Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет комп'ютерних систем і автоматики
(повне найменування інституту, факультету)

Кафедра комп'ютерних систем управління
(повна назва кафедри)

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:

Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 1. Підсистема параметризації.

Виконав: студентка 2-го курсу, групи 2АКІТ-20м
спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва спеціальності)

Людмила Дихніч
(ім'я та прізвище)

Керівник: д.т.н., професор каф. КСУ

Вячеслав Ковтун
(ім'я та прізвище)
«____» _____ 2021 р.

Опонент: к.т.н., доцент каф. АІІТ

Володимир Гармаш
(ім'я та прізвище)
«____» _____ 2021 р.

Допущено до захисту
Завідувач кафедри КСУ
д.т.н., проф.
Володимир Дубовой
(ім'я та прізвище)
«____» _____ 2021 р.

Вінниця ВНТУ – 2021 рік

Вінницький національний технічний університет
Факультет комп’ютерних систем і автоматики
Кафедра комп’ютерних систем управління
Рівень вищої освіти ІІ-й (магістерський)
Галузь знань – 15 Автоматизація та приладобудування
Спеціальність – 151 Автоматизація та комп'ютерно-інтегровані технології
Освітньо-професійна програма Інтелектуальні комп’ютерні системи

ЗАТВЕРДЖУЮ
Завідувач кафедри КСУ
д.т.н., проф.
_____Володимир Дубовой
«01» 10 2021 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Дихніч Людмилі Дмитрівні

1. Тема роботи: Розробка застосунку для голосового управління типовими операціями.системи планування ресурсів підприємства. Ч. 1. Підсистема параметризації.

Керівник роботи: _д.т.н. проф. Ковтун В.В.
Затверджені наказом ВНТУ від «_24__»____09____ 2021 року №277
2. Строк подання студентом роботи: «_10_» ____12____2021 р.

3. Вихідні дані до роботи: експлуатаційні дані з об’єкту дослідження, сучасна програмна архітектура, мова програмування *Python*.

4. Зміст розрахунково-пояснювальної записки: огляд предметної області; розробка математичного апарату; розробка програмного забезпечення та експериментальні дослідження.

5. Перелік графічного матеріалу: архітектура системи; робочий процес системи; приклад вихідних даних; інтерфейс користувача розробленої системи; лістинг програмного забезпечення.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання отримав
4	доцент кафедри ЕПВМ, доцент, к.е.н. Кавецький В.В.		

7. Дата видачі завдання: «__01_»_____10_____ 2021р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд предметної області	04.09.21р.	
2	Розробка математичного апарату	22.09.21р.	
3	Розробка програмного забезпечення та експериментальні дослідження	30.10.21р.	
4	Підготовка економічної частини	12.11.21р.	
5	Оформлення пояснювальної записки і графічного матеріалу	08.12.21р.	
6	Попередній захист роботи	17.12.21р.	
7	Остаточний захист роботи	22.12.21р.	

Студент _____ Людмила Дихніч
(підпис) (ім'я та прізвище)

Керівник роботи _____ В'ячеслав Ковтун
(підпис) (ім'я та прізвище)

АНОТАЦІЯ

Магістерська робота присвячена пошуку процесу аналізу акустичних даних та розпізнавання акустичної інформації. Проаналізовано представлення та розпізнавання акустичної інформації та визначено основні компоненти системи автоматичного розпізнавання акустичної інформації (команд):

- Попередня обробка звукових сигналів;
- Перетворення сигналу у вектор ознак;
- Розпізнавати звукову інформацію (класифікація).

Розроблено та розглянуто метод попередньої обробки та виділення ознак мовленнєвого сигналу, серед якого обрано один із найбільш популярних та корисних методів, який базується на знаходженні коефіцієнта збереження Мела (MFCC). Враховуючи метод акустичного розпізнавання інформації, обрано метод динамічного програмування. В описі цих методів наведено їх короткий опис, класифікацію, завдання, які вони вирішують (попередня обробка та розпізнавання), алгоритми побудови систем розпізнавання на основі цих методів та їх застосування.

Розроблено програмний комплекс, що дозволяє створити базу даних голосових команд і включає реалізацію всіх вищезгаданих методів і алгоритмів аналізу та розпізнавання акустичної інформації. Розглянута модель розпізнавання голосових команд має на меті створення мовного інтерфейсу, який дозволить істотно підвищити ефективність роботи людино-машинної системи.

.

ABSTRACT

The master's work is devoted to finding a process of acoustic data analysis and recognition of acoustic information. Analyzed the representation and recognition of acoustic information and identified the main components of the system of automatic recognition of acoustic information (commands):

- Front-end processing of acoustic signals;
- Transformation of the signal into a vector of features;
- Recognize audio information (classification).

Developed and reviewed the method of pre-processing and identification of signs of speech signal, including one of the most popular and useful methods, which is based on the discovery of the coefficient of conservation of Mel (MFCC). Taking into account the method of acoustic information retrieval, the method of dynamical programming is inverted. The description of these methods includes their short description, classification, tasks they perform (pre-processing and recognition), algorithms for building recognition systems based on these methods and their application.

Developed software system, which allows the creation of a database of voice commands and includes the implementation of all of the above methods and algorithms for analysis and recognition of acoustic information. This model of recognition of voice commands is aimed at creating a language interface, which will significantly improve the efficiency of the man-machine system.

ЗМІСТ

ВСТУП

З моменту появи першого комп’ютера одним з найважливіших питань розвитку комп’ютерної техніки став процес взаємодії людини та комп’ютера. Довгий час він підходив лише для вузьких техніків через посередника-програміста «спілкується» з машиною. Така ситуація триває до появи діалогового інтерфейсу, коли користувач може особисто ввести команду, надіслану машині з клавіатури, і отримати змістовну відповідь. Поява графічних інтерфейсів позбавила людини знання будь-яких команд, що призвело до широкого використання персональних комп’ютерів. Однак люди завжди шукають більш універсальний і природний спосіб взаємодії з комп’ютерами. Навіть в епоху перфокарт у науковій фантастиці люди однаково спілкуються з комп’ютерами. У той же час було зроблено перший крок до реалізації мовного інтерфейсу.

Однак, якщо порівняти продуктивність сучасних систем розпізнавання з продуктивністю системи на початку цієї галузі науки, можна сказати, що дослідники досягли великих успіхів за останні кілька десятиліть. Це змушує деяких експертів сумніватися в можливості впровадження мовних інтерфейсів найближчим часом. Інші вважають, що проблема майже вирішена. Більшість експертів погоджуються, що для розвитку розпізнавання мови потрібен певний час [1]. В останні роки інформаційні технології швидко розвиваються. Одним із пріоритетних напрямків досліджень у цій галузі є завдання зберігання, обробки та передачі мультимедійних даних. На жаль, поки що комп'ютери не змогли повністю замінити фахівців з багатьох завдань аналізу мультимедійних даних. Ці завдання включають синхронний переклад, автоматичну сегментацію зображень і відеорядів, а також автоматичне скорочення. Одним з основних завдань обробки мультимедійної інформації є розпізнавання та аналіз природної мови людини.

Завдання мовного аналізу включають широкий спектр завдань. Традиційно вони поділяються на три підкатегорії: завдання на розпізнавання,

завдання на класифікацію та завдання на діагностику. Завдання на розпізнавання включає завдання перевірки та розпізнавання мовця. Завдання класифікації включають завдання на розпізнавання ключових слів, комбіновані завдання на розпізнавання мовлення та завдання семантичного аналізу мови. До категорії діагностичного завдання входять завдання на визначення психофізичного стану мовця. За останні роки у вищезгаданій роботі було досягнуто значних успіхів. Наприклад, завдяки високій точності розробленого методу алгоритми ідентифікації або верифікації мовця широко використовуються в криміналістичних процедурах або описі прав доступу. Завдання розпізнавання мови залишається актуальним. Спектр отриманих рішень досить широкий: автоматична стенографія, автоматичний довідковий термінал з голосовим керуванням, синхронний переклад, якісна система стиснення та передачі голосового сигналу, сегментація мультимедійної інформації, система індексування та пошуку.

Мова є найбільш природною формою людського спілкування, тому реалізація інтерфейсів на основі аналізу голосової інформації є перспективним напрямком для розвитку інтелектуальних систем управління. Однією з невирішених проблем у сфері інформаційно-вимірювальних систем є побудова системи автоматичного розпізнавання мовних сигналів, на які не впливає мовець. Його рішення дозволить розширити коло користувачів таких систем та значно підвищити ефективність обміну інформацією в людино-машинних системах. Реалізація мовного інтерфейсу є дуже складною технічною проблемою, і її вирішення є перетином багатьох наукових галузей. Тому у сприйнятті мови людина використовує механізм асоціативного аналізу не тільки для розкладання і порівняння почутих звуків, а й для збору фонем у мові. Зображення це не тільки вибрати найбільш підходяще слово за звуком. Подібність також відображається в інтонації, емоційному забарвленні, контексті слів, фразах, реченнях і навіть у всьому тексті. Тому, навіть якщо інформації дуже мало, людина може розпізнати мову. Наприклад, коли людина слухає незнайомий їй текст іноземною мовою,

вимоги до якості звуку значно вищі, ніж коли вона сприймає рідну мову. Зараз не час для безпосереднього впровадження мовних інтерфейсів у повсякденне життя кінцевих користувачів, але поточний прогрес важко оцінити. Програми та системи з методами мовного введення набувають все більшого поширення, але з огляду на всі їхні недоліки варто розглянути перспективи розвитку вузькоспеціалізованих систем із зрозумілими додатками [2]:

-Система контролю присутності персоналу;

-Перевірка особи та контроль доступу;

- Телефонний банкінг;

-Інформаційний голосовий введення замість тексту;

-Автоматизована система заповнення анкет та шаблонних інформаційних форм;

-Здійснювати біометричну реєстрацію в різних багатосторонніх телефонних системах;

-Розробка інформаційно-довідкових сервісів різного призначення, в яких клієнти запитують інформацію, дані, що цікавлять його, та отримують інформацію на мовній чи іншій формі; телефонні лінії підтримки клієнтів, електронна комерція;

-Управління системою життєзабезпечення інвалідів та будівництвом системи інтелектуального житла, так званого «розумного дому»[3];

-Управління освітленням, водопостачанням, опаленням, кондиціонуванням тощо;

-Створити окрему систему автоматичного перекладу з однієї мови на іншу мову, яка працює в режимі реального часу;

-Судово-медичні експертизи, особливо системи, що використовуються для відтворення спотворених і шумних голосових повідомлень;

-У майбутньому - пошук в інформаційній мережі лінгвістичної інформації про дане ключове слово чи питання.

Об’єктом дослідження є процес розпізнавання та керування голосовими командами комп’ютерною системою.

Предмет дослідження – метод розпізнавання голосових команд, націлене на керування комп’ютерною системою, на основі методу динамічного програмування.

Мета: прискорити процес розпізнавання голосових команд за допомогою методу динамічного програмування та розробити більш точну систему розпізнавання голосових команд на основі додаткового аналізу.

Практичне значення отриманих у роботі результатів полягає в розробленні вдосконаленого методу для прискорення процесу розпізнавання та керування голосовими командами для комп’ютерних систем.

Новизна роботи полягає в тому, що запропонована система може бути використана в різних сферах для полегшення роботи за комп’ютером. Програмна реалізація системи розпізнавання голосових команд, розроблена в цій роботі, може бути використана для створення розумних будинків або автомобілів.

Апробація. Представлені в роботі результати апробовані в результаті участі в конференції Всеукраїнська науково-практична Інтернет-конференція студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2022)»:

Публікації: Ірина Олександрівна Майданевич, Людмила Дмитрівна Дихніч «Розробка erp-застосунку для голосового управління типовими операціями», ВНПІК САМН «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ», 2021. URL: https://conferences.vntu.edu.ua/index.php/mn/mn2022/schedConf/presentations

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ДОВЕДЕННЯ РАЦІОНАЛЬНОСТІ ТЕМИ МАГІСТЕРСЬКОЇ РОБОТИ

1.1 Класифікація актуальних комп’ютерних систем

Як говорить Вікіпедія: «Планува́ння ресу́рсів підприє́мства (ERP-система) (англ. Enterprise Resource Planning System — Система планування ресурсів підприємства) — корпоративна інформаційна система (КІС), призначена для автоматизації обліку й керування. Зазвичай ERP-системи будуються за модульним принципом і в тому або іншому ступені охоплюють всі ключові процеси діяльності компанії». Відштовхуючись від цього визначення далі досліджуватимемо спрощені версії таких систем, орієнтовані на приватне використання (наприклад, системи класу «розумний дім»).

Автоматичне розпізнавання мовлення – це процес автоматичного перекладу людської мови комп’ютера чи іншої машини в текстовий формат. Хоча вираз цієї проблеми настільки простий, дослідники з усього світу намагаються вирішити цю проблему вже більше півстоліття. Сама людська мова є дуже складним об’єктом, на неї впливають освіта, фізичні умови та психологія людини, діалекти чи особливості людської вимови, контекст вимови мови та особливості самої мови, включаючи вплив таких факторів, як фонетика. , фонологія та граматика. Крім того, аналіз голосових сигналів повинен враховувати зовнішні фактори, такі як фоновий шум, приміщення, відстань від пристрою захоплення голосу, зміни в каналах зв’язку тощо. У зв'язку з цим вивчення мови та її автоматичне розпізнавання є міждисциплінарним процесом, що вимагає знань у таких галузях [4]:

- Цифрова обробка сигналу, потрібно витягти з сигналу найбільш корисну інформацію з урахуванням умов прийнятого сигналу;

- Фізика, щоб з'ясувати взаємозв'язок між прийнятим цифровим сигналом і процесом сприйняття голосу та голосу;

- Фонетика і фонологія, що описують звукову систему даної мови та співвідношення між цими звуками;

- Мовознавство, вміє розуміти зв’язок між звуками і словами, їх значення та структуру в реченнях;

- Теорія інформації, що дозволяє оцінити наявність мови в сигналах - представлений у компактному кодованому вигляді для полегшення подальшої обробки;

- Розпізнавання образів, область науки, яка виявляє закономірності в мовних сигналах, що дозволяє класифікувати та розрізняти певні звуки, слова та інші об’єкти.

- Інформатика, наука про ефективні комп'ютерні алгоритми та методи їх програмно-апаратної реалізації, які можуть бути використані в системах розпізнавання мовлення;

- Математика, яка є повною теоретичною основою методів і алгоритмів, що розробляються.

Систему розпізнавання мовлення можна класифікувати за багатьма характеристиками, які можуть змінюватися залежно від методу застосування для вирішення проблеми розпізнавання мовлення. Ось лише основні типи ознак [5]. За типом мови розпізнавання:

- Окремі слова;

- Споріднені слова та фрази;

- Мова Злиття.

Ключова відмінність у відокремленні окремих слів від інших типів – це можливість визначити початок і кінець слова, що набагато складніше в інших ситуаціях. Тому перше завдання спрощено, і немає потреби використовувати занадто складні методи [6].

Різниця у вимові між мовцями створює додаткові проблеми для системи розпізнавання мовлення. Тому найскладнішим завданням цих двох завдань є створення незалежної від мовця системи, в якій потрібно використовувати методи регулювання змін між мовцями. Рішення першої проблеми полягає в тому, щоб навчити систему на основі голосу одного

оратора, або адаптувати існуючу систему незалежно від диктатора до нового оратора. Через якість розпізнавання мовлення [7]:

- Чиста мова (SNR> 40 дБ);
- Низький рівень шуму (SNR ~ 20-40dB);
- Дуже шумна мова (SNR <20 дБ).

Чиста мова відноситься до звукового сигналу, який містить лише людську мову без будь-яких чужорідних звуків, не пов'язаних із мовленням. Однак у практичному застосуванні через зовнішні перешкоди (відлуння, сторонні звуки тощо), технічні перешкоди, що виникають під час отримання та передачі аудіо сигналу, а також перешкоди, що виникають під час передачі, зазвичай немає повністю безшумних аудіо даних. Сама вимова мови. . Ідентифікація шуму є складним завданням, і це не завжди складно. За способом поділу мови на основні одиниці [8]:

- Розпізнавання фонем;
- Розпізнавати через частини слова (склади, інтервали);
- Розпізнавання тексту.

