словами та вивід результату класифікації.

Система була протестована з використанням різних тестових сценаріїв, включаючи різні типи аудіо-файлів, неправильні вхідні дані та помилкові ситуації. Було проведено тестування функціональності, продуктивності, безпеки, масштабованості та сумісності системи.

На основі результатів тестування можна зробити висновок, що інформаційна система класифікації аудіозаписів працездатна і виконує свої функції належним чином. Вона здатна розпізнавати мову з аудіо-файлів, класифікувати текст за ключовими словами та надавати результат класифікації. Система також відповідає принципам безпеки програмного забезпечення і може працювати з великими обсягами даних без втрати продуктивності.

## Висновки

Кваліфікаційна робота бакалавра присвячена розробці та програмній реалізації методу неймережевого стенографування для класифікації аудіозаписів за ключовими словами, який дозволяє за аудіофайлом автоматизовано визначати його семантичну категорію.

В даній роботі було розглянуто проблему автоматизованого стенографування аудіозаписів та класифікації текстів за ключовими словами. Було проведений аналіз підходів до цих задач, а також огляд існуючих рішень.

Основним методом, який був розглянутий, є неймережеве стенографування для класифікації аудіозаписів за ключовими словами. Були розглянуті етапи виконання цього методу, від підготовки даних до проектування структури інформаційної системи та даталогічної моделі бази даних.

У розділі 3 була описана програмна реалізація інформаційної системи класифікації аудіозаписів. Була описана структура модулів системи та їх взаємозв'язок. Особливості реалізації були описані, включаючи використання бібліотек для розпізнавання мови та створення графічного інтерфейсу користувача. Система була протестована з використанням різних тестових сценаріїв, і результати показали, що вона працює належним чином і виконує свої функції. Вона здатна розпізнавати мову з аудіозаписів, класифікувати текст за ключовими словами і надавати результат класифікації. Система також відповідає принципам безпеки програмного забезпечення і може працювати з великими обсягами даних без втрати продуктивності.

Загальною метою роботи було розроблення ефективної інформаційної системи для автоматизованого стенографування аудіозаписів та класифікації текстів. За результатами роботи можна зробити висновок, що поставлені цілі були досягнуті, і система пропонує ефективне рішення для автоматизованого стенографування аудіозаписів і класифікації текстів за ключовими словами.

Вона може бути використана в різних сферах, де потрібно обробляти великі обсяги аудіо даних і визначати їх зміст.

## Перелік посилань

1. Wikipedia Viterbi algorithm [Електронний ресурс] – Режим доступу [https://en.wikipedia.org/wiki/Viterbi\\_algorithm](https://en.wikipedia.org/wiki/Viterbi_algorithm).
2. Wikipedia Bellman-Ford algorithm [Електронний ресурс] – Режим доступу [https://en.wikipedia.org/wiki/Bellman%E2%80%93Ford\\_algorithm](https://en.wikipedia.org/wiki/Bellman%E2%80%93Ford_algorithm).
3. MicrosoftSpeechRecognition [Електронний ресурс] – Режим доступу <https://support.microsoft.com/en-us/windows/use-voice-recognition-in-windows-83ff75bd-63eb-0b6c-18d4-6fae94050571>
4. Automatic Speech Recognition [Електронний ресурс] – Режим доступу <https://usabilitygeek.com/automatic-speech-recognition-asr-software-an-introduction/#:~:text=Automatic%20Speech%20Recognition%20or%20ASR,variations%20resembles%20normal%20human%20conversation.>
5. Google Cloud Speech API [Електронний ресурс] – Режим доступу <https://cloud.google.com/speech-to-text>.
6. Google Cloud Storage [Електронний ресурс] – Режим доступу <https://cloud.google.com/storage>
7. .HashDork Розпізнавання іменованих об'єктів (NER) – концепція, застосування та API [Електронний ресурс] – Режим доступу <https://hashdork.com/uk/визнання-іменованої-сутності/>
8. Rish, Irina. (2001). «An empirical study of the naive Bayes classifier». IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence с.41-47
9. Horn David, Siegelmann Hava "Support vector clustering" (2001) Journal of Machine Learning Research с.125–137
10. Klaus Greff; Rupesh Kumar Srivastava; Jan Koutník; Bas R. Steunebrink; Jürgen Schmidhuber (2015). «LSTM: A Search Space Odyssey» с.30-31
11. Wikipedia Fourier analysis [Електронний ресурс] – Режим доступу [https://en.wikipedia.org/wiki/Fourier\\_analysis](https://en.wikipedia.org/wiki/Fourier_analysis)
12. LibriSpeech [Електронний ресурс] – Режим доступу <https://paperswithcode.com/dataset/librispeech>