Людська мова формується шляхом вимовляння окремих звуків, які при поєднанні утворюють змістовні слова та речення. Тому при створенні системи розпізнавання мови можна виділити неподільні природні основні одиниці, такі як фонеми, склади, слова, або розпізнати інші штучно створені одиниці. Якість розпізнавання, очевидно, залежить від вибору тих чи інших типів базових одиниць, тому вибір тут більше залежить від завдань, потреб та внутрішньої інтуїції розробника. За розміром словника:

- Є невеликий словник близько 100 слів;
- Є невеликий словник близько 1000 слів;
- Володіти великим словником приблизно з 5000 слів або більше.

Розмір словникового запасу визначає шлях вирішення проблеми розпізнавання мови. Особливо система розпізнавання мовлення Словники часто використовують фонеми або склади для сегментації слів. Для менших

розмірів словника може бути достатньо використання слів як основної одиниці. За типом бібліотеки словника [9]:

- Певна (закрита) лексика;

- Невизначений (необмежений) словниковий запас.

Розпізнавання мови може виконуватися у заздалегідь визначеному та незмінному просторі слів або у всьому просторі природної мови, допускаючи слова, яких немає в словнику. Останнє є складним завданням як для обчислень, так і для алгоритмів. За типом синтаксису [10]:

- Виправлена граматика;

- Природна граматика.

Окрім лексики, розпізнавання мови також пов’язане з граматикою чи структурою виразу, який необхідно розпізнати. Це можуть бути вирази з певною структурою і синтаксисом, тобто мати фіксовану граматику, або мати загальну структуру, властиву природній мові. Відповідно до розташування пристрою захоплення звуку:

- Близька відстань (до 20 см);

- Віддалене розташування (понад 20 см).

Важливим фактором у побудові системи розпізнавання мовлення є відстань від динаміка до пристрою захоплення мови (мікрофона, телефону тощо), оскільки воно може залежати від ступеня зовнішнього шуму чи луни, що безпосередньо впливає на сигнал до- коефіцієнт шуму (SNR) [11].

За типом розв’язуваного завдання:

- Розпізнати одну команду;

- Розпізнавати комбіноване мовлення;

- Розпізнавання голосу телефону;

- Розпізнавання новин, лекцій тощо;

- Підтвердити діалог;

- Розпізнавати спонтанне мовлення.

Дана класифікація не є такою класифікацією, але вказує на низку ключових завдань, які представляють особливий інтерес для наукової

спільноти через їх практичну цінність і складність у розв’язанні. Сучасні системи автоматичного розпізнавання мовлення засновані на застосуванні статистичних методів, а їх суть полягає в обробці великої кількості мовних даних для встановлення відповідної акустичної моделі [12]. Збір таких мовних даних та їх орфографічна транскрипція називається акустичним корпусом. За типом вимови тексту розрізняють два типи акустичних корпусів – корпус, що містить матеріали для читання (новини, статті, слова тощо) і спонтанне мовлення (діалоги, лекції тощо).

Більшість систем автоматичного розпізнавання мовлення (ASR) складаються з процедур аналізу й обробки аналогових сигналів і процедур розпізнавання. Під час аналізу аналогового сигналу голос виділяє атрибути, які використовуються для визначення того, що буде сказано пізніше в процесі розпізнавання. Розглянемо коротку історію системи ASR у контексті цих двох процесів [13]. Перша спроба створити систему ASR була зроблена в 1950-х роках. Для розпізнавання чисел була створена залежна від мовця система [14]. Як сигнальну характеристику використовується спектральний резонанс голосних у слові. У 1959 році був створений модуль, який може розпізнавати десять голосних незалежно від мовця. У 1960-х роках Японія виготовила кілька машин, які могли використовувати спеціальні аналізатори спектру для визначення гучних звуків. Також було створено пристрій для розпізнавання фонем [17]. У 1970-х роках у сфері розпізнавання мови було два великих відкриття: використання методів динамічного часу. Викривлення (DTW) [18], засноване на часовому вирівнюванні мовних діалектів, і метод лінійного прогнозного кодування (Linear Predictive Coding-LPC) [19], успішно використовується для ідентифікації сигналів з низькою бітовою швидкістю (кількість бітів інформація, що передається за секунду)). В AT У лабораторії & T Bell створена система розпізнавання, в якій обробка акустичного сигналу базується на аналізі LPC та процесу розпізнавання на основі DTW [20]. У 1980-х роках дослідження розпізнавання мовлення перейшли від методів на основі шаблонів до методів статистичного

моделювання. Використовується прихована ERP-подібна ГУовська модель (HMM). Робота Бейкера [21] є однією з перших, яка використовує HMM для вирішення проблем розпізнавання мови. Наприкінці 1980-х до проблеми розпізнавання був застосований метод, заснований на штучній нейронній мережі (ANN). Сьогодні більшість систем ASR використовують HMM в процесі розпізнавання. З 1990-х років розпізнавання мови покращилося. Кількість впізнаваних слів у словнику зросла до десятків тисяч. Використання алгоритмів швидкого декодування дозволяє проводити розпізнавання в режимі реального часу. У сучасних системах, пов'язаних з диктантами, які розпізнають одне слово, його кількість досягає 20 000 слів, а відсоток помилок становить менше 0,1% [22]. У незалежній від мовця системі, яка розпізнає комбіновану мову з тисячі слів, є близько 5% помилок [23].

Використання сучасних методів розпізнавання мовлення в реальному часі вимагає великих обчислювальних ресурсів, кількість яких зазвичай обмежена. Сьогодні багато алгоритмів не можуть бути широко використані, наприклад, у мобільних пристроях, що змушує дослідників шукати більш ефективні та оптимізовані методи. Через його простоту та малу кількість операцій на ітерацію він розглядається в дипломі Цей алгоритм можна використовувати як альтернативу існуючим методам розпізнавання мовлення в реальному часі. Найпереконливіші приклади англійського акустичного корпусу, який використовується для завдань розпізнавання мови, є наступними. TIMIT – мовленнєвий репрезентативний корпус об'єднаних мов, який має на меті розпізнавання злитого мовлення та проведення мовленнєвих досліджень [24]. Комутатор – це акустичний корпус спонтанного голосу телефону, призначений для розпізнавання телефонних розмов [25]. TIDIGITS – велика мовна бібліотека, зосереджена на задачах незалежного розпізнавання цифрових послідовностей під диктовку Aurora2 – акустичний корпус, що містить шум Версія мовної бібліотеки [25].

1.2 Методи вирішення проблеми розпізнавання голосових команд

Методи вирішення проблеми розпізнавання мови можна розділити на три категорії [26]:

- Акустико-мовленнєвий метод;
- Метод розпізнавання зображень;
- Методи штучного інтелекту.

Акустично-мовленнєвий метод використовує характеристики мови (фонеми) для представлення та розпізнавання мовних сигналів. Усі фонеми можна описати з різними характеристиками, що відрізняє фонеми одна від одної. Прикладами таких ознак розрізнення голосних і приголосних є участь/відсутність мови, наявність/відсутність формантної структури, «періодичність», амплітуда сигналу. Формантна структура голосних різна. Приголосні можна розрізняти за сигналом часу та спектральною структурою. Тож майте гарний набір Особливості, легко побудувати класифікатор у вигляді дерева рішень [26], який буде позначати мовні сигнали фонемами. Наступний етап Це визначення слова, яке представляє послідовність фонем. Послідовність фонем можна розшифрувати шляхом прямого порівняння зі словами в словнику. Через високу мінливість мови та складність точного вимірювання акустичних властивостей визначення та класифікація фонем є неоднозначним процесом, тому можуть виникнути великі проблеми з декодуванням. Тому цей метод вимагає глибокого розуміння мовлення та способу їх опису, що не завжди можливо.

Спосіб розпізнавання образів зустрічається в мовному сигналі певних зображень без чіткого виділення та сегментації. Всі методи також включають два етапи: навчання і розпізнавання образів. На першому етапі система надає набір мовленнєвих сигналів, за якими система повинна розпізнавати певні закономірності та вивчати образи, які повністю описують мову. Далі, на етапі розпізнавання, система повинна порівняти поданий їй мовний сигнал із зображенням, засвоєним на першому етапі, і класифікувати невідому мову за певним ступенем подібності. Метод штучного інтелекту використовує

методи та алгоритми перших двох методів. Системи штучного інтелекту намагаються приймати рішення так само, як і люди. Наприклад, експертна система може використовувати метод акустичного мовлення, щоб сегментувати та позначати мовний сигнал, а потім вивчати та коригувати.

Основні характеристики акустичних сигналів - основні елементи периферичної слухової системи - характеристики равлика та тональна організація слухової системи [26] - передбачають використання спектру Фур'є як базової характеристики для побудови вищих рівнів. Тонотопічна організація означає, що спектральні компоненти акустичного сигналу поширюються до слухової кори периферичної системи, майже без змішування. Сусідня частота поширюються в топологічно суміжних нейронних каналах. Прийняти спектр використовує «віконний» аналіз з довжиною вікна 15-25 мс і будь-яким вікном згладжування (Хеммінга, Ханнінга та ін.) [27] або рекурсивним фільтром з таким же часом загасання. Довжина вікна залежить від характеру голосового сигналу або зміни частоти складу, а його діапазон становить 8-12 Гц. Вікно аналізу зазвичай переміщується на 10 мс, забезпечуючи частоту власного вектора 100 Гц. У популярному наборі кепстральних коефіцієнтів частоти кепстри (MFCC) [28], спектром маніпулюють для імітації обробки слухової системи: складові канали спектру, зібраного відповідно до частотної шкали кепстри, і значення енергії в кожному є логарифмічними. . Кепстряна шкала – це псевдо логарифмічна шкала частот, отримана шляхом експериментів у психоакустичних експериментах. Його важливість полягає не тільки в тому, що він відповідає нашому уявленню про роботу слухової системи, але й у тому, що він поєднує більш широку область високочастотних спектральних компонентів, що може значно зменшити розмірність вектора ознак. амплітуда логарифмічного аналогового сигналу, що поширюється через нейронний канал Характеристики стиснення. Крім того, виконайте протилежне косинусне перетворення і перейдіть до опікуна, залишивши лише перші 12 компонентів. Косинусне перетворення призводить до

декореляції спектральних компонент, подібно до перетворення Карунена-Лоева [29]. Насправді, природа вагової функції Карунена-Лоева наштовхнула на ідею використання косинусного перетворення. Косинусне перетворення не має аналога в механізмі обробки сигналів нервової системи.

Автоматичне розпізнавання мовлення є унікальним завданням для моделювання системи, яка розвивалася протягом сотень тисяч років протягом філогенезу. У цій системі «передавач» і «приймач» сигналу контролюються органом – мозком. За ці тисячі років вони знайшли «універсальну мову», яку необхідно розшифрувати. Це дуже сумнівно Дешифрування дає альтернативи, і що ще більш підозріло, так це вони кращі за «натуральні». Очевидним результатом цих міркувань є те, що перспективна система розпізнавання мовлення повинна максимально використовувати фізіологічні досягнення в області слухового аналізу. Однак слід пам’ятати, що сліпе копіювання відкритих механізмів сприйняття може навіть погіршити розпізнавання, оскільки в живих системах механізми обробки рідко ізольовані один від одного. Швидше, мова може йти про загальний принцип обробки інформації в живих системах – багатоетапну ієрархічну обробку з використанням великої кількості нейронів і зворотного зв’язку. Виходячи з наведених міркувань, ми оцінимо методи та результати системи розпізнавання з «біологічної» точки зору.

Нижче ми розглянемо три найбільш успішні методи, які використовуються в сучасних системах розпізнавання мовлення. Перший метод використовується для покращення розпізнавання мовлення, заснований на виборі вектора атрибутів із сигналу з урахуванням особливостей сприйняття звуку вухом людини. Він включає аналіз несучої частоти та вирівнювання гучності сигналу. Найбільш поширеними методами, які використовують цей метод, є метод капітальної частоти (Mel Frequency Cepstral Coefficients, MFCC, Davis & Mermelstein, 1980) і метод лінійного прогнозування (Perceptual Linear Prediction, PLP, Hermansky, 1990). Синхронізація з шаблоном і розширене порівняння (маскування) & Lilly,

1997) є характеристиками людського сприйняття, які можна моделювати та використовувати для виділення характеристик більшої стійкості до шуму. З цією метою був створений метод зміни розміру кадру (аналіз змінної частоти кадрів, VFR, Zhu & Alwan, 2000). З огляду на деталі роботи нервових клітин, відповідальних за слухові рецептори, був запропонований метод з автокореляцією (Subband-Autocorrelation, SBCOR, Kajita & Itakura, 1994) [30]. Інший метод заснований на аналізі звукових сигналів. Різниця між картиною, отриманою в процесі навчання шумового сигналу, і «чистого» сигналу є основною причиною нестабільності системи розпізнавання. Мета цього методу – зменшити цю різницю. Передбачається, що шум в звуковому сигналі є адитивним і стабільним. Розрахункове значення середнього шуму віднімається з Cepstral Mean Subtraction (CMS, Furui, 1981) або спектру (Spectral Subtraction, SS, Virag, 1999) і розраховується з даних шуму. Деякі модифікації цього методу включають нелінійне спектральне віднімання (нелінійне спектральне віднімання, NSS, Lockwood & Boudy, 1992), яке використовує спектральні огинаючі. Такі методи вимагають хорошої оцінки шуму, що важко отримати на практиці, особливо у випадку нестаціонарного фонового шуму [31]. Інший спосіб впоратися з різницею між властивостями, отриманими від шумового сигналу і чистого сигналу, - це використання фільтра високих частот. Припустимо, що шум у сигналі не є статичним, а повільно змінюється з часом. Метод RASTA (Relative Spectral Analysis, Hermansky & Morgan, 1994) представлений шляхом реєстрації відносних спектральних змін. І виключіть ці повільні зміни, викликані шумом. У цьому випадку чітка оцінка шуму не потрібна. Третій метод заснований на використанні багатовимірного простору (Ephraim & Trees, 1994). Основна ідея цього методу - знайти лінійну карту, яка мінімізує функцію вартості. Зазвичай вектор атрибутів множиться на матрицю перетворення як таке відображення. Прикладами таких методів є аналіз головних компонентів (PCA) та аналіз незалежних компонентів (ICA, Koscor, 2000), а також проектування багатовимірних підпросторів (Gales, 2002) [32].

Інший метод вирішення проблеми розпізнавання мови заснований на методі та алгоритмі порівняння мовного сигналу із зразком [10]. Ідея полягає в наступному. Існує набір опорних сигналів, які можуть бути закодовані в часовій або частотній області, і вони являють собою словник для ідентифікації. Стандарти можуть бути сформовані шляхом усереднення сигналів одного типу та представлення їх у певній формі кодування (наприклад, у кодовій книзі). Саме розпізнавання здійснюється шляхом порівняння нового сигналу з усіма вибірками та визначення найбільш підходящого кандидата на основі певної міри або подібності. Найпопулярнішим з цих методів є алгоритм динамічного викривлення часу (Dynamic Time Warping), або алгоритм DTW скорочено Т.К.Венчук [14]. Цей алгоритм дозволяє ефективно виміряти подібність двох часових рядів на основі динамічного програмування.