13. VoxCeleb [Электронный ресурс] – Режим доступа  
<https://www.tensorflow.org/datasets/catalog/voxceleb?hl=ru>
14. Mozilla. Commonvoice [Электронный ресурс] – Режим доступа  
<https://commonvoice.mozilla.org/en/datasets>
15. UrbanSound8K.Classification [Электронный ресурс] – Режим доступа  
<https://www.kaggle.com/code/prabhavsingh/urbansound8k-classification>
16. GitHub ESC-50 [Электронный ресурс] – Режим доступа  
<https://github.com/karolpiczak/ESC-50>
17. Python [Электронный ресурс] – Режим доступа <https://www.python.org/>
18. Tensorflow [Электронный ресурс] – Режим доступа  
<https://www.tensorflow.org/?hl=ru>
19. Keras. Головна сторінка [Электронный ресурс] – Режим доступа  
<https://keras.io/api/applications/vgg/>
20. PyPi Librosa [Электронный ресурс] – Режим доступа:  
<https://pypi.org/project/librosa/>
21. PyPi PyAudio [Электронный ресурс] – Режим доступа  
<https://pypi.org/project/PyAudio/>
22. MongoDB [Электронный ресурс] – Режим доступа  
<https://www.mongodb.com/community/forums/t/mongodb-4-4-19-is-released/215463>
23. MySQL [Электронный ресурс] – Режим доступа  
<https://www.mysql.com/>
24. Django MeetDjango [Электронный ресурс] – Режим доступа  
<https://www.djangoproject.com/>
25. PyPi SpeechRecognition [Электронный ресурс] – Режим доступа  
<https://pypi.org/project/SpeechRecognition/>
26. Narakeet. TextToVoice [Электронный ресурс] – Режим доступа  
<https://www.narakeet.com/app/text-to-audio>
27. Zvukogram TextToEnglish [Электронный ресурс] – Режим доступа:  
<https://zvukogram.com/speech/tts-english/>

# ДОДАТКИ

## Додаток А

### Програмні коди

```

import speech_recognition as sr
import tkinter as tk
from tkinter import filedialog, messagebox

# Класифікація тексту за ключовими словами
1 usage
class TextClassifier:
    def __init__(self):
        # Список ключових слів для класифікації з питань будівництва
        self.construction_keywords = ['будівництво', 'будівля', 'будівельний', 'архітектура', 'планування',
                                     'конструкція',
                                     'фундамент', 'цегла', 'бетон', 'стіна', 'покрівля', 'вікна', 'двері', 'драбина']

        # Список ключових слів для класифікації з питань навчання
        self.education_keywords = ['навчання', 'освіта', 'вчителі', 'студент', 'школа', 'університет', 'викладання',
                                   'дисципліна', 'екзамени', 'знання', 'педагогіка', 'лектор', 'курс', 'інформація']

        # Список ключових слів для класифікації з питань медицини
        self.medical_keywords = ['медицина', 'лікар', 'хвороба', 'лікарня', 'терапія', 'діагностика', 'здоров\''я',
                                 'операція', 'ліки', 'вакцинація', 'інфекція', 'реабілітація', 'аналіз', 'симптом']

# Функція класифікації тексту
3 usages
def classify_text(self, text, topic):
    # Перетворення тексту в малі літери та розділення на окремі слова
    words = text.lower().split()

    # Перевірка наявності ключових слів в тексті відповідно до теми
    if topic == 'construction':
        keywords = self.construction_keywords
    elif topic == 'education':
        keywords = self.education_keywords
    elif topic == 'medical':
        keywords = self.medical_keywords
    else:
        return False

    for keyword in keywords:
        if keyword in words:
            return True

```

```

return False

# Функція, которая будет вызываться при нажатии кнопки "Открыть файл"
1 usage
def open_file():
    # Открываем диалоговое окно для выбора аудио-файла
    file_path = filedialog.askopenfilename()

    # Проверяем, что пользователь выбрал файл
    if file_path:
        try:
            # Создание объекта Recognizer для использования распознавания речи
            r = sr.Recognizer()

            # Открытие аудио-файла для распознавания
            with sr.AudioFile(file_path) as source:
                # Считывание аудио-файла и преобразование в объект AudioData
                audio_data = r.record(source)

                # Распознавание речи с использованием Google Web Speech API
                text = r.recognize_google(audio_data, language='uk-UA')

            # Вывод распознанного текста
            text_output.config(state=tk.NORMAL)
            text_output.delete(1.0, tk.END)
            text_output.insert(tk.END, text)
            text_output.config(state=tk.DISABLED)

            # Класифікація тексту
            classifier = TextClassifier()
            is_about_construction = classifier.classify_text(text, 'construction')
            is_about_education = classifier.classify_text(text, 'education')
            is_about_medical = classifier.classify_text(text, 'medical')

            # Вывод результату класифікації
            classification_output.config(state=tk.NORMAL)
            classification_output.delete(1.0, tk.END)
            if is_about_construction:
                classification_output.insert(tk.END, "Текст містить інформацію про будівництво\n")
            if is about education:

```

```

        classification_output.insert(tk.END, "Текст містить інформацію про навчання\n")
    if is_about_medical:
        classification_output.insert(tk.END, "Текст містить інформацію про медицину\n")
    if not is_about_construction and not is_about_education and not is_about_medical:
        classification_output.insert(tk.END,
                                     "Текст не містить інформації про будівництво, навчання та медицину\n")
    classification_output.config(state=tk.DISABLED)

except Exception as e:
    messagebox.showerror('Помилка', f'Помилка при розпізнаванні мови: {str(e)}')

# Створення головного вікна
root = tk.Tk()
root.title('Розпізнавання мови')

# Встановлення фоновго кольору
root.configure(bg='grey')

# Створення фрейму для віджетів
frame = tk.Frame(root, padx=20, pady=20, bg='grey')
frame.pack()

# Створення віджету розпізаного тексту
text_output = tk.Text(frame, height=10, state=tk.DISABLED, font=('Times New Roman', 12))
text_output.pack(pady=10)

# Змінна для збереження тексту класифікації
classification_text = tk.StringVar()

# Створення віджету відображення класифікації тексту
classification_output = tk.Label(frame, text='Класифікація:', font=('Times New Roman', 14), bg='grey')
classification_output.pack()

# Створення віджету відображення назви теми
topic_output = tk.Label(frame, font=('Times New Roman', 12), bg='grey')
topic_output.pack()

# Створення кнопки відкрити файл
open_button = tk.Button(frame, text='Відкрити файл', font=('Times New Roman', 12), command=open_file, bg='#FFFFFF')
open_button.pack(pady=10)
# Створення віджету відображення класифікації тексту
classification_output = tk.Text(frame, height=2, state=tk.DISABLED, font=('Times New Roman', 12))
classification_output.pack(pady=10)

# Вирівнювання тексту по центру
text_output.tag_configure('center', justify='center')
text_output.insert(tk.END, 'Розпізнаний текст', 'center')

classification_output.tag_configure('center', justify='center')
classification_output.insert(tk.END, 'Класифікація', 'center')

topic_output.configure(justify='center')

# Запуск головного циклу програми
root.mainloop()