До середини 20 століття з розвитком комп'ютерів стало можливим розпізнавати обмежений набір команд майже в реальному часі (зручно для користувачів). Кожна команда представлена одним або кількома вибірками - набором спектральних векторів. Кількість векторів у кожній команді та кожному наборі реалізації, як правило, різна і залежить від тривалості оголошення. Отриманий голосовий сигнал «Х» повинен належати до однієї з команд і бути представлений у такому ж вигляді. Тому необхідно порівняти стандартний набір різного змісту та тривалості з набором «Х» і вирішити, який ідеальний «Х» ближчий. Основна складність цієї задачі полягає в тому, що кількість векторів у порівнянні різна. Очевидний алгоритм, заснований на градієнтному методі, підходить для «жадібного» алгоритму, який дає дуже нестабільні результати навіть для одного і того ж динаміка, і зовсім не працює навіть при невеликих шумах. Перше рішення на основі алгоритму динамічного програмування було запропоновано Т.К. в 1968 році. Вінчук [13]. Цей рік можна вважати рубежем, де стає можливим практичне застосування системи розпізнавання мови. Самостійно, але невдовзі, той самий підхід запропонував В.М. Величко та Н. Г. Загоруко [14]. На Заході

цей метод також був запропонований самостійно, але він запізнився на 10 років [15]. Оскільки цей алгоритм не тільки відіграв надзвичайно важливу роль на ранніх етапах розвитку систем розпізнавання мовлення, але й продовжував використовуватися в іншій формі чи іншою назвою в сучасних системах, і його можна використовувати з «біології», ми будемо здійснити опис. Ідея цього методу проста і може бути розглянута на якісному рівні. Завдання полягає в тому, щоб порівняти два набори векторів різної довжини, а у векторному просторі є метрика або метрика близькості. Уявіть, що ми порівнюємо еталон із самим собою: відкладаємо вектор стандартного символу на осі X і Y. Тоді на квадраті зі сторонами, рівними числу ідеальних векторів (N), буде «гірський пейзаж», симетричний діагоналі (0,0) (N, N), але діагональ абсолютно пряма «Долина» з висотою 0 (оскільки відстань від вектора до самого себе дорівнює 0). Якщо порівняти два різних критерії, що належать одному слову, «топографічна карта» буде спотворена, але якщо використані об'єкти повністю відображають процес сприйняття, можна сподіватися, що певна долина все ще знаходиться близько до діагоналі вздовж ломаної лінії. Тепер це прямокутник N, M, де M – довжина другого зразка. Метод динамічного програмування дозволяє розрахувати мінімальну загальну висоту або кумулятивну відстань, отриману при переміщенні від точки (0,0) до точки (N, M), і за потреби вказується шлях відновлення відстані. Отримане число зазвичай нормується кількістю пропущених вузлів, сумою довжин слів або довжиною коротших слів і розглядається як відстань між двома висловлюваннями. Звичайно, система, яка використовується в реальній реалізації, має багато контрольованих параметрів, які можуть оптимізувати якість ідентифікації та скоротити час виставлення рахунків. Цей метод дозволяє версії, що залежить від диктанту, розпізнавати 100-300 слів за ідеальних умов з імовірністю 90-98% [34].

Щоб надати кожному слову якість, яка не має відношення до диктанту системи, напишіть кілька критеріїв від різних мовців (у процесі навчання, якщо його неможливо розпізнати, додайте критерії від нового мовця). Крім

того, існують рішення для стандартизації динаміків і кластеризації динаміків. Очевидно, що цей метод не має аналогів у роботі живих систем, і він має багато недоліків, які роблять його непридатним для розпізнавання великих словників, великої кількості нових мовців і, звичайно, злиття голосів. Спочатку ми помічаємо довільність близькості в просторі векторів ознак. Кватерніонний блок (сума модуля різниці компонентів), Евклідов і Махаланобіс використовуються як запобіжний підхід до спектрального вектора. Коефіцієнт утримання використовує метрику Кульбака-Лейблера [16] або проекцію [17], а коефіцієнт лінійного прогнозування використовує метрику Ітакура-Сайто [18], яка не має суттєвого впливу на якість розпізнавання. Оскільки загальний частотний спектр мовлення падає зі швидкістю приблизно 6 дБ/окт [19], висока частота вносить дуже незначний внесок у відстань між векторами порівняно з низькою частотою. Для боротьби з цим явищем розрізняють мовні сигнали, хоча з огляду на штучну природу методу необхідно вводити множники для кожної спектральної складової та оптимізувати їх шляхом тестування. Це вже потребує великої кількості баз мовних даних, які мають тільки почали формуватися в ті роки. . Незалежно від того, які показники та коефіцієнти використовуються, відносний внесок різних спектральних компонентів у відстань залишається незмінним і знову довільним, тоді як слухова система вибирає необхідні компоненти зі спектру та ігнорує інші компоненти. Однак основним недоліком цього методу є його «ієрогліфічність», тобто словникові слова представлені як цілісні об’єкти без внутрішньої структури, що унеможливлює побудову словника. Хоча через низьку дискримінаційну силу цього методу (слова просто плутають), створювати словник із більш ніж 100-300 слів більше не має сенсу. Зверніть увагу, що термін «розпізнавання за допомогою динамічного програмування» по суті відноситься до набору алгоритмів для динамічного програмування та представлення мови, що використовує ланцюжки векторів ознак без структурування слів, що може ввести в оману алгоритм значення ідеї використання динамічного

програмування. Як зазначалося вище, цей алгоритм існує в більш сучасних системах розпізнавання, хоча він є штучним, але обчислює лише мінімальне значення кумулятивної відстані, а не максимальне значення кумулятивного журналу ймовірності [35].

Нехай X = {x1, x2, ..., xN} і Y = {y1, y2, ..., yM} – дві послідовності, що представляють дискретну (часову) послідовність або набір векторів. Виразимо відстань між компонентами (векторами) xi та yj у вигляді Dij = d (xi, yj), яку можна задати як середньоквадратичну відстань векторного набору або інші метрики. Визначте Cij як відстань між послідовністю Xi = {x1, x2, ..., xi} і Yj = {y1, y2, ..., yj} таким рекурсивним способом:

$$C11 = D11,\ Ci1 = Di1 + Ci\text{-}11,\ C1j = D1j + C1j\text{-}1,$$

$$Cij = Dij + \min\{Ci\text{-}1j,\ Cij\text{-}1,\ Ci\text{-}1j\text{-}1\},\ 1 \le i \le N,\ 1 \le j \le M.$$

Тоді очевидно, що значення CNM буде відстанню між вихідною послідовністю X і Y (згідно з цим визначенням). Однак у випадку з Загальним ця відстань не буде мірою в звичайному розумінні, оскільки це не буде нерівність трикутника. Перевага цього алгоритму полягає в тому, що він може порівнювати мовні сигнали з різною швидкістю динаміка. Крім того, алгоритм DTW не вимагає великої кількості обчислювальних ресурсів і пам’яті, що робить його популярним у вбудованих системах і мобільних телефонах. Недоліком цього алгоритму та всього стандартного методу є обчислювальна складність, коли є дуже великий словник приблизно з тисячі слів [36].

Інший алгоритм – прихована ERP-подібна ГУовська модель. Щоб створити систему розпізнавання мови з великим словниковим запасом, потрібно тренуватися на репрезентативних даних. У цьому випадку часто використовуються статистичні методи машинного навчання, які можуть

витягувати закономірності з невизначеної та неповної інформації. У голосових сигналах джерелами невизначеності та неповноти є зміни голосу та розповіді, зовнішнього середовища та каналів зв'язку. Оскільки прихована марківська модель має природну здатність описувати тимчасові та спектральні характеристики мови, вона стала найбільш успішним і популярним статистичним методом у проблемах розпізнавання мови. Баум і його колеги дали теоретичну основу прихованої ERP-подібної ГУовської моделі в класичних роботах кінця 1960-х і початку 1970-х років; у 1970-х він дав її в класичних роботах своїх колег з IBM.

Крім мовного сигналу, ланцюг ERP-подібна ГУова будується як односторонній перехідний процес між станами в дискретні моменти. Ймовірність переходу в наступний стан залежить тільки від поточного стану, і не залежить від того, в якому стані перебуває процес. знаходиться в попередній момент [двадцять два]. Однак ця вимога призводить до некоректних гістограм станів життєвого циклу [23-26] і від неї відмовилися в сучасних системах. Модель, в якій ймовірність переходу в наступний стан залежить від часу перебування в поточному стані, називається неоднорідною ERP-подібною ГУовською або напівмарківською. Отже, марківська модель звуку або слів – це один або кілька безперервних станів, які визначають функцію щільності ймовірності та функцію ймовірності переходу в просторі ознак. Для простору квантованих ознак функція щільності ймовірності може бути виражена в дискретній або безперервній формі. У другому випадку вони зазвичай апроксимуються сумою гаусових функцій з діагональними коваріаційними матрицями. Діагональність коваріаційної матриці зменшує кількість параметрів навчання та спрощує деякі алгоритми, а також надає аналітичні розв'язки для деяких задач, що дозволяє використовувати лише числові розв'язки для повної коваріаційної матриці.

Ми ввели процес навчання на якісному рівні, тобто метод отримання щільності ймовірності стану та функції ймовірності переходу до наступного стану. Для навчання використовується мовна база даних (записи

мовленнєвих сигналів і відповідні тексти), частина якої сегментується (ERP ГУується) досвідченими лінгвістами, для чого ми побудуємо ERP-подібну ГУовську модель (зазвичай фонеми, без залучення складності) Одиниця або фрагмент визначає це складне поняття). Після правди Матеріал перекладено на серію векторів, програм, Використовуючи встановлений лінгвістом межу, вектор ознак кожної фонеми збирається в окремий набір, для яких легко побудувати функцію щільності ймовірності через набір наближених функцій Гаусса. Програма також аналізує тривалість сегментів і будує їх гістограми. Знаючи гістограму тривалості кожної фонеми, легко розрахувати ймовірність виходу після виходу з відповідного стану фонеми на певний проміжок часу. Після отримання першої оцінки параметрів стану (функції щільності ймовірності та ймовірності переходу) ми використовуємо алгоритм Баума-Уелша [22, 27] або Вітербо [22, 27] для повторної оцінки параметрів, щоб максимізувати характеристики бази даних імовірнісний ланцюг станів Векторна послідовність. Наступним кроком є використання несегментованої частини решти мовної бази даних. Справа в тому, що хоча отриманий стан недостатньо точний для розпізнавання мовлення, коли текст сегмента відомий, вони можуть дуже точно сегментувати мовний матеріал. Цей метод називається «примусовим вирівнюванням», він дозволяє використовувати дуже великі бази даних, і, як ми побачимо пізніше, розмір мовних баз даних завжди недостатній.

Однак, порівняно з методом динамічного програмування, опис словникового слова за станом, що відповідає фонемі, не призводить до істотного покращення якості розпізнавання. Цьому є просте пояснення. Інваріантна фонема, яку ми представляємо, насправді є повним звуковим рядом, і іноді існують великі відмінності в складі векторних ознак. Адже мовний апарат не завмирає ні в якому положенні для захоплення наступної фонеми, а безперервний рух артикулятора створює безперервну траєкторію в просторі символів. Тому сусідні фонеми впливатимуть на вимову цієї фонеми. Цей ефект називається «коартикуляцією». Іншими словами, наша

функція щільності ймовірності складається з усічених траєкторій у просторах ознак, які перетинаються в різних напрямках. Функції щільності ймовірності різних фонем значно перетинаються в просторі ознак, що призводить до більших помилок. Тому для більш точного опису всі поєднання цієї фонеми з попередніми та наступними звуками слід розглядати як окремі акустичні об'єкти, для яких можна будувати стан. Ці об'єкти називаються «Трифонами», оскільки вони з'єднують три послідовні фонеми. Подібно до визначення «Bifons», воно поєднує попередню або наступну фонему для опису фонеми. Біфони використовуються для опису початку або кінця мовного сегмента, а також коли даних недостатньо для побудови стану Трифона. Розглянуті досі контекстно-вільні фонеми аналогічно називають монофонами.

Розглянемо траєкторію в просторі ознак, яка перетинає область функції щільності ймовірності даної фонеми – відрізок траєкторії є довгим об'єктом, а вектори ознак на початку і в кінці визначаються фонемами до і після, і можуть бути істотно відрізняються. Описувати такі об'єкти в одному стані не рекомендується, оскільки це спричинить додаткові помилки через перетин з іншими подібними об'єктами. Для опису трифона зазвичай використовуються три стани. Крайній стан описує область сигналу, на яку впливають сусідні фонеми, а центр - ту частину центральної фонеми, на яку найменше впливають сусідні фонеми. Однак кількість станів не повинна збігатися з глибиною контекстної залежності – можна розглянути п'ять тонів [28, 29] і використовувати три стани для їх моделювання або для моделювання кількох станів одного тону. Необхідність побудови стану для трифонів, тобто з урахуванням контексту, приносить нові труднощі - кількість мовленнєвих одиниць зросла настільки, що навіть дуже великих баз даних недостатньо для оцінки їх статистики. Нижче наведено дані з [30], що стосуються англійської мови, та широко використовуваної бази даних Wall Street Journal Pronunciation Lexicon. Для англійської мови кількість фонем становить близько 50 (кількість не є фіксованою – деякі поширені дво- або

тритональні можна віднести до окремих фонем заздалегідь). Тоді загальна кількість трифонів становить 503 = 125 000, деякі з яких заборонені фонетичними правилами цієї мови і ніколи не з'являться, є 95 221 трифонів. За згадані понад 57 годин ігрового часу в базі даних, що містить понад 36 000 речень, всього 22 804 трифони, з них лише 14 545 трифонів, які з’являються більше 10 разів. Очевидно, що вивчення стану прихованої марківської моделі вимагає великої кількості вибірок моделованого об'єкта. Число 10 можна вважати мінімально достатнім. Тому більше 80 000 трифонів є невидимими або невидимими, але їх можна знайти в роботі системи розпізнавання.

Кількість параметрів ERP-подібної ГУовської моделі може досягати 1000-2000 (сюди входять матриця переходів і параметри функції Гаусса наближеної функції щільності ймовірності). Якщо це число помножити на потрійне число (50000-100000), загальна кількість параметрів, які підлягають оцінці в процесі навчання, становить приблизно 108-109. Тому з’явилася нетривіальна задача – оцінка мільйонів параметрів, більшість з яких не з’явилася в навчальній базі даних. Ця проблема вирішується статусом посилання [30, 31]. З’єднайте або об’єднайте стани функції з найбільшою щільністю ймовірностей, що перекриваються. Процес починається знизу, починається з моно, розбиває моно на трифон з функцією щільності ймовірності найменшого перекриття і закінчується, коли більше не вистачає даних для навчання нового трифона. Тому створюються лише ті три тони, які ділять функцію щільності цього єдиного тону на велику і малу частини і які можна ефективно викладати.

Отримана система розпізнавання значно перевершила систему, засновану на методі динамічного програмування. Однак для нових динаміків або інших каналів передачі якість розпізнавання значно знижується. Необхідно адаптувати систему розпізнавання до нових мовців на основі кількох мовних матеріалів або якимось чином у процесі. Починаючи з 1990-х років, вирішенню цих проблем присвячена велика робота. Нормалізація характеристики та адаптація моделі. Нормалізація ознак відноситься до

спотворення вхідного мовного сигналу або його вектора ознак, щоб наблизити середню характеристику з вектором, що становить базу даних. Для цього використовують довжину уздовж каналу мінус середній утримувач і нормалізацію [32, 33]. Адаптація означає виявлення руху та спотворення моделі системи, тобто функції щільності ймовірності стану, щоб вони найкраще відповідали мовним даним нового мовця. Використовуйте байєсівську адаптацію або максимізацію апостеріорної ймовірності [34] та лінійну регресію максимальної ймовірності [35-37].

Зі сфери розпізнавання мовця та особи виникли методи адаптації за допомогою власних мовців [38, 39]. Для адаптації моделі до шуму використовуються векторні ряди Тейлора [9, 40]. Усі ці методи підвищили якість розпізнавання до дуже високого рівня, тому систему розпізнавання можна використовувати в голосових системах самообслуговування (IVR) і системах, призначених для співрозмовників. Модель мови також використовується для розпізнавання комбінованого мовлення. Довільна модель мови дозволяє формально описати мову, а точніше, саме ті аспекти, які необхідні для підвищення якості автоматичного розпізнавання мови. Визначаючи можливий порядок слів, ми піднімаємося на більш високий рівень опису мови порівняно з мовленням, тому необхідно враховувати системні відносини вищого рівня. Модель, яка використовується для опису слова в реченні, може бути дуже складною. Враховуючи синтаксис та семантичну структуру речення, вона також може бути дуже простою. Вона передбачає, що ймовірність будь-якого слова здається рівною (у цьому випадку ми в основному даємо вгору Мовний аналіз і правила) та характеристики природної мови). Модель мови дозволяє з’ясувати, які послідовності слів у мові є більш імовірними, а які менш імовірними. На жаль, усі мовні моделі української мови мають найменший внесок у розпізнавання, оскільки порядок слів у реченні досить вільний, а його композиційний характер виражається багатослівними формами, через традиційне скорочення вимови. Повертаючись до оцінки методу з точки зору

«біології», помічаємо абсолютну штучність методу. Він повинен відпочивати в певній межі, власне, так і зробив.