```

## Додаток Б

### Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

## МЕТОД НЕЙРОМЕРЕЖЕВОГО СТЕНОГРАФУВАННЯ ДЛЯ КЛАСИФІКАЦІЇ АУДІОЗАПИСІВ ЗА КЛЮЧОВИМИ СЛОВАМИ



**Виконав:**

*студент 4 курсу, групи КН-19-1*

Жайворон Дмитро Олегович

**Керівник:**

*к.п.н., доцент кафедри КН*

Петровський Сергій Степанович



## Актуальність

- **Об'єм аудіоданих:** За останні роки обсяг аудіоданих значно зріс завдяки широкому поширенню веб-контенту, стрімінгових платформ, подкастів та інших медіаформатів. Класифікація аудіозаписів за ключовими словами дозволяє розрізняти та організувати цей великий обсяг даних для полегшення пошуку та отримання потрібної інформації.
- **Покращення пошуку:** Застосування методів нейромережевого стенографування дозволяє автоматично присвоювати ключові слова аудіозаписам на основі їх вмісту. Це спрощує процес пошуку, дозволяючи користувачам швидко знаходити відповідні аудіозаписи за ключовими словами, що покращує їх взаємодію з великими обсягами даних.
- **Потоковий аналіз:** Нейромережевий стенографічний метод може бути використаний для аналізу аудіозаписів в реальному часі, що має важливе значення для таких сфер, як моніторинг медіа, медіааналітика, обробка мовлення та інші. Завдяки класифікації аудіозаписів за ключовими словами можна швидко і ефективно відстежувати певні теми, події або вимоги користувачів у реальному часі.



## Мета і задачі роботи

**Метою кваліфікаційної роботи бакалавра** є розробка та програмна реалізація методу неймережевого стенографування для класифікації аудіозаписів за ключовими словами:

1. розробити метод неймережевого стенографування для класифікації аудіозаписів за ключовими словами;
2. розробити архітектуру нейронної мережі для стенографування аудіозаписів;
3. спроектувати структуру інформаційної системи класифікації аудіозаписів, що використовує метод неймережевого стенографування для класифікації аудіозаписів за ключовими словами, й структуру відповідної БД;
4. виконати вибір засобів розробки інформаційної системи ідентифікації особистості;
5. виконати програмну реалізацію інформаційної системи класифікації аудіозаписів;
6. провести тестування розробленої інформаційної системи.

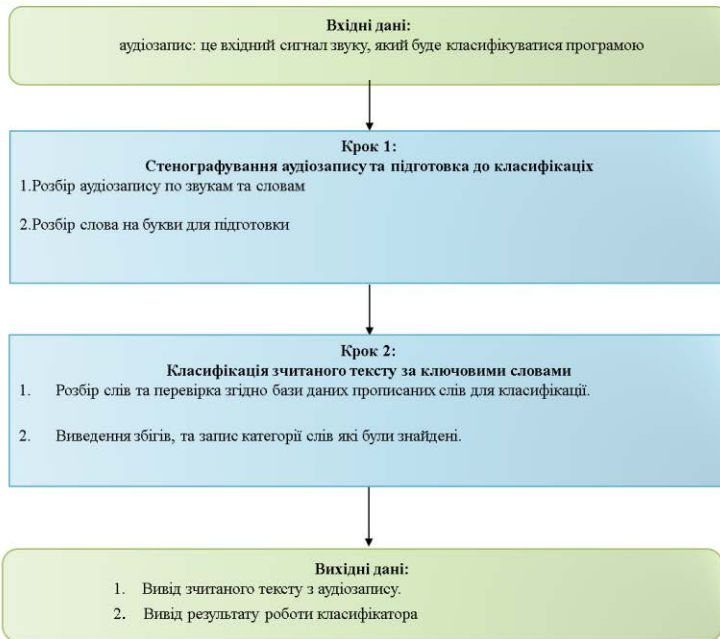
## Структура модулів інформаційної системи класифікації аудіозаписів створеної в рамках роботи над кваліфікаційною роботою:

**Головний модуль:** цей модуль відповідає за запуск інформаційної системи та взаємодію з користувачем. Містить функції, які відповідають на події взаємодії користувача, наприклад, натискання кнопки "Відкрити файл". Викликає функції з модуля класифікатора для аналізу та класифікації аудіозапису. Цей модуль є основою всієї системи і керує виконанням інших модулів.

**Модуль розпізнавання мови (speech\_recognition):** цей модуль залежить від сторонньої бібліотеки `speech_recognition`, яка використовується для розпізнавання мовлення з аудіо-файлу. Використовується для створення об'єкту `Recognizer` і `AudioFile`, а також для запису аудіо-файлу та його розпізнавання.

**Модуль класифікатора (TextClassifier):** цей модуль містить клас `TextClassifier`, який відповідає за класифікацію тексту згідно зазначеними ключовими словами. Цей модуль дозволяє визначати тему аудіозапису з використанням ключових слів.

**Модуль графічного інтерфейсу (tkinter):** цей модуль залежить від стандартної бібліотеки `tkinter`, яка використовується для створення графічного інтерфейсу користувача. Цей модуль дозволяє користувачеві взаємодіяти з системою шляхом натискання кнопок, введення тексту



## Схема етапів роботи програми

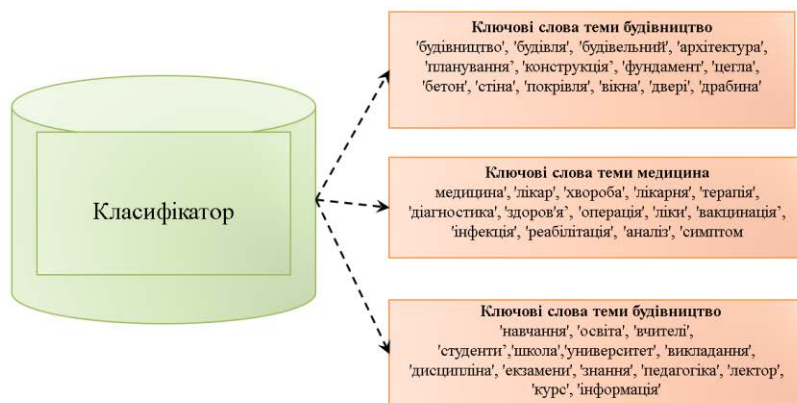


## Вибір засобів розробки інформаційної системи класифікації аудіозаписів

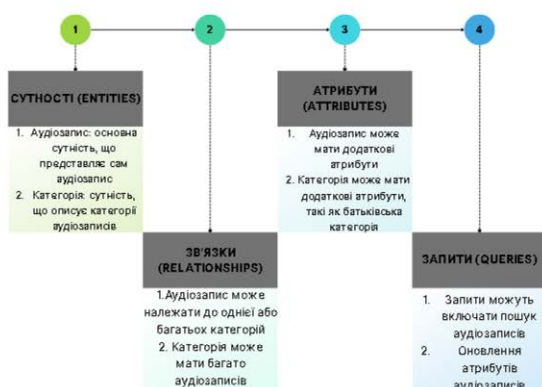
- **Speech recognizer** використовується в інформаційних системах для стенографування аудіозаписів, оскільки вона дозволяє замінити ручний процес стенографування, що може займати багато часу і бути неточним. Замість цього, система може автоматично розпізнавати мовлення на аудіозаписах і перетворювати його на текст, що значно збільшує швидкість та точність процесу.
- **Python** - це один з найкращих варіантів для розробки інформаційних систем класифікації аудіозаписів. Він має багато переваг, таких як простота та лаконічність коду, багато бібліотек для роботи з даними, підтримку машинного навчання та паралельної обробки даних. Можна використовувати для роботи з неймережами в Python, належать Keras, Theano та PyTorch.
- **PyCharm** є середовищем розробки для Python, яке надає широкий набір функцій, таких як автоматичне доповнення коду, візуальне налагодження, інтеграцію з інструментами управління версіями, підтримку віртуальних середовищ та багато іншого, що сприяє підвищенню продуктивності та зручності розробки



## Робота модуля класифікатора з набором ключових слів



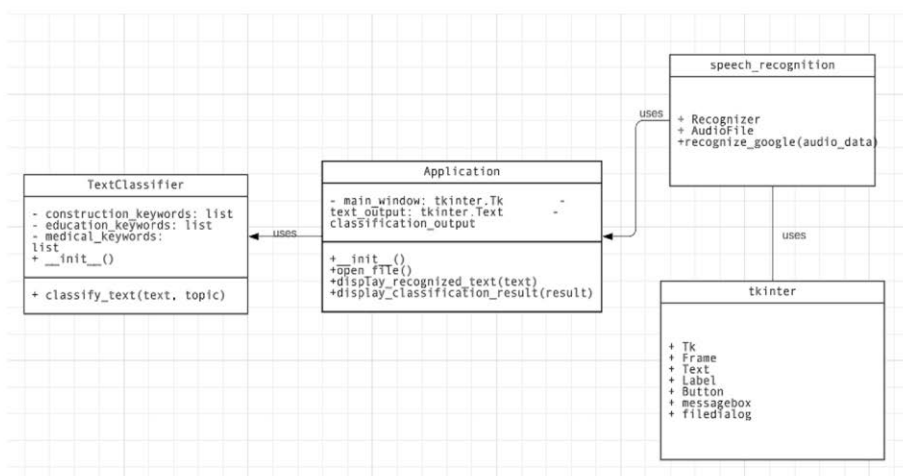
## Схематичне зображення даталогічної моделі бази даних