1.3 Марківський підхід до розпізнавання голосових команд

У сучасних системах розпізнавання мови широко використовується так звана безперервна прихована марківська модель, її функція розподілу ймовірностей виражається у вигляді суміші нормальних розподілів (модель суміші Гаусса). Безперервна латентна марківська модель з N станів {1, 2, ..., N} і M сумішей (для кожного стану) визначається триплетом $\lambda = \{A, B, \pi\}$, де

1) A = {aij}-ймовірність переходу з одного стану в інший, а саме

$$a_{ij} = P[q_{t+1} = j[q_t = i],\ 1 \le i, j \le N,$$

2) B – ймовірність розподілу вектора спостереження, а саме

$$b_j(X) = \sum_{k=1}^{M} c_{jk} \aleph(X, \mu_{jk} \Sigma_{jk}),\ 1 \le j \le N,$$

Де X – вектор спостереження, а cjk – вага k-ї суміші стану j, яка є функцією середнього значення μjk та нормального розподілу коваріаційної матриці Σjk. На практиці коваріаційна матриця виражається як діагональна матриця.

3) $\pi = \{\pi i\}$-імовірність розподілу початкового стану моделі, а саме

$$\prod_i = P\ [q_1 = i],\ 1 \le i \le N.$$

У процесі застосування паливно-мастильних матеріалів вирішуються три основні задачі [38].

Задача 1. Нехай задана послідовність спостереження O = {o1, o2, ..., oT} і модель $\lambda = \{A, B, \pi\}$, тоді нам потрібно ефективно обчислити умовну

ймовірність P (O | λ), а саме ймовірність цієї послідовності спостережень для даної моделі.

Задача 2. Враховуючи задану послідовність спостереження O = {o1, o2, ..., oT} і модель λ = {A, B, π}, нам потрібно визначити послідовність станів Q = {q1, q2, .. ., qT}, що найбільше відповідає цій послідовності спостережень.

Задача 3. Необхідно знайти алгоритм, який може ефективно знаходити параметри моделі λ = {A, B, π}, щоб максимізувати ймовірність P (O | λ).

Перше завдання – це завдання оцінки, яке дозволить вибрати найкращу модель серед кількох конкуруючих моделей. Для її розв'язання використовується програма прямого чи зворотного обчислення [1]. Друге завдання – це завдання декодування, тобто знайти найкращу послідовність станів, яка в певному сенсі може генерувати цю послідовність спостереження. Залежно від задачі, яку потрібно розв'язувати, критерій оптимальності може бути довільним. Задача декодування зазвичай вирішується за допомогою алгоритму Вітербі або його модифікації. Третє завдання – це завдання вивчення моделі, тобто знаходження оптимальних параметрів моделі для максимізації ймовірності P(O|λ). Тут використовується ітераційний метод Баума-Велча (Baum-Welch) або метод максимізації очікування (Expectation-Maximization).

Крім алгоритмічного вирішення цих трьох задач, необхідно також вибрати паливно-мастильну архітектуру. Основними параметрами є кількість станів, зв'язок між ними, а також наявність зв'язаного стану та тривалість стану. Перевага прихованих марківських моделей полягає в тому, що вони повністю моделюють тимчасові характеристики мови. До недоліків можна віднести складність розуміння процесу розпізнавання та неможливість покращити якість розпізнавання мовлення шляхом аналізу природи помилки. Крім того, використання суміші нормальних розподілів також має свої недоліки, тобто воно не може ефективно описати дані, що оточують деяке різноманіття. Зокрема, це показує, що мова може бути змодельована деякими динамічними системами з невеликою кількістю параметрів, а це означає, що

розміри мовних особливостей набагато менші, ніж ті, які зазвичай використовуються в сумішах нормального розподілу.

Технологія розпізнавання мовлення ще дуже молода, але дуже перспективна в комерційному сенсі, що передбачає великий капітал і швидке зростання. Однак, незважаючи на достатнє фінансування, з моменту використання моделі MERP ГУова (середина 1960-х років), приблизно за 40 років, прогрес у ідентифікації був відносно невеликий. Новому методу не вдається подолати результати, отримані неперервною ERP-подібною ГУовською моделлю гауссової апроксимації з ймовірністю стану, або поліпшення настільки незначне, що не варто істотного ускладнення системи. Досягнуті результати не дозволяють використовувати системи розпізнавання мови як масштабні комерційні продукти, хоча специфічні процедури у вузьких дисциплінах ефективні протягом тривалого часу.

Багато дослідників вважають, що природа проблеми відповідає можливостям штучних нейронних мереж. Я почав експериментувати з нейронними мережами дуже давно. Прикладом може бути стаття 1990 року [41], в якій було представлено багато перспективних ідей. Зокрема, використовується довгострокова функція у вигляді супервектора, який складається з 9 послідовних векторів спектру кепстри та циклічного зв’язку між вихідним шаром і вхідним шаром, що дозволяє враховувати контекстну релевантність. Зверніть увагу, що довгострокові символи описують сегменти траєкторій у просторі символів досить «біологічно». Хоча система фактично використовує ті ж функції, що й стандартна модель MERP ГУова, плюс згадані покращення, неможливо перевершити стандартну систему, засновану на суміші Гаусса. Цей факт викликав таке здивування в науковому середовищі, що в 1996 році була опублікована красномовна стаття під назвою «На шляху до підвищення рівня помилок розпізнавання мовлення» [42], в якій намагалися пояснити створення системи розпізнавання мови. Автор пояснив, що причиною відсутності прогресу є те, що ERP-подібна ГУовська модель на основі гауссової суміші використовується десятками

дослідницьких центрів по всьому світу, і після кількох років оптимізації її практично неможливо створити будь-яку нову, грубу. Хоча цей аргумент важко спростувати, але останні роботи з використанням різних типів багатошарових нейронних мереж показують, що існує ще одна проста причина – нейронні мережі не мають достатньої інформаційної потужності, оскільки потужність комп'ютерів не дозволяє використання декількох шарів і джерел. Шар складається з тисяч нейронів, що відповідають трифонам (замість десятків монофонів у ранній системі).

Багато дослідників вважають, що природа проблеми відповідає можливостям штучних нейронних мереж. Я почав експериментувати з нейронними мережами дуже давно. Прикладом є стаття 1990 р. [41], в якій було представлено багато перспективних ідей. Зокрема, використовується довгострокова функція супер вектора, яка складається з 9 безперервних векторів кепстряного спектру та циклічного співвідношення між вихідним шаром і вхідним шаром, що дозволяє враховувати контекстну релевантність. Зверніть увагу, що довгостроковий символ дуже «біологічно» описує відрізок траєкторії в просторі символів. Хоча система фактично використовує ті ж функції, що й стандартна модель MERP ГУова, плюс згадані вдосконалення, неможливо перевершити стандартну систему, засновану на суміші Гаусса. Цей факт викликав таке здивування в науковому середовищі, що в 1996 році була опублікована красномовна стаття під назвою «Покращення рівня помилок розпізнавання мовлення» [42], в якій намагалися пояснити створення системи розпізнавання мови. Автор пояснив, що причиною відсутності прогресу є те, що ERP-подібну ГУовську модель на основі гаусової суміші використовують десятки дослідницьких центрів по всьому світу, після кількох років оптимізації створити якісь нові і грубі практично неможливо. Хоча цей аргумент важко спростувати, останні роботи з використанням різних типів багатошарових нейронних мереж показують, що існує проста причина – нейронні мережі не мають достатньої інформаційної потужності, оскільки потужність комп'ютерів не дозволяє використовувати

кілька шарів і джерел. Нейрони, що відповідають трифону (замість десятків монофонів у ранній системі).

1.4 Нейромережевий підхід до розпізнавання голосових команд

Нейронна мережа або персептрон з будь-якою кількістю прихованих шарів є загальним апроксиматором [42], тобто мережа з одним прихованим шаром, використана до цього етапу, може апроксимувати будь-яку поверхню в просторі ознак. Однак успіху розпізнавання мови можна досягти лише за допомогою багатошарової мережі. Це тому, що неможливо або надзвичайно важко створити розумну техніку для ініціалізації масштабу мережі з одним прихованим шаром, що призводить до далеко не оптимального набору масштабу під час навчання. Використання багатошарових нейронних мереж ставить перед собою нове завдання – розробку нових алгоритмів навчання, що може стати трендом у майбутній роботі, пов’язаної з використанням нейронних мереж. Одним із методів є використання пошарового навчання для ініціалізації, починаючи з нижніх шарів [41, 42]. Вхідний вектор ознак розглядається як цільова функція першого прихованого шару. Вихідний вектор може містити кілька послідовних векторів MFCC або мельспектральних ознак. Щоб уникнути такого ж перетворення, вхідний вектор буде шумним. Таким же чином навчається наступний шар нейронної мережі відтворювати вихідний сигнал попереднього шару. Тому існує цілих 5-7 рівнів професорів. Після виконання ініціалізації першого шару включається стандартний алгоритм зворотного поширення помилки всієї мережі, а його цільова функція відображає належність вхідного сигналу до відповідного трифона. У порівнянні з класичним методом гаусової суміші цей метод має очевидні переваги – результат розпізнавання завжди кращий. Багатошарова мережа, натренована на мовному матеріалі за 309 годин мовлення, показує, що він краще, ніж навчання за 2000 годин мовлення. Метод гаусової суміші дає кращі результати.

Слід зазначити, що запропонований алгоритм навчання створює систему, подібну до функціонального аудіювання. У слуховій системі виявляються нейрони, які реагують на певні події акустичними сигналами [8]. У міру того, як сигнал проникає в центральну частину слухової системи, природа сигналу, що виділяється спеціалізованими нейронами, стає все більш складною і вибірковою. Початкове навчання кожного рівня нейронної мережі виконує одне і те ж завдання - кожен шар вчиться шукати ознаки сигналів вищого рівня. Якщо внутрішній шар нейронної мережі випромінює голосовий сигнал Сигнали, характерні для загальних мов, вони можуть використовуватися однаково на всіх мовах, Для кожної нової мови викладається лише вихідний рівень нейронної мережі. Це буде дуже важливо, оскільки для вивчення одного рівня нейронної мережі буде потрібно менша мовна база даних, ніж навчання всіх 5-7 шарів.

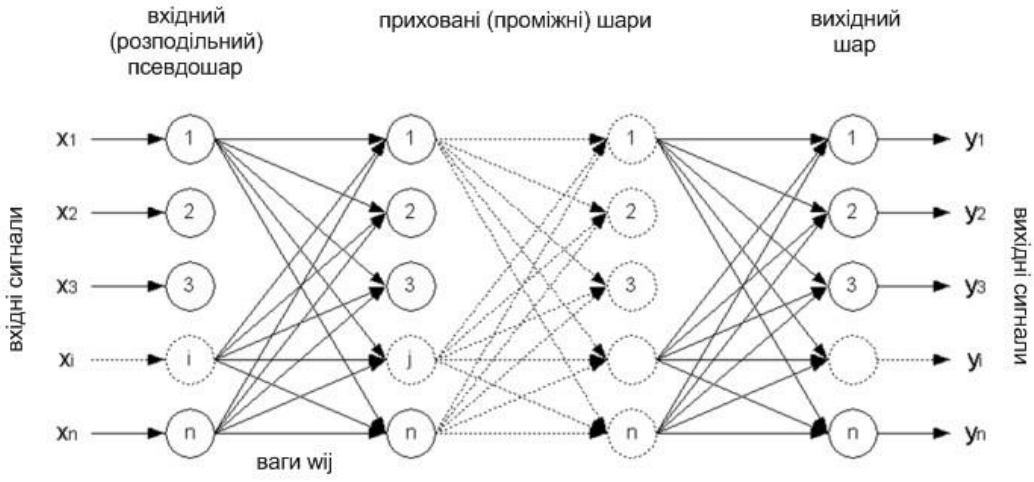


Рис. 1.1 – Навчання нейронної мережі.

Експерименти повністю підтвердили цю можливість. Порівняно з одномовною моделлю, комбіноване використання баз даних французької, німецької та італійської мов може знизити відносні помилки розпізнавання на 3,3-5,4% [42]. Слід зазначити, що інформація про особливості голосового сигналу, що міститься у внутрішньому шарі нейронної мережі, може бути використана для розпізнавання мови, яка не має відношення до мови.

Експерименти проводилися з використанням внутрішнього шару нейронної мережі, навченої на основі європейських мов, для завершення вихідного шару китайської нейронної мережі. Оскільки китайська база даних збільшилася з 3 годин до 139 годин, відносний приріст становив від 21,1% до 8,3% [40]. Ця технологія відкриває можливість створення систем розпізнавання для будь-якої мови, включаючи мови з низькими ресурсами.

Оскільки нейронна мережа не може розпізнати динамічні об’єкти, для порівняння моделі з сигналом все ще використовується модель MERP ГУова, але тепер як вектор ознак використовується трифонів апостеріорний набір ймовірностей, отриманий на виході нейронної мережі. Цей метод використання нейронних мереж є одним із перших монофонічних методів, запропонованих Г. Германським та його співавторами [41]. Очевидно, що розвиток багатошарових нейронних мереж з пошаровим навчанням є найбільшим кроком до біологічного механізму обробки сигналів. Насправді залишився лише один надзвичайно штучний елемент – це все той же алгоритм динамічного програмування в моделі MERP ГУова, але його назва Вітербо, оскільки апостеріорна ймовірність трифонів, отримана нейронною мережею, все ще залишається «Розтягніть» модель з її допомогою.

Багато експертів вважають, що для ідентифікації фонем можна використовувати повторювані нейронні мережі. Повторювані нейронні мережі складаються з нейронів, об’єднаних у спрямований циклічний процес. Це забезпечує пам’ять для нейронної мережі, тому вона може розпізнавати процеси, а не лише статичні об’єкти, як описано вище, глибокі нейронні мережі. Хочеться сподіватися, що така мережа зможе визначити існування фонем чи інших акустичних об’єктів та накопичити вхідну інформацію, що дозволить відмовитися від динамічного програмування та формалістичних методів ERP-подібної ГУовської моделі. Ідея використання рекурентних нейронних мереж була вивчена дуже рано [39, 41], але відсутність комп’ютерних можливостей не давала переваги основним методам того часу.

Опишіть сигнал довжиною приблизно 25 мс. Вікно аналізу рухається на 10 мілісекунд. Щоб відобразити сегмент сигналу довжиною 300 мс (такий вид контекстно-залежної розмірності знайдено в [42]), потрібно близько 30 векторів ознак, тому розмірність створеного супер вектора може бути від 300 до 1000. Використовувати векторну розмірність незручно. Більш ефективним здається створення нейронної мережі з рекурентними з'єднаннями, яка зберігає інформацію про сигнал як інтегратор рекурентного фільтра з витоком. Він заснований на резервуарній нейронній мережі, побудованій шляхом інтеграції нейронів і витоку [42]. Така мережа містить шари взаємопов'язаних нейронів, на відміну від «персептронів», в яких можуть бути з'єднані тільки нейрони різних шарів. У роботі досліджується двонаправлена нейронна мережа, яка дозволяє розглянути попередній та наступний контексти, пов'язані з розглянутим сегментом.

Штучна нейронна мережа – це математична модель біологічних нейронів, вперше змодельована Маккалоу і Піттсом у 1943 році, а ERP ГУ застосував її до машинного навчання Хебба та розпізнавання образів у 1954 та 1949 роках.) [КлERP ГУ] і Розенблат у 1958 році. Завдяки роботі Мінського та Пеперта в 1969 році інтерес до нейронних мереж на деякий час зменшився [35] Вони показали, що вони не можуть використовувати нейронні мережі для моделювання функції «вимкнено АБО», а комп'ютер був недостатньо підготовлений. У той час викладання великої нейронної мережі. Однак із появою обчислювальної потужності та концепцій глибокого навчання в 2000-х роках почалася нова хвиля популярності нейронних мереж.

Поки що найпопулярнішими в задачах розпізнавання мовлення є багатошарові нейронні мережі (Deep Neural Networks), які він спрямований на заповнення пробілу в стандартній суміші нормальних розподілів, згаданих вище. Багатошарові нейронні мережі були успішно використані в акустичному моделюванні в Університеті Торонто, Microsoft Research, Google і IBM Research. Багатошарова нейронна мережа – це одностороння

штучна нейронна мережа з одним або кількома шарами (шарами), з прихованими нейронами між вхідним і вихідним шарами. На вході нейронна мережа отримує акустичні параметри фіксованого розміру, витягнуті з мовного сигналу. Кожен прихований нейрон j використовує логічну функцію для відображення вхідного сигналу x_j з попереднього рівня на вихідний сигнал yj. Для наступного шару:

$$y_j = \frac{1}{1+e^{-x}},\ x_j = b_j + \sum_i y_i w_{ij},$$

Де b_j – коефіцієнт відхилення нейрона j, i – індекс нейрона у верхньому шарі, wij – вага зв'язку даних i-го нейрона та j-го нейрона у верхньому шарі. , вихідний нейрон j буде мати значення. Вхідний сигнал xj перетворюється в клас ймовірності p_j як:

$$p_j = \frac{\exp\left(x_j\right)}{\sum_k \exp\left(x_k\right)}$$

Де k – індекс для всіх класів.