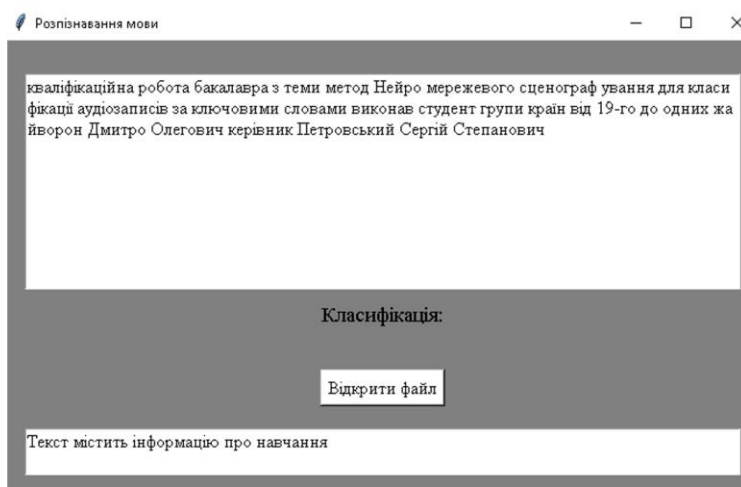
Ця даталогічна модель бази даних відображає основні сутності, зв'язки та атрибути, які були використані в інформаційній системі класифікації аудіозаписів.



## Діаграма класів програмної реалізації методу нейромережевого стенографування для класифікації аудіозаписів за ключовими словами



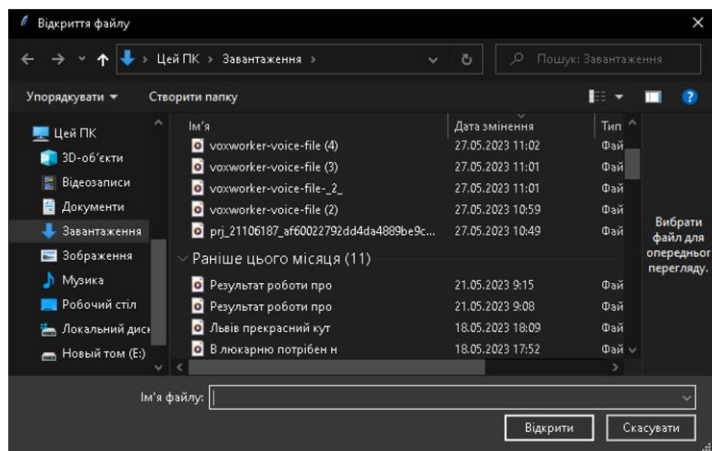
## Програмна реалізація методу нейромережевого стенографування для класифікації аудіозаписів за ключовими словами



Вигляд інтерфейсу та зчитаного аудіофайлу з класифікацією за ключовими словами



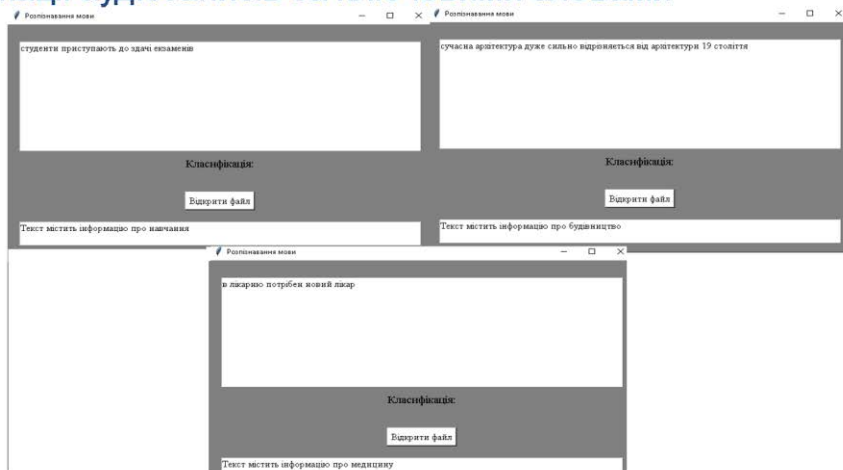
## Програмна реалізація методу нейромережевого стенографування для класифікації аудіозаписів за ключовими словами



Форма вибору аудіофайлу для стенографува



## Програмна реалізація методу нейромережевого стенографування для класифікації аудіозаписів за ключовими словами



Результати класифікації текстів з різних тем



## Висновки

Кваліфікаційна робота бакалавра присвячена розробці та програмній реалізації методу нейромережевого стенографування для класифікації аудіозаписів за ключовими словами, який дозволяє за аудіофайлом автоматизовано визначати його семантичну категорію.