Навчання нейронної мережі полягає в підтримці найкращого набору ваг і використанні алгоритму зворотного поширення помилки між реальними результатами та фактичними результатами, отриманими на виході нейронної мережі для навчання [15]. Багатошарова нейронна мережа з нейронами в кожному шарі вимагає дуже великих обчислювальних ресурсів, даних і часу для повного навчання. Метод градієнтного спуску, який використовується під час навчання, може знайти лише локальне оптимальне значення. Якщо початкове значення ваги не встановлено, воно зменшить мережа, що використовується для перепідготовки. Щоб уникнути цієї ситуації та

перенавчання мережі, пропонується новий метод навчання мережі, який називається генеративним попереднім навчанням. Ідея полягає в наступному.

Кожен рівень навчається окремо, а вихідні дані попереднього рівня є вхідними даними для навчання наступного шару після навчання. Отримана в результаті цього навчання шкала є кращою початковою умовою для того, щоб нейронна мережа остаточно розрізняла навчання, при якому ці ваги будуть лише незначно змінюватися. Оскільки вона може апроксимувати будь-яку (статичну) нелінійну функцію, багатошарова нейронна мережа може повністю замінити суміш нормальних розподілів, але досі не знайдено жодного методу, який би повністю замінив приховану марківську модель. Прихована марківська модель знаходиться в часі моделювання, показує переваги з точки зору залежностей.

1.5 Технології попереднього оброблення мовних сигналів

За останні десятиліття сучасні системи автоматичного розпізнавання мовлення досягли значного прогресу, від простих програм, які покладаються на диктант, до систем автоматичної транскрипції новин, телефонних розмов і лекцій, які не залежать від диктанту. Хоча такі системи отримали широке застосування, проблеми розпізнавання мовлення далекі від вирішення щодо шумів, спотворення рядків, іноземних акцентів, швидкості та манери мови тощо. Проте з огляду на останній прогрес активно проводяться дослідження з розпізнавання мови, існує велика кількість літератури з цього питання.

Рисунок 1.2 – Компоненти систем розпізнавання мови

На Рис.1.2 показана загальна структура сучасних систем розпізнавання мови, що включають такі її компоненти як попередня обробка сигналу, акустична модель, мовна модель і пошук гіпотез.

Первинна обробка сигналу включає в себе такі процеси, як виділення та перетворення акустичних характеристик сигналу, адаптація до впливу шуму або змін між різними динаміками. Стандартним методом виділення акустичних ознак є обчислення вхідного вектора на основі базового спектрального коефіцієнта MFCC або коефіцієнта сприйняття лінійного прогнозування PLP. Далі вектор перетворюється на простір меншої розмірності за допомогою матриці лінійної проекції, щоб максимально розділити мовний клас. Зазвичай це робиться за допомогою лінійного дискримінантного аналізу (LDA) або його розширення на HLDA. Алгоритм SPLICE або QE використовується для забезпечення більшої завадостійкості. Обидва ці алгоритми були успішно протестовані на даних про шум з Wall Street Journal (WSJ). Адаптація до різних динаміків, нормалізуючи довжину мовного шляху опорного динаміка (VTLN), застосовуючи максимальну лінійну регресію правдоподібності в просторі параметрів (fMLLR) або мінімальну помилку мовлення в просторі параметрів (fMPE).

Акустична модель відображає акустичні характеристики мови (фонема, дифон, трифон тощо), на яку орієнтована система розпізнавання мови. Найпопулярніший метод тут заснований на моделі прихованої MERP ГУова (PMM) і змішаного нормального розподілу (GMM) або багатошарової нейронної мережі (DNN). Статистична мовна модель визначає ймовірність розподілу P (w1, w2, ..., wn) n слів w1, w2, ..., wn. Це значення апроксимується за допомогою n-грамів, розрахованих як частота їх повторення в репрезентативному тілі тексту. Вдалий метод побудовою мовної моделі є мовна модель Кнейзера-Нея, ієрархічна модель Пітман-Йор, модель максимальної ентропії або так звана модель «М». Ці методи засновані на статистичних методах. Проте є й інші способи використання синтаксичної структури мови. Метою декодера є використання інформації з акустичної та мовної моделей для обчислення найбільш вірогідної послідовності слів W з послідовністю акустичного вектора X. Класичним методом пошуку гіпотез є алгоритм Вітербі, який використовує динамічне програмування для

ефективних обчислень. Однак також використовуються методи, засновані на зважених кінцевих перетворювачах (WFST). Експерименти з використанням цих моделей дозволили досягти високої продуктивності. Наприклад, англійська система розпізнавання телефону (English CTS, R04 Test Set) незалежно від диктатора досягла показника помилки слів близько 15,2% [19]. Прикладами сучасних систем розпізнавання мовлення є системи з відкритим кодом HTK, CMU Sphinx, Kaldi, Julius та комерційні продукти Nuance's Dragon NaturallySpeaking, IBM ViaVoice, Microsoft Windows Speech Recognition, Apple SIRI та Google Voice Search [22].

Хоча на тлі більш ніж 30-річної стагнації автоматичне розпізнавання мовлення досягло значного прогресу за останні 3-4 роки, але в порівнянні з людиною можливості системи розпізнавання все ще дуже обмежені. Найважливіше те, що «термінальний пристрій» слухової системи може зрозуміти сказане, тому людині не складе труднощів у визначенні віддаленого, зворотного, мова з акцентом. Розмова з поганими каналами спілкування також може відрізнити мову мовця від поліфонії та визначити спонтанну мову. Усі ці завдання, особливо два останні, створили величезні труднощі для сучасних систем розпізнавання. Поки що автоматичні системи розпізнавання мови перевершували людей лише в задачах розуміння та моделювання мови. Не грає ролі, наприклад, для ідентифікації ізольованих команд або чисел. Розвиток систем розпізнавання мовлення буде пов’язаний з удосконаленням структури нейронних мереж, обов’язковою наявністю різних рівнів зворотного зв’язку, розробкою нових методів навчання таких нейронних мереж за допомогою алгоритмів динамічного програмування. Структура нейронної мережі повинна мати механізм адаптації та механізм повернення корекції. Якщо в архітектурі нейронної мережі з’являються певні елементи слухової системи, то метод навчання спирається на навчання дітей, наприклад, подання першого етапу найпростіших звуко модульованих голосних і сполучень приголосних, що не дивно.

2 ОПИС ТЕХНОЛОГІЙ РОЗРОБКИ ТА ПРОГРАМНИХ СЕРЕДОВИЩ

2.1 Опис середовища розробки

Python – потужна мова програмування, яку легко освоїти. Він має ефективні, розширені структури даних і прості, але ефективні методи об’єктно-орієнтованого програмування. Елегантний синтаксис Python, динамічний введення та мова інтерпретації роблять його ідеальним для написання сценаріїв і швидкої розробки додатків у багатьох областях на більшості платформ. Інтерпретатор мови Python і багата стандартна бібліотека (вихідний код і двійкові дистрибутиви всіх основних операційних систем) доступні на сайті Python і можуть розповсюджуватися безкоштовно. На цьому ж сайті є дистрибутиви та посилання на багато модулів, програм, утиліт та додаткових документів. Інтерпретатор Python можна легко розширити за допомогою функцій і типів даних, розроблених на C або C++ (або іншій мові, яку можна викликати із C). Python також можна використовувати як мову сценаріїв, вбудовану в програму для інших налаштувань функцій. Цей підручник має окреслити основні поняття та особливості Python для читача. Під час використання цього посібника зазвичай добре мати під рукою інтерпретатор Python, але всі приклади є самодостатніми, тому цю статтю можна легко прочитати. Опис стандартних об'єктів і модулів - посилання на бібліотеку Python. Довідковий посібник Python надає більш формальне визначення мови. Щоб написати розширення на C і C++, прочитайте Розширення та вбудовування інтерпретатора Python і Довідник API Python/C. Також є кілька книг, які детально описують Python.

Цей огляд не є вичерпним, у ньому розглядаються не всі функції чи навіть усі найбільш часто використовувані функції. Натомість він містить функції мови, які мають бути пріоритетними, і дозволяють читачам мати загальне розуміння смаку та стилю мови. Після прочитання ви зможете читати та створювати власні модулі та програми, а також будете готові

ознайомитися з різними модулями бібліотеки Python, описаними в довідці про бібліотеку Python. Python – це інтерпретована високорівнева об'єктно-орієнтована мова програмування. Для проектів у різних галузях це одна з рідкісних мов програмування, які є одночасно простими і потужними. Розширені структури даних, стислий синтаксис, динамічна семантика та ефективні методи об’єктно-орієнтованого програмування роблять його привабливим для написання сценаріїв, розробки додатків і веб-рішень. Код Python зазвичай організовується у функції та класи, які можна об’єднати в модулі, а модулі можна об’єднати в пакети. Однією з граматичних особливостей мови є використання відступів (пробілів або табуляції) для виділення блоків програмного коду, що дозволяє забезпечити відсутність круглих дужок, наприклад, «початок-кінець» або "{-}", тому поведінка та коректність програми можуть залежати від початкових пробілів у тексті. Вирази є зрілими операторами в мові програмування Python. Зміст, граматика, асоціативність і пріоритет операцій досить поширені для загальних мов програмування, спрямовані на зменшення кількості використовуваних дужок [7].

Python надає механізм для запису коду pydoc. Найкраще заповнити кожен модуль, клас, функцію або метод рядком документів. У цьому випадку ви можете отримати будь-яку допомогу в інтернеті, створити гіпертекстові документи для всього модуля і навіть застосувати doctest Автоматичне тестування модуля. Багата стандартна бібліотека є одним із привабливих аспектів мови програмування Python. Він має інструменти для обробки багатьох мережевих протоколів та інтернет-форматів, наприклад модулі для запису HTTP-серверів і клієнтів, для планування та створення повідомлень електронної пошти, а також для обробки XML. Набір модулів для операційної системи дозволяє писати кросплатформні програми. Є модулі для обробки регулярних виразів, кодування тексту, мультимедійних форматів, протоколів шифрування, файлів, серіалізації даних, підтримки модульного тестування тощо.

Дизайн мови Python побудований на основі об'єктно-орієнтованої моделі програмування.

Характеристики об'єктно-орієнтованої моделі програмування:

1. Клас також є об’єктом.

2. Множинне успадкування.

3. Поліморфізм. Усі функції віртуальні.

4. Упаковка. Поле може бути відкритим або прихованим.

5. Управління життєвим циклом об'єкта (конструктор, деструктор, розподільник пам'яті).

6. Перевантаження оператора (крім «is», «.», «=" та логічних символів).

7. Властивості. Ви можете використовувати функції для моделювання полів.

8. Контроль доступу до поля (моделювання полів і методів, частковий доступ тощо).

9. Метод виконання найпоширеніших операцій (значення істинності, метод «len»), глибоке копіювання, серіалізація, ітерація тощо).

10. Метапрограмування.

11. Інтроспекція.

12. Класи та статичні методи, поля класів.

13. Класи, вкладені в функції та інші класи.

Програмне забезпечення, написане мовою програмування Python, розроблено у вигляді модулів, а модулі можуть бути зібрані в пакети. Модулі можна розміщувати в каталогах і ZIP-архівах. Модулі можуть бути двох типів: написані на самій мові Python, і розширені модулі, написані іншими мовами програмування (з англійських модулів розширення).

Переваги та недоліки Python:

- Python – інтерпретована мова програмування. З одного боку, це значно спрощує налагодження програми, а з іншого – призводить до відносно низької швидкості виконання.

- Динамічний введення тексту. Немає необхідності заздалегідь оголошувати типи змінних у Python, що дуже зручно в розробці.

- Відмінна підтримка модульності. Python дозволяє розкладати програми на модулі, що допомагає повторно використовувати код в інших програмах.

- Вбудована підтримка Unicode у рядках. У Python не обов’язково писати все англійською, у програмі можна використовувати рідну мову.

- Підтримка об'єктно-орієнтованого програмування. У той же час його реалізація в Python є однією з найлегших для розуміння.

- Автоматичний збір сміття, без витоків пам'яті.

- Якщо функцій python недостатньо, інтегруйте їх із C та C++.

- Чітка і стисла граматика допомагає чітко відображати код. Зручна функціональна система дозволяє грамотно створювати код, і при необхідності його буде легко зрозуміти іншим. Ви також зможете навчитися читати програми та модулі, написані іншими.

- Велика кількість модулів. У деяких випадках для написання програми достатньо знайти відповідні модулі та правильно їх поєднати. Тому можна розглядати написання програм на більш високому рівні, використовуючи готові елементи, які виконують різні операції.

- Різні бібліотеки підтримки. Python має велику колекцію і функціональні можливості, які називають стандартною бібліотекою. Бібліотека надає багато функцій, необхідних для програм, від пошуку тексту через шаблони до веб-функцій. Python дозволяє розширення з ваших власних бібліотек і бібліотек, створених іншими розробниками.

- Швидкість розробки. У порівнянні зі скомпільованими або строго типізованими мовами (такими як C, C++ або Java), Python у багато разів більш дружній до розробників. Кількість коду в Python зазвичай становить одну третину або навіть одну п’яту еквівалентного коду в C++ або Java, що означає менше введення з клавіатури, менше затримок і менше обслуговування. Крім того, програма на Python запускається миттєво,

минаючи тривалу фазу компіляції та зв’язування, необхідні для деяких інших мов програмування, що ще більше підвищує ефективність роботи програміста.

- Кросплатформеність. Програми, написані на Python, будуть абсолютно однаковими, незалежно від того, працюють вони в операційних системах Windows або Linux. Розбіжності трапляються лише в окремих випадках, і їх легко передбачити наперед через наявність детальної документації.

2.2 Організація програмного забезпечення

На додаток до стандартної бібліотеки існує більшість інших бібліотек, які надають інтерфейси для всіх системних викликів на різних платформах. По-перше, ми повинні розрізняти numpy (цифровий пітон) і skipy (науковий пітон). Бібліотека Numpy призначена для обробки багатовимірних масивів, що дозволяє досягти продуктивності наукових обчислень порівняно зі спеціалізованими пакетами. Масив – це контейнер, який містить елементи одного типу (якщо елементи є вкладеними масивами, вони мають однакову довжину) і організовані в упорядковану багатовимірну матрицю. Доступ до елементів здійснюється за допомогою індексного кортежу цілих чисел. Numpy є основним пакетом наукових обчислень на Python. Numpy – це розширення мови програмування Python, яке додає підтримку великих багатовимірних масивів і матриць, а також велику багаторівневу бібліотеку функцій для обробки цих масивів. Зокрема, ми використовуємо таку функцію np.arrange(), яка приймає додатне ціле число n як параметр і повертає масив, що містить n-1 елементів з 0 до n-1. Ви також можете ініціалізувати масив за допомогою функції np.linspace(), яка приймає три параметри: число: початковий елемент масиву, кількість елементів у масиві та останній елемент масиву. Повертає масив чисел, рівномірно розподілених від початкового значення до кінцевого. Використовуйте np.array(), щоб вказати весь масив [8]. Бібліотека Sciру використовується для обробки файлів .wav. SciPy – це

відкрита бібліотека високоякісних наукових інструментів для мови програмування Python. SciPy містить модулі для оптимізації, інтеграції, спеціальних функцій, обробки сигналів, обробки зображень, генетичних алгоритмів, розв'язування загальних диференціальних рівнянь та інших поширених проблем у науці та техніці. Scipy вимагає, щоб бібліотека Numpy була встановлена заздалегідь. Функція scipy.io.wavefile використовується для відкриття та зчитування інформації з файлу аудиту, а саме функція read(), яка повертає вибірку та частоту дискретизації.

Scipy використовує Numpy і надає доступ до різних математичних алгоритмів. Numpy і Scipy прості у використанні, але в той же час дуже потужні. Matplotlib – бібліотека для побудови графіки та зображень і візуалізації обчислень на мові програмування Python. Бібліотека Matplotlib побудована на основі принципу ООП, але має програмний інтерфейс pylab. Scipy використовує Matplotlib. Бібліотека Matplotlib-A на мові програмування Python для візуалізації даних з використанням двовимірної (2D) графіки (також підтримується 3D-графіка). Matplotlib – це гнучкий і простий у налаштуванні пакет, який разом із NumPy, SciPy та IPython забезпечує функціональність, подібну до MATLAB. На даний момент пакет програмного забезпечення можна використовувати з кількома графічними бібліотеками. Пакет (бібліотека) Python NumPy надає програмістам спосіб ефективно обробляти велику кількість кубів. Як компонент і основа, пакет NumPy включено в більшість проектів, Використовуйте мову Python і вимагайте більш-менш виснажливих обчислень. Зокрема, з його допомогою написані популярні пакети обчислювальної математики та наукової графіки SciPy та mathplotLib. Модуль os забезпечує багато функцій, які працюють з операційною системою, і їх поведінка зазвичай не залежить від операційної системи, тому програма залишається переносною. Стандартні інструменти мови програмування Python дозволяють отримати доступ до об'єктів WWW із програм у простих і складних ситуаціях, у тому числі коли потрібно передати дані форми, ідентифікацію доступу до проксі тощо.

Слід зазначити, що при роботі з WWW в основному використовується протокол HTTP, але WWW не тільки охоплює HTTP, а й включає інші рішення. Використовувана схема зазвичай вказується на початку URL-адреси.

Urllib2 – це модуль, який дозволяє використовувати URL-адреси. Модуль має власні функції та класи, які допомагають обробляти URL-адреси. Urllib2 надає дуже простий інтерфейс у вигляді функції urlopen(). Ця функція може використовувати різні протоколи (HTTP, FTP, ()) для вилучення URL-адрес. Він приймає адресу як параметр для доступу до видалених даних. Крім того, urllib2 надає інтерфейси для обробки поширених ситуацій, таких як базова аутентифікація, файли cookie, проксі-сервери тощо. Модуль urllib2 має спеціальний клас для реалізації запиту на відкриття URL-адреси. Цей клас називається urllib2.Request. Цей екземпляр містить статус запиту. Ви можете вказати вихідні дані, які будуть надіслані на сервер у запиті. Крім того, ви можете надіслати на сервер додаткову інформацію (метадані) про дані, надіслані на сервер, або в самому запиті, який передається у вигляді HTTP-заголовків. Модуль Beautiful Soup – це парсер для розбору HTML/XML, він навіть може перетворити неправильні теги в дерево розбору. Він також підтримує природні методи навігації пошуку і модифікацію дерева розбору. У більшості випадків це допомагає заощадити час і робочі дні. Бібліотека PyWavelets (pywt), призначена для вейвлет-перетворення, також використовується для обробки аудіофайлів .wav. Це модуль з математичними функціями, який дозволяє аналізувати різні частотні компоненти даних. Вейвлет, як засіб багатомасштабного аналізу, дозволяє виділити основні характеристики сигналу і короткочасні високочастотні складові в акустичному сигналі. Вейвлет-перетворення замінює перетворення Фур'є, оскільки перетворення Фур'є має багато недоліків у порівнянні з вейвлет-перетворенням, що призводить до втрати інформації тимчасових характеристик обробленого сигналу. Цей аналіз передбачає використання частотно-часового позиціонування, наприклад вікна даних.

3 ПОПЕРЕДНЯ ОБРОБКА ГОЛОСОВИХ СИГНАЛІВ

3.1.Виділити характеристики мовного сигналу та розпізнавання

Звукові сигнали є одним із засобів взаємодії людини з навколишнім середовищем, між людьми. Звук залежить від багатьох фізіологічних параметрів динаміка, і по суті є індивідуальною характеристикою кожної людини. Однак звук не є постійною особливістю, він буде змінюватися протягом життя людини, а також на нього впливатиме здоров'я та емоції. Розглянемо архітектуру сучасних систем розпізнавання мови. Будемо вважати, що вхідними даними є сам сигнал у форматі .wav або з мікрофон. Процес розпізнавання мови включає наступні етапи (рис. 3.1):

- Попередня обробка та параметризація сигналу;
- Перетворення сигналу у вектор ознак;
- Визначте мовну частину (класифікацію).

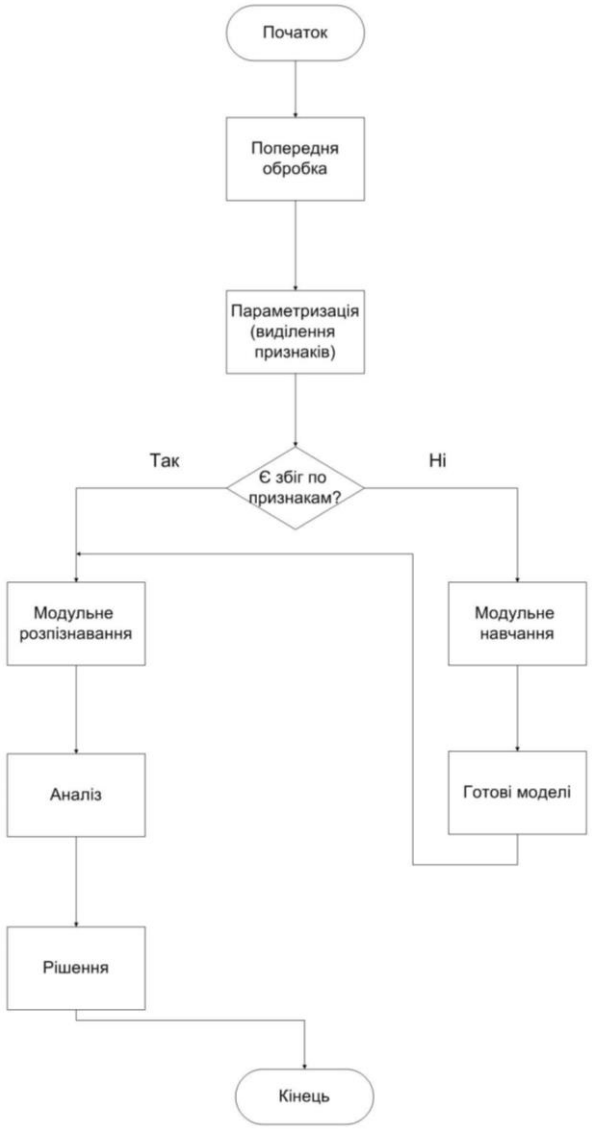


Рисунок 3.1 – Базові складові системи ERP ГУ

Більшість сучасних систем автоматичної ідентифікації використовують модульну архітектуру. Модуль попередньої обробки включає (Рисунок 3.2):

- Цифрова фільтрація;
- Розділити мовний сигнал за допомогою накладання кадрів;
- Спектральне перетворення;
- Обробка сигналу за допомогою віконних функцій;
- Спектральне перетворення.

Рисунок 3.2 – Стадії попереднього оброблення мовного сигналу.

Попередньо оброблений мовний сигнал перетворюється в набір векторів ознак, що містить спектральні ознаки: коефіцієнти збереження частоти кепстри MFCC, які, у свою чергу, містять інформацію про сигнал, і область, що містить мову, перетворюється в набір коефіцієнтів, а потім одиницю (класифікація) визнається:

1) Використовуйте набір програмних смугових фільтрів (DFT) для отримання частотного спектру мовного сигналу;

2) Перетворення прийнятого частотного спектру голосового сигналу:

а) Логарифмічна зміна масштабу в амплітудно-частотному просторі;

б) згладити спектр, щоб виділити його огинаючу;

в) Кепстровий аналіз, тобто обернене перетворення Фур'є логарифма прямого перетворення.

Сучасні системи робочих станцій базуються на статистичному моделюванні. Ці системи спочатку навчаються на довгостроково зібраних мовних даних (процес навчання полягає в налаштуванні параметрів статистичної моделі). Потім, на етапі розпізнавання, система порівнює вхідне зображення з зображенням, введеним відповідно до навчальної моделі. Іншими словами, вектор ознак, що відповідає невідомій розпізнаваній мові, порівнюється з моделлю для обчислення ознаки ймовірності кожної моделі. Цей метод має очевидний недолік – він ефективний лише тоді, коли навчальний образ і мовна особливість тесту близькі, чого важко досягти на практиці. Проте сьогодні стандартним методом процесу ідентифікації моделі є статистичний метод прихованої мERP-подібної ГУовської моделі (РММ).

Останній етап – аналіз інформації, отриманої на перших кількох етапах, з урахуванням граматики та словника (мовної моделі): винесення рішення про те, яка модель близька до невідомого вектора символів. Модель мови дозволяє отримати значення ймовірності будь-якої послідовності слів, незалежно від спостережуваної послідовності. У вирішенні задачі обробки

мовного сигналу важливу роль відіграє етап параметризації - процес вибору вектора ознак мовного сигналу. На цьому етапі попередньо оброблений (відфільтрований, дискретований, квантований, сегментований тощо) мовний сигнал перетворюється у вектор ознак, що містить необхідну інформацію про сигнал. Вектор символів представляє характеристики мовного сигналу в часі. Більшість використовуваних векторів ознак так чи інакше пов'язані з частотним спектром або пов'язаними характеристиками мовного сигналу. Коефіцієнти MFCC використовуються як вектори ознак у дипломних проектах. Результатом попередньої обробки мовного сигналу є перетворення вихідної аналогової мови в цифрову форму для подальшої обробки. Основне припущення, зроблене в сучасних системах робочих станцій, полягає в тому, що мовні сигнали вважаються статичними (тобто їх ймовірнісні характеристики є відносно постійними) з інтервалами в десятки мілісекунд. Тому головною функцією попередньої обробки є сегментація – розкладання вхідного мовного сигналу на фіксовані інтервали та отримання оцінки спектру кожного інтервалу. Фільтрація покликана зменшити вплив локального спотворення на ознаки, і в майбутньому буде використовуватися для розпізнавання. Крім того, передній фільтр високих частот використовується для видалення постійних компонентів, які з'являються під час роботи записуючого пристрою. Для цих цілей можна використовувати простий фільтр першого порядку з коефіцієнтом 0,97:

$$y[n]=x[n]-0,97x[n-1]$$

Де x [n] – вхідний сигнал до фільтрації, а y [n] – вихідний сигнал після фільтрації. Метою сегментації мовного сигналу є отримання векторів ознак однакової довжини: спочатку мовний сигнал сегментується на частковому рівні, а потім перетворюється у кожному кадрі. Перекриття використовується для запобігання втрати інформації на кордонах сегментів.

Обробка сигналу за допомогою віконних функцій спрямована на зменшення негативного впливу граничних ефектів (спектральних спотворень), викликаних сегментацією. Щоб мінімізувати спектральні

спотворення, використовуються вікна w [n], які зводяться до нуля на початку і в кінці кожного сегмента.

Вікна Хеммінга зазвичай використовуються як такі вікна:

$$w[n] =\geq \begin{cases} 0,54 - 0,46\cos\dfrac{2\pi n}{N-1}, 0 \le n \le N-1, \\ 0,\ n < 0, n \ge N. \end{cases}$$

На рисунку 3.3 показано вікно Хеммінга та його частотний спектр. Після застосування віконної функції для отримання спектральних вибірок сегмента використовується дискретне перетворення Фур'є (DFT):

$$X[k] = \sum_{n=0} x[n] w[n] \exp\left(-j\frac{2\pi kn}{N}\right),$$

Де x[n] – зворотний відлік сигналу у часовій області, N – кількість відліків у сегменті, w[n] – віконна функція, а X[k] – зворотний відлік сигналу в спектральній області. . Отриманий зразок спектру переходить до наступного кроку Обробка, тобто обчислення різних типів векторів ознак: MFCC.

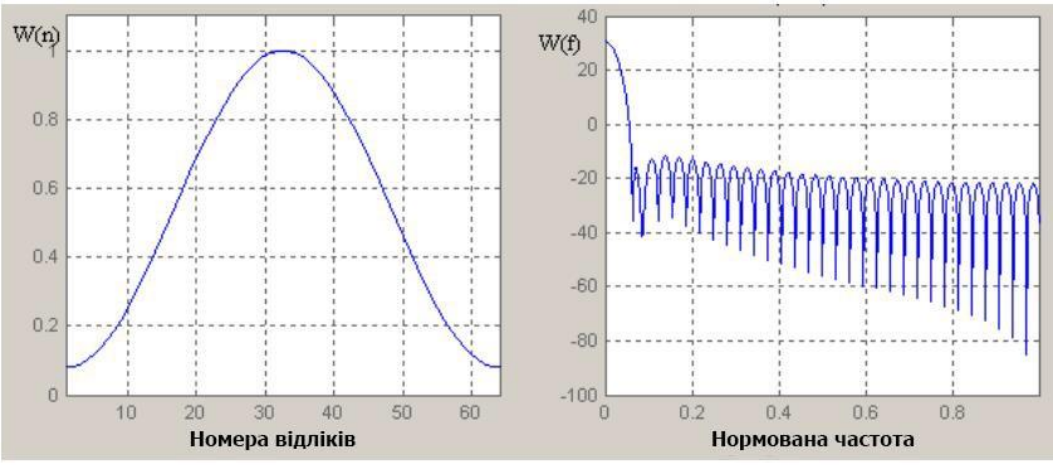


Рисунок 3.3 – Вікно Хеммінга та його спектр.

Сучасні методи запису дозволяють подавати звукові сигнали у вигляді часових рядів, показуючи зміни частоти в часі. Частотний спектр сигналу, його представлення в частотному просторі має більше аналітичної інформації, ніж сам сигнал. Для обчислення частотного спектру часто використовується швидке перетворення Фур'є, і його алгоритм дуже простий у реалізації, зі складністю O(Nlog2N), що менше, ніж складність класичного алгоритму дискретного перетворення Фур'є O(N2). Люди реагують на зміну частоти, тому вирішуючи задачі, пов'язані з аналізом людських голосів, часто використовують «кепстр» (результати). Застосування перетворення Фур'є в спектрі сигналу. Також в процесі еволюції звук в діапазоні низьких частот містить більше корисної інформації, ніж звук у діапазоні високих частот. З урахуванням цих особливостей слуху людини було розроблено коефіцієнт збереження частоти кепстри («кепстрою» – абревіатура англійського слова «melody»). За допомогою цих коефіцієнтів можна більш ретельно проаналізувати інформацію, отриману з низькочастотного діапазону, і зменшити вплив високочастотних компонентів, які зазвичай містять зовнішні шуми, на результати розпізнавання. Весь запис розбивається на невеликі інтервали тривалістю ~10-30 мс (квазістатичний час сигналу), які називаються кадрами. Набір коефіцієнтів збереження частоти кепстри розраховується окремо для кожного кадру, який буде використаний для майбутньої кластеризації. Алгоритм розрахунку кепстрового коефіцієнта частоти кепстри можна розділити на наступні етапи:

-Сигнал розбивається на кадри;

-Застосувати вагову функцію (вікно) до кожного кадру;

-Застосування перетворення Фур'є;

-Розрахунок кепстри.

За звичайних умов звуковий сигнал нестабільний, тобто його амплітуда та частотний спектр змінюються з часом, що робить багато методів аналізу непридатними. Але короткий інтервал близько 10-30 мс можна вважати стабільним. Зазвичай використовуваний метод кадрування сигналу полягає в

наступному: розділіть сигнал на інтервали довжиною N мс таким чином: початок першого кадру збігається з початком запису, а другий кадр починається з інтервалом M мс (M < N), відповідно на NM мс Перекриття з першим кадром. Незважаючи на свою стаціонарність, таке подання сигналу не дозволяє використовувати перетворення Фур'є. Якщо частота ERP подібних ГУонік (частотних складових) сигналу не узгоджується з основною частотою перетворення Фур’є, у спектрі можуть бути «зайві» ERP-подібні ГУоніки, які будуть лише Ідея «підняти шум» отримала. Цей ефект називається «спектральним розмиттям» або «спектральним витоком». На рисунку 3.3 показано випадок N = 20 мс і M = 16 мс.