В даній роботі було розглянуто проблему автоматизованого стенографування аудіозаписів та класифікації текстів за ключовими словами. Було проведений аналіз підходів до цих задач, а також огляд існуючих рішень.

Основним методом, який був розглянутий, є нейромережеве стенографування для класифікації аудіозаписів за ключовими словами. Були розглянуті етапи виконання цього методу, від підготовки даних до проєктування структури інформаційної системи та даталогічної моделі бази даних.



ДЯКУЮ ЗА УВАГУ!



Ім'я користувача:  
Кафедра КН

ID перевірки:  
1015412357

Дата перевірки:  
04.06.2023 15:00:25 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
04.06.2023 15:05:20 EEST

ID користувача:  
100005671

Назва документа: КН-19-1 Жайворон

Кількість сторінок: 60 Кількість слів: 11491 Кількість символів: 91221 Розмір файлу: 990.44 KB ID файлу: 1015075501

## 3.06% Схожість

Найбільша схожість: 0.96% з джерелом з Бібліотеки (ID файлу: 1015072628)

2.35% Джерела з Інтернету

334

Сторінка 62

1.93% Джерела з Бібліотеки

117

Сторінка 64

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

1

## Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 10%

ID: 114641 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА Додано в БД: 2023-06-04 Автора: Д.О. Жайворон Керівники: С.С.Петровський Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	80310	1230	1625 (2%)	25 (2%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами

Автор: студент групи КН-19-1 Жайворон Дмитро Олександрович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.н.н., доцент Петровський С.С.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<i>відповідає</i>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданій поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданій поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укріплення запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

*Підтвердження:*

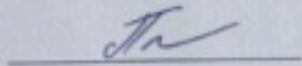
*Запозичення, виявлені в роботі Жайворон Д.О., не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти програмного коду, що не мають авторства і містять поширені конструкції; серед запозичень знаходяться загальновідомі терміни, скорочення та матеріали статей.*

*Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості, складає:*

*- за системою Anti-Plagiarism: 1%;*

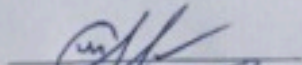
*- за системою Unichек: 3.06 %, є допустимими запозиченнями які відносяться до описаних вище.*

Керівник роботи



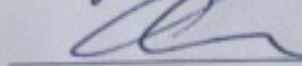
*Сергій ПЕТРОВСЬКИЙ*

Гарант ОП



*Олександр МАЗУРЕЦЬ*

Завідувач кафедри КН



*Олександр БАРМАК*



## ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу бакалавра

студента гр. КН-19-1 Жайворон Дмитро Олегович

за темою Метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами

### 1. Актуальність теми

Актуальним завданням, яке потребує аналізу і досліджується у даній роботі, є методи нейромережевого стенографування для класифікації аудіозаписів за ключовими словами та огляд існуючих рішень. Для ефективного використання автоматизованих систем стенографування аудіозаписів, необхідно передбачити застосування програмного модуля, який би дозволяв класифікацію аудіозаписів за ключовими словами. Розробка такого методу є актуальною задачею комп'ютерних наук.

### 2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

За стандартом, а саме описом предметної області, об'єктами вивчення та діяльності є математичні, інформаційні, імітаційні моделі реальних явищ, об'єктів, систем і процесів та методи і технології отримання, зберігання, обробки, передачі та використання інформації. Метою роботи саме є розробка методу нейромережевого стенографування для класифікації аудіозаписів за ключовими словами та розробка інформаційної системи реалізації вказаного методу. При вирішенні поставленої задачі використано математичні моделі, методи та алгоритми розв'язання теоретичних і прикладних задач, що виникають при розробці інформаційних технологій. Тому результати виконання кваліфікаційної роботи бакалавра відповідають стандарту бакалавра спеціальності 122 – Комп'ютерні науки.

### 3. Професійні та особистісні якості бакалавра

При роботі над кваліфікаційною роботою бакалавра Жайворон Дмитро Олегович проявив себе як кваліфікований фахівець та дисциплінований студент, вчасно виконуючи поставлені етапи дослідження. Як в процесі написання пояснювальної записки, так і при розробці прикладного програмного забезпечення проявив достатні для одержання успішного результату компетентності та результати навчання. Опанував професійні скіли за напрямком «Комп'ютерні науки» та достатньо значний софт скіл.

#### **4. Ступінь самостійності під час виконання кваліфікаційної роботи**

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував всі поставлені задачі.

#### **5. Ступінь оволодіння методами дослідження**

При реалізації кваліфікаційної роботи показав достатній рівень компетентностей та володіння необхідними інструментами та обладнанням, методами, методиками та технологіями предметної області комп'ютерних наук.

#### **6. Повнота та якість розкриття теми роботи**

Тема роботи в повній мірі обґрунтована й розкрита, проведено аналіз актуальності та відомих досліджень в межах обраної теми, поставлені завдання, які у роботі виконані, та розроблено програмне забезпечення для валідації та верифікації запропонованого метода.

**7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу**

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.


**8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин**

Розроблений у роботі метод та його програмна реалізація може бути використана у медичних і військових застосуваннях, системах безпеки для підвищення ефективності систем розпізнавання та ідентифікації аудіозаписів.

**9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота**

Враховуючи достатній рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «задовільно».

Керівник \_\_\_\_\_



к.п.н., доцент каф. КН Сергій ПЕТРОВСЬКИЙ



## РЕЦЕНЗІЯ

### на кваліфікаційну роботу бакалавра

студента гр. КН-19-1 Жайворон Дмитро Олегович

за темою: Метод нейромережевого стенографування для класифікації аудіозаписів за ключовими словами

#### 1. Актуальність обраної теми

Актуальністю дипломної роботи можна вважати те, що на даний момент в комп'ютерних технологіях задачі розпізнавання та розуміння контексту мови є дуже актуальними, тому що це може полегшити спосіб спілкування між людиною та комп'ютером, ці технології використовуються в медичних і військових застосуваннях, системах безпеки, автоматизованих системах розпізнавання та ідентифікації

#### 2. Повнота розкриття мети та завдань роботи

У процесі виконання кваліфікаційної роботи були розкриті мета та завдання. Автором проведено аналіз процесу стенографування та класифікації текстів. Проведено систематизацію основних принципів та методів, що використовуються в задачах стенографування і класифікації. Загалом визначені в роботі завдання є повними та збалансованими, а мета роботи повністю розкритою.

#### 3. Зміст кожного розділу роботи

У першому розділі кваліфікаційної роботи досліджено особливості застосування нейромереж та підходів для виконання завдання й досягнення мети роботи. У другому розділі описує методи нейромережевого стенографування аудіозаписів та класифікацію їх за ключовими словами. В третьому розділі описується процес розробки програми та тестування готового результату виконання роботи.

#### 4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблений модуль програмного забезпечення забезпечує високу точність розпізнавання тексту аудіозаписів та їх подальшу класифікацію згідно тем котрі описані в роботі. Розроблений модуль має потенціал для розширення та вдосконалення наприклад, додавання ширшої класифікації та додавання розпізнавання на іноземних мовах.

#### 5. Якість оформлення кваліфікаційної роботи бакалавра

Подана кваліфікаційна робота має задовільний рівень якості оформлення і висвітлення матеріалу. Автор вдало засвоїв вимоги до структури, стилю та оформлення науково-дослідницької роботи, як наслідок заголовки розділів є чіткими та зрозумілими, а структура роботи є логічно впорядкованою.

#### 6. Недоліки кваліфікаційної роботи бакалавра

Робота має недоліки. Записка містить певну кількість орфографічних помилок, які впливають на читабельність. Автор подав недостатньо посилань на авторитетні джерела. Недостатньо описані способи розробки.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «задовільно».

Рецензент Фук к.т.н., доцент каф. АКИТ та Р  
С.О. Корницька