Застосування вагової функції (вікно). Одним з можливих рішень є застосування спеціального типу вагової функції до сигналу:

$$\omega(n),\ 0 \le n \le N-1$$

Результат застосування вагової функції до кожного кадру такий:

$$y(n) = x(n) * \omega(n),\ 0 \le n \le N-1$$

Результат x (n) - значення часового ряду в точці n; y (n) - це зважене значення ряду часу в точці n (Рисунок 3.4).

Переважною є функція «м'якої» вагової ваги, яка зменшить значення на межі кадру до нуля. Ця операція називається «гладка». Найбільш використовуваною є зважувальна функція Хеммінга, яку можна виразити такою формулою:

$$\omega(n) = 0.53836 - 0.46165 * \cos(2\pi n / N-1)$$

Наступним кроком є застосування перетворення Фур'є для перетворення сигналу з простору-часу в частоту. На практиці найчастіше використовується швидке перетворення Фур'є, яке має такий вигляд:

$$Y_n = \sum_{k=0}^{N-1} y_k \cdot e^{-\frac{2\pi jkn}{T}}, 0 \le n \le N-1, j = \sqrt{-1}$$

Серед них yk – зважене значення часового ряду в точці k, Yn – комплексна амплітуда n-ої ERP-подібної ГУоніки сигналу, виражена у часових рядах. Результатом цього кроку є спектр сигналу.

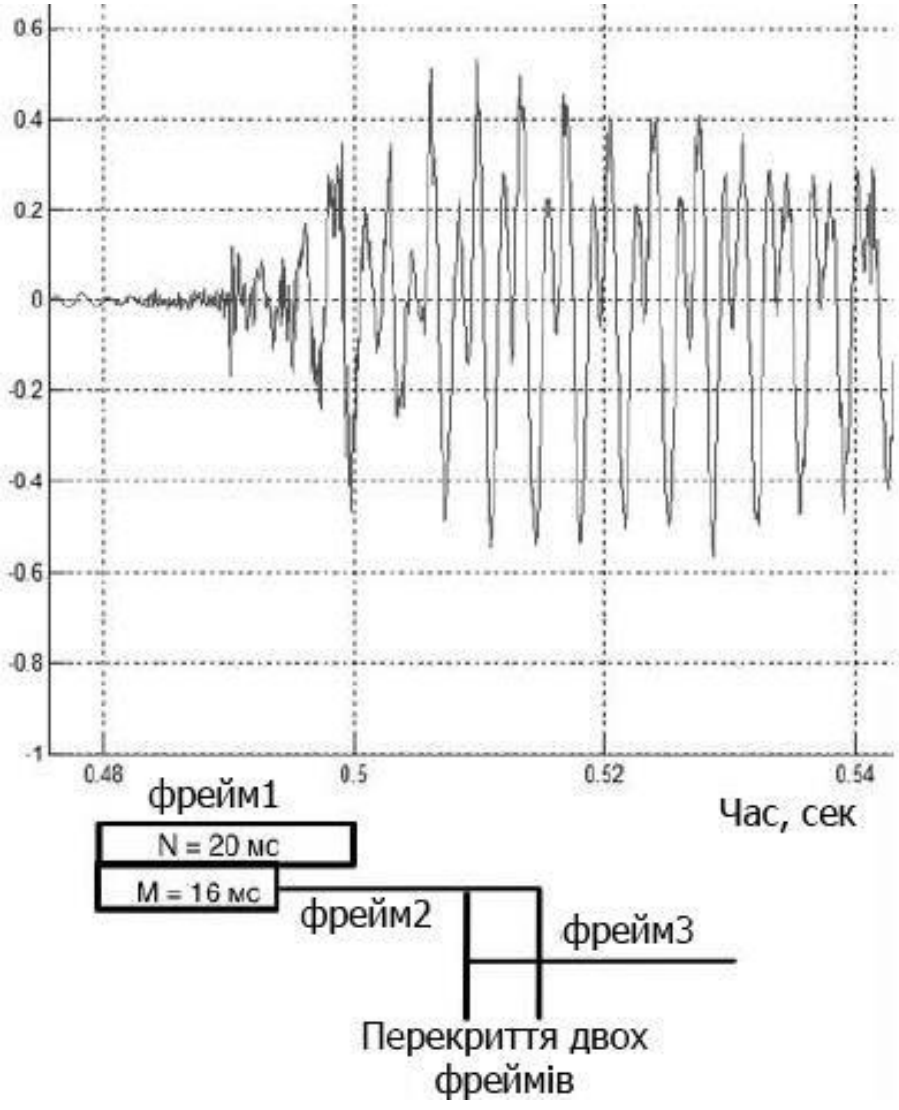


Рисунок 3.4 – Сегментація мовного сигналу.

Для того щоб усунути негативний вплив шуму, сигнал обробляється спеціальним частотним фільтром, про який піде мова нижче. Принцип роботи смугового фільтра полягає в наступному: у всьому наборі ERP подібний ГУонік, які складають звуковий сигнал, фільтр залишає лише ті ERP-подібні ГУоніки, частоти яких потрапляють в задану смугу пропускання.

Якщо система розпізнавання може розпізнати слово, незалежно від того, хто його сказав, то система розпізнавання не залежить від мовця. Насправді реалізувати таку систему дуже важко, оскільки звуковий сигнал багато в чому залежить від гучності, інтонації, стану та настрою оратора. (Рисунок 3.5) показує його символи запису однієї і тієї ж фрази, сказаної різними мовцями. Для отримання інформації з цих сигналів часто використовуються тональні фільтри, які усереднюють спектральні складові в певних частотних діапазонах, тим самим зменшуючи залежність сигналу від динаміка. Цей тип фільтрів є основою технології MFCC (Mel Frequency Cepstral Coefficient), яка використовується в системі розпізнавання, розглянутій у цій статті.

Рисунок 3.5 – Два різних мовця говорять одну фразу.

На цьому етапі використання кепстряного частотного фільтра до спектру сигналу застосовується спеціальний тип фільтра. Кожному значенню

частоти, отриманому на попередньому кроці, присвоюється значення на шкалі частот кепстри. Це значення шкали для частот нижче 1000 Гц точно відповідає частотному спектру сигналу, отриманому за допомогою перетворення Фур'є, а частоти вище 1000 Гц є логарифмічними. Результатом є виправлений енергетичний спектр сигналу mel (f) для кожної ERP-подібної ГУоніки частоти f, який розраховується за такою наближеною формулою:

$$\text{mel}(f) = 2595*\lg(1+f/700)$$

До цього спектру застосовується спеціальний тип фільтра, який відповідає певному набору кепстряних коефіцієнтів $\tilde{s}_k$ для кожної частоти, k = 1, ..., К, де К - кількість коефіцієнтів кепстри (на практиці часто вибирають Значення від 12 до 24). На попередньому кроці алгоритму отримані коефіцієнти $\tilde{s}_k$ необхідно перетворити. Для цього зручно використовувати дискретне косинусне перетворення, яке описується такою формулою [10, 12]:

$$\tilde{c}_n = \sum_{k=1}^{K} \lg\left(\tilde{s}_k\right) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K} \right], 0 \le n \le K$$

Серед них $\tilde{c}_n$ - отриманий коефіцієнт утримання частоти кепстри.

Алгоритм звукового вектора застосовується до кожного кадру, тому останній відповідає набору кепстряних коефіцієнтів, який використовується як модель користувача для кластеризації в більшості робіт, званий звуковим вектором. Але зміна співвідношення кепстри також містить деяку інформацію про користувача. Основна відмінність цієї роботи від попередньої полягає в розширенні акустичного вектора шляхом врахування динамічної зміни коефіцієнта кепстри δі, який виражається як різниця між коефіцієнтом утримання частоти кепстри кадру та попереднього:

$$\delta_i\left(\tilde{c}_k[i]\right) = \tilde{c}_k[i-1] - \tilde{c}_k[i]$$

У цьому методі перший кадр не може бути використаний для кластеризації, оскільки зміна коефіцієнта збереження частоти кепстри дорівнює нулю. А L-кількість елементів звукового вектора x-подвоюється:

$$L = |x| = \left\|\left[\tilde{c}_1,\ldots,\tilde{c}_k,\delta\left(\tilde{c}_1\right),\ldots,\delta\left(\tilde{c}_k\right)\right]\right\| = 2K$$

Припустимо, є дві числові послідовності (a1, a2, ..., ap) і (b1, b2, ..., bn). Як бачимо, довжина двох послідовностей може бути різною. Алгоритм спочатку використовує різні типи відхилень для обчислення локальних відхилень між елементами двох послідовностей. Найпоширенішим методом розрахунку відхилення є обчислення абсолютного відхилення (евклідової відстані) між значеннями двох елементів. В результаті отримуємо матрицю відхилень з n рядків і t стовпців Звичайні члени:

dtj = \ ai-bj \, i = 1, n, j = 1, t [4].

Використовуйте алгоритм динамічного програмування та наступні критерії оптимізації, щоб визначити мінімальну відстань у матриці між послідовностями:

aij = dij + min (ai-1j-1, ai-1j, aij-1),

де: atj – мінімальна відстань між послідовностями (a1, a2 ,., an) і (b1, b2, ..., bn).

Шлях деформації - це мінімальна відстань між елементами a11 і anm в матриці, і складається з тих елементів atj, які представляють відстань до anm. Розрахунки проводили на двох коротких послідовностях. Результати наведені на Рис. 3.6, де виділено порядок деформації.

	-2	10	-10	15	-13	20	-5	14	2
3	5	12	25	37	53	70	78	89	90
-13	16	28	15	43	37	70	78	105	104
14	32	20	39	16	43	43	62	62	74
-7	37	37	23	38	22	49	45	66	71
9	48	38	42	29	44	33	47	50	57
-2	48	50	46	46	40	55	36	52	**54**

Рисунок 3.6 – Визначення відстань між двома послідовностями.

Відповідно до алгоритму швидкої збіжності [5], існують три умови для забезпечення роботи DTW:

1. Монотонний – шлях ніколи не повертається, ніколи не повторюється, тобто використовувані індекс і у ніколи не зменшуються.

2. Безперервність – послідовність рухається поступово: індекс і у збільшуються не більше ніж на 1 за один крок.

3. Межа – послідовність починається з лівого нижнього кута й закінчується у верхньому правому куті.

Оскільки метод зворотного програмування, який використовується в динамічному програмуванні, найбільш придатний для визначення основи послідовності, необхідно використовувати певний динамічний тип структури, який називається «стеком». Як і будь-який алгоритм динамічного програмування, DWT має поліноміальну складність. Коли ми маємо справу з більшими послідовностями, виникають дві незручності: запам’ятовування більших числових матриць і виконання великої кількості обчислень відхилень. У цьому випадку для вирішення двох вищезазначених проблем використовується вдосконалена версія алгоритму Fastdwt. Рішення полягає в тому, щоб розкласти матрицю станів на 2, 4, 8, 16 тощо. Повторюючи процес поділу вхідної послідовності на дві частини, виходить менша матриця. Тому обчислення відхилення виконується лише для цих малих матриць, а шлях деформації обчислюється для малих матриць.

3.2 Особливості оброблення

Розроблене програмне забезпечення для аналізу акустичних даних і розпізнавання акустичної інформації має такі режими роботи:

Рисунок 3.7 – Базова модель розпізнавання голосових команд.

Функція програми:

- Зберегти команду в аудіофайл;

- Додати команду до словника для подальшого використання; - Виконувати всі системні команди, зазначені в словнику;

- Порівняння з шаблонами;

- Виконувати команди з аудіофайлів.

Параметри, необхідні для запуску програми: - Хм <шлях акустичної моделі>. Якщо ви завантажите модель за посиланням вище, акустична модель буде розташована в папці zero_ru_cont_8k_v3 / zero_ru.cd_cont_4000. - dict <шлях до словника> - jsgf <синтаксичний шлях> - lm <шлях до мовної моделі> - logfn <шлях до файлу журналу>. За замовчуванням журнал записується на стандартний вихід. - infile <шлях до голосового файлу>. Вам потрібно переконатися, що частота дискретизації файлу відповідає частоті дискретизації моделі. - inmic <так|ні>. Звук у режимі реального часу з мікрофона. - remove_noise <так|ні>. Фільтрація шуму. За замовчуванням є. Спочатку аудіосигнал проходить попередню обробку та параметризацію, включаючи цифрову фільтрацію, сегментацію мовних сигналів за допомогою

кадрів, що перекриваються, перетворення спектру, обробку сигналу за допомогою віконних функцій (тобто вікон Хеммінга) та перетворення спектру. Після первинної обробки сигналу формується набір векторів ознак, який містить спектральні ознаки: коефіцієнти мель-частотного кепралу MFCC, які, у свою чергу, містять інформацію про те, які частини мови сигналу існують, і перетворюються в коефіцієнти, а потім вводять визнана одиниця. Потім перетворіть сигнал у вектор ознак. На етапі розпізнавання вхідне зображення порівнюється з існуючим шаблоном. Іншими словами, вектор ознак порівнюється з моделлю для обчислення ймовірності кожної моделі. Важливим кроком є параметризація, тому вона виконується оформити. Далі він визначається за допомогою алгоритму динамічного програмування. на закінчення: У цьому розділі вичерпно представлені алгоритми та методи аналізу акустичного сигналу (попередня обробка, параметризація, вибір і розпізнавання векторів ознак). Описується теорія ERP ГУівського ланцюга та її застосування в задачах розпізнавання мови: намальовано спосіб малювання прихованих ERP-подібних ГУівських моделей для мовних об’єктів (слів), розглянуто процес розпізнавання. Також представлено принципи та характеристики розробленого програмного забезпечення.

3.3 Оцінка якості аналізу мовленнєвого сигналу створеною системою

Виберіть такі критерії для перевірки розробленого програмного забезпечення: відстань, кут, шум. 1). Нормальна відстань від комп'ютера (0,3-1м). Кут = 0 градусів. Відсутність шуму.

Команда	Коефіцієнт розпізнавання
«Блокнот»	90%
«Ворд»	85%
«Ком’ютер»	89%
«Пейнт»	92%

Нормальна відстань від комп'ютера (0,3-1м). Кут = 45 градусів. Відсутність шуму.

Команда	Кут	Коефіцієнт розпізнавання
«Блокнот»	45°	85%
«Ворд»	45°	83%
«Ком’ютер»	45°	84%
«Пейнт»	45°	89%

Нормальна відстань від комп'ютера (0,3-1м). Кут = 45 градусів. Відсутність шуму.

Команда	Кут	Коефіцієнт розпізнавання
«Блокнот»	90°	82%
«Ворд»	90°	79%
«Ком’ютер»	90°	85%
«Пейнт»	90°	88%

Комп'ютер знаходиться на відстані 5 метрів. Кут = 0 градусів. Відсутність шуму.

Команда	Відстань	Коефіцієнт розпізнавання
«Блокнот»	5м	75%
«Ворд»	5м	72%
«Ком’ютер»	5м	80%
«Пейнт»	5м	79%

Нормальна відстань від комп'ютера (0,3-1м). Кут = 0 градусів. Шум = 100% (фоновий шум відповідає гучності команди користувача).

Команда	Шум	Коефіцієнт розпізнавання
«Блокнот»	100%	60%
«Ворд»	100%	42%
«Ком’ютер»	100%	53%
«Пейнт»	100%	50%

І тримайтеся на відстані 5 метрів від комп’ютера. Кут = 90 градусів. Шум = 100%.

Команда	Відстань	Кут	Шум	Коефіцієнт розпізнавання
«Блокнот»	5м	90°	100%	49%
«Ворд»	5м	90°	100%	32%
«Ком’ютер»	5м	90°	100%	45%
«Пейнт»	5м	90°	100%	43%

Можна зробити висновок, що на звичайній відстані від мікрофона кут майже не впливає на якість і точність розпізнавання команд; відстань - має певний вплив на якість, але безпосередньо залежить від якості мікрофона і Його фізичні характеристики. Шум є фактором, який найбільше впливає на розпізнавання мови. Демонстрація програми: сідайте. Шум = 100%.

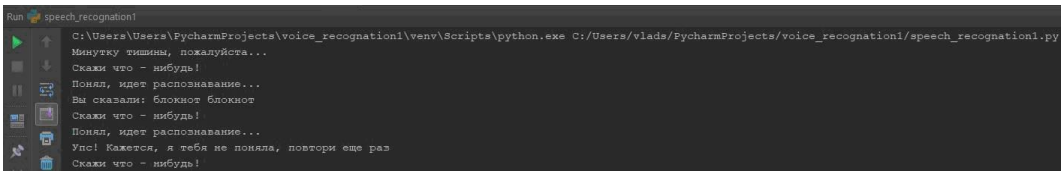


Рисунок 3.8 – Результат розпізнавання

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи. Магістерська кваліфікаційна робота з розробки та дослідження «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Підсистема параметризації.» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;

2) розраховано витрати на здійснення науково-технічної розробки;

3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Розробка застосунку для голосового управління типовими

операціями системи планування ресурсів підприємства. Підсистема параметризації.» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [51].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездат-ність продукту в реальних
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі влас-тивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Продовження таблиці 4.1

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансуван-ня

10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використо-вуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використо-вуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на вироб-ництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	3	3	3
2. Ринкові переваги (наявність аналогів)	3	3	3
3. Ринкові переваги (ціна продукту)	3	3	3

Продовження таблиці 4.2

Критерії	Експерт(ПІБ, посада)		
	1	2	2
	Бали:		
4. Ринкові переваги (технічні властивості)	2	2	2
5. Ринкові переваги (експлуатаційні витрати)	3	3	3
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	3	3	4
8. Практична здійсненність (наявність фахівців)	4	3	3
9. Практична здійсненність (наявність фінансів)	3	3	4
10. Практична здійсненність (необхідність нових матеріалів)	3	3	3
11. Практична здійсненність (термін реалізації)	3	3	4
12. Практична здійсненність (розробка документів)	3	3	3
Сума балів	36	35	38
Середньоарифметична сума балів $СБ_c$	36,3		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [51].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ ,розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Підсистема параметризації.» становить 36,3 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення розрахуємо за формулою [52]:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i \,, \tag{4.1}$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу *і-го* технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і при цьому має виконуватись умова $\sum_{i=1}^{k} \alpha_i = 1$;

β_i – відносне значення *і-го* технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ні}}{I_{ai}} \,, \tag{4.2}$$

де $I_{ні}$ та $I_{на}$ – чисельні значення конкретного *і*-го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ні}} \,; \tag{4.3}$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю-вання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Швидкість пошуку слів	мс	10	6	1,67	0,1
Обсяг кількості слів	шт.	30	80	1,33	0,15
Правильність розпізнавання слів	%	85	85	1	0,3
Знаходження потрібної інформації	%	80	90	1,13	0,25
Дружність інтерфейсу	бал	6	7	1,16	0,2

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення складе:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i = 1{,}67 \cdot 0{,}15 + 1{,}33 \cdot 0{,}1 + 1 \cdot 0{,}3 + 1{,}13 \cdot 0{,}25 + 1{,}16 \cdot 0{,}2 = 1{,}18.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,18 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Розробка застосунку для голосового управління типовими операціями

системи планування ресурсів підприємства. Ч. 1. Підсистема параметризації.», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [51]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.4)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=21 дні.

$$З_о = 12220{,}00 \cdot 34 / 21 = 19784{,}76 \; грн.$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	12220,00	581,90	34	19784,76
Інженер-програміст	11650,00	554,76	21	11650,00

Продовження таблиці 4.5

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Аналітик голосової параметризації	11900,00	566,67	9	5100,00
Інженер-проектувальник автоматизованих систем управління	11450,00	545,24	24	13085,71
Всього: 49620,48				

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Розробка застосунку для голосового управління типовими операціями. Підсистема параметризації» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i , \qquad (4.5)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}} , \qquad (4.6)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $Мм$=2379,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [51];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно T_p = 21 дн;

$t_{зм}$ – тривалість зміни, год.

$C_1 = 2379{,}00 \cdot 1{,}10 \cdot 1{,}65 / (21 \cdot 8) = 25{,}70$ *грн.*
$З_{р1} = 25{,}70 \cdot 6{,}00 = 154{,}21$ *грн.*

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Установка офісного обладнання	6,00	2	1,10	25,70	154,21
Підготовка робочого місця розробника ERP-подібної системи	8,00	3	1,35	31,54	252,34
Інсталяція програмного забезпечення розробки програмного модуля підсистеми параметризації	5,50	5	1,70	39,72	218,46
Формування інформаційної бази досліджень	12,00	4	1,50	35,05	420,57
Тренування системи розпізнавання мовленя	8,00	4	1,50	35,05	280,38
Всього					1325,97

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (4.7)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 11%.

$З_{дод}$ = (49620,48 + 1325,97) · 11 / 100% = 5604,11 грн.

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (4.8)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

Зн = (49620,48 + 1325,97 + 5604,11) · 22 / 100% = 12441,12 грн.

4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Підсистема параметризації.».

Витрати на матеріали (*М*), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$М = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{вj}, \quad (4.9)$$

де $Н_j$ – норма витрат матеріалу *j*-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу *j*-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = *1,1 … 1,15*);

$В_j$ – маса відходів *j*-го найменування, кг;

$Ц_{вj}$ – вартість відходів *j*-го найменування, грн/кг.

$M_1 = 3,00 \cdot 112,00 \cdot 1,11 - 0,000 \cdot 0,00 = 372,96$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір канцелярський офісний SKIPER-500 (A4)	112,00	3,00	-	-	372,96
Папір для заміток SKIPER-100 (A5)/70	65,00	3,00	-	-	216,45
Начиння канцелярське SKIPER-z12	165,00	4,00	-	-	732,60
Органайзер офісний SKIPER-Razor	210,00	4,00	-	-	932,40
Картридж для принтера HP-A2142	780,00	2,00	-	-	1731,60
Диск оптичний SKIPER-DVD	12,00	4,00	-	-	53,28
FLASH-пам'ять SKIPER-10c 32GB	315,00	1,00	-	-	349,65
Всього					4388,94

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Розробка застосунку для голосового управління типовими операціями. Підсистема параметризації» відсутні.

4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування

необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{спец} = \sum_{i=1}^{k} Ц_i \cdot C_{пр.i} \cdot K_i \ , \qquad (4.10)$$

де $Ц_i$ – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.i}$ –кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10…1,12$);

k – кількість найменувань устаткування.

$B_{спец} = 12450,00 \cdot 1 \cdot 1,1 = 13695,00$ *грн.*

Отримані результати зведемо до таблиці:

Таблиця 4.8 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Синтезатор мови цифровий програмований	1	12450,00	13695,00
Блок інтерфейсний Rapid-ZX2020	1	1670,00	1837,00
Всього			15532,00

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (4.11)$$

де $Ц_{іпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$С_{прг.і}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$К_і$ – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($К_і$ *= 1,10…1,12);*

k – кількість найменувань програмних засобів.

$В_{прг} = 5530,00 \cdot 1 \cdot 1,11 = 6138,30$ *грн.*

Отримані результати зведемо до таблиці:

Таблиця 4.9 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
ОС Windows 11	1	5530,00	6138,30
Прикладний пакет Microsoft Office 2019	1	4370,00	4850,70
Прикладний пакет MATLab Project	1	6800,00	7548,00
Всього			18537,00

4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$А_{обл} = \frac{Ц_б}{Т_в} \cdot \frac{t_{вик}}{12}, \quad (4.12)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$Т_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$А_{обл} = (25890,00 \cdot 2) / (2 \cdot 12) = 2157,50$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.10 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Програмно-аналітичний комплекс	25890,00	2	2	2157,50
Графічно-обчислювальний комплекс обробки даних	21460,00	2	2	1788,33
Програмне забезпечення розробки підсистеми параметризації	11300,00	2	2	941,67
Обладнання виводу інформації	7800,00	4	2	325,00
Обладнання вводу голосової інформації	6700,00	4	2	279,17
Офісна оргтехніка	9460,00	5	2	315,33
Приміщення дослідницької лабораторії	322000,00	20	2	2683,33
Всього				8490,33

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вті}}{\eta_i}, \qquad (4.13)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_е$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_е$ = 4,25 грн;

$К_{вп і}$ – коефіцієнт, що враховує використання потужності, $К_{вп і} < 1$;

$\eta_і$ – коефіцієнт корисної дії обладнання, $\eta_і < 1$.

$В_е = 0,20 \cdot 240,0 \cdot 4,25 \cdot 0,95 / 0,97 = 204,00$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.11 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Програмно-аналітичний комплекс	0,20	240,0	204,00
Графічно-обчислювальний комплекс обробки даних	0,65	200,0	552,50
Синтезатор мови цифровий програмований	0,01	10,0	0,43
Обладнання виводу інформації	0,40	25,0	42,50
Обладнання вводу голосової інформації	0,02	40,0	3,40
Офісна оргтехніка	0,60	40,0	102,00
Всього			904,83

4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Розробка застосунку для голосового управління типовими операціями. Підсистема параметризації» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов’язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з’їзди, конференції, наради, пов’язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_о + З_р) \cdot \frac{Н_{св}}{100\%}, \quad (4.14)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св}$ = 25%.

$В_{св}$ = (49620,48 + 1325,97) · 25 / 100% = 12736,61 грн.

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30…45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{сп} = (З_о + З_р) \cdot \frac{Н_{сп}}{100\%}, \quad (4.15)$$

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо $Н_{сп}$= 35%.

$В_{сп}$ = (49620,48 + 1325,97) · 35 / 100% = 17831,26 грн.

4.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.16)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 55%.

$І_в$ = (49620,48 + 1325,97) · 55 / 100% = 28020,55 грн.

4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.17)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 105%.

$В_{нзв}$ = (49620,48 + 1325,97) · 105 / 100% = 53493,77 грн.

Витрати на проведення науково-дослідної роботи на тему «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Підсистема параметризації.» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + В_{спец} + В_{прг} + А_{обл} + В_е + В_{св} + В_{сп} + І_в + В_{нзв}. \quad (4.18)$$

$В_{заг}$= 49620,48 +1325,97 +5604,11 +12441,12311 +4388,94 +0,00 +15532,00 + 18537,00 + 8490,33 +904,83 +12736,61 +17831,26 +28020,55 +53493,77 = 228926,97 грн.

Загальні витрати *ЗВ* на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.19)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,9.

ЗВ = 228926,97 / 0,9 = 254363,30 грн.

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Підсистема параметризації.» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	500	750	1500	2000

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, приймемо 6500 осіб;

$Ц_б$ – вартість програмного продукту у році до впровадження результатів розробки, приймемо 7850,00 грн;

$\pm\Delta Ц_о$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, приймемо 150,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних

результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [51]:

$$\Delta П_i = (\pm \Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.20)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2021 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо $\rho = 25\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2021 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$\Delta П_1 = (150,00 \cdot 6500,00 + 8000,00 \cdot 500) \cdot 0,83 \cdot 0,25 \cdot (1\text{-}0,18/100\%) = 846496,25$ грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2 = (150,00 \cdot 6500,00 + 8000,00 \cdot 1250) \cdot 0,83 \cdot 0,25 \cdot (1\text{-}0,18/100\%) = 1867396,25$ грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3 = (150,00 \cdot 6500,00 + 8000,00 \cdot 2750) \cdot 0,83 \cdot 0,25 \cdot (1\text{-}0,18/100\%) = 3909196,25$ грн.

Збільшення чистого прибутку 4-го року:

$\Delta П_4 = (150,00 \cdot 6500,00 + 8000,00 \cdot 4750) \cdot 0,83 \cdot 0,25 \cdot (1\text{-}0,18/100\%) = 6631596,25$ грн.

Приведена вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.21)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ=0,12;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$ПП$ =846496,25/(1+0,12)^1+1867396,25/(1+0,12)^2+3909196,25/(1+0,12)^3+ +6631596,25/(1+0,12)^4=755800,22+1488676,86+2782488,68+4214499,31=9241465,06 грн.

Величина початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \tag{4.22}$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=2,5;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 254363,30 грн.

$PV = k_{інв} \cdot ЗВ$ = 2,5 · 254363,30 = 635908,25 грн.

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \tag{4.23}$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 9241465,06 грн;

PV – теперішня вартість початкових інвестицій, 635908,25 грн.

$E_{абс} = ПП - PV$ = 9241465,06 - 635908,25 = 8605556,81 грн.

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.24)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 8605556,81 грн;

PV – теперішня вартість початкових інвестицій, 635908,25 грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1$ = (1+8605556,81/635908,25)$^{1/4}$= 0,95.

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (4.25)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2021 році в Україні d =0,1;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,15.

$\tau_{мін}$ = 0,1+0,15 = 0,25 < 0,95 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Підсистема параметризації.» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \tag{4.26}$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 0,95 = 1,05 р.

$T_{ок}$ < 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Підсистема параметризації.» становить 36,3 бала, що свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,18 рази.

Також термін окупності становить 1,05 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Підсистема параметризації.».

ВИСНОВКИ

За результатами магістерської роботи проаналізовано процес аналізу акустичних даних та розпізнавання акустичної інформації та розроблено програмне рішення цієї проблеми. Проаналізовано представлення та розпізнавання акустичної інформації та визначено основні компоненти системи автоматичного розпізнавання акустичної інформації (команд):

- Попередня обробка звукових сигналів;
- Перетворення сигналу у вектор ознак;
- Розпізнавати звукову інформацію (класифікація).

Розроблено та розглянуто метод попередньої обробки та виділення ознак мовленнєвого сигналу, серед якого обрано один із найбільш популярних та корисних методів, який базується на знаходженні коефіцієнта збереження Мела (MFCC). Враховуючи метод акустичного розпізнавання інформації, обрано метод динамічного програмування. В описі цих методів наведено їх короткий опис, класифікацію, завдання, які вони вирішують (попередня обробка та розпізнавання), алгоритми побудови систем розпізнавання на основі цих методів та їх застосування. Тому ефективна система розпізнавання повинна включати такі етапи обробки вхідного сигналу, параметризації та розпізнавання. Розроблено програмний комплекс, що дозволяє створити базу даних голосових команд і включає реалізацію всіх вищезгаданих методів і алгоритмів аналізу та розпізнавання акустичної інформації. Розглянута модель розпізнавання голосових команд має на меті створення мовного інтерфейсу, який дозволить істотно підвищити ефективність роботи людино-машинної системи. Модель заснована на використанні частотного аналізу мовних сигналів, особливо перетворюючих Фур'є, це забезпечить високошвидкісне програмне забезпечення. Класифікація мовних команд здійснюється на основі динамічного алгоритму за часом, який обробляє кожну групу команд і забезпечує середнє значення

подібності, що дозволяє отримати більш високий коефіцієнт точності в порівнянні з іншими системами розпізнавання команд. Проводив експериментальні дослідження з розпізнавання мовлення та різних технічних звуків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Аграновский, А.В. Теоретические аспекты алгоритмов обработки и классификации речевых сигналов/ А.В. Аграновский, Д.А. Леднов – М.: Радио и связь, 2006. – 164 с.
2. Винцюк, Т.К., Анализ, распознавание и интерпретация речевых сигналов/ Т.К. Винцюк - Киев: Наук.думка, 2005. - 264с.
3. Голд, Б. Цифровая обработка сигналов/ Б. Голд, Ч. Рейдер. – М.: Сов. Радио, 1973. – 368 с.
4. Сорокин В.Н., Цыплихин А.И. Сегментация и распознавание гласных. // Информационные процессы, т. 4 , № 2, с. 202-220, 2005 г.
5. Жиляков, Е.Г. Вариационные методы анализа и построения функций по эмпирическим данным/Е.Г. Жиляков. – Белгород: Изд-во БелГУ, 2007. -160 с.
6. Жиляков, Е.Г. Методы обработки речевых данных в информационно-телекоммуникационных системах на основе частотных представлений/ Е.Г. Жиляков, С.П. Белов, Е.И. Прохоренко. – Белгород: Изд-во БелГУ, 2007. - 136 с.
7. Жиляков, Е.Г. Сегментация речевых сигналов на основе анализа распределения энергии по частотным интервалам/ Е.Г. Жиляков, Е.И. Прохоренко, А.В. Болдышев, А.А. Фирсова, М.В. Фатова // Научные ведомости Белгородского государственного университета. Серия: История. Политология. Экономика. Информатика, Том 18 – 2011. - №7-1 (102). – С. 187- 196
8. Кипяткова, И.С. Автоматическая обработка разговорной русской речи: монография / И.С. Кипяткова, А.Л. Ронжин, А.А. Карпов. СПИИРАН. − СПб.: ГУАП, 2013. − 314 с.
9. Куприянов, М.С. Цифровая обработка сигналов: процессоры, алгоритмы, средства проектирования/ М.С. Куприянов. – СПб.: Политехника, 1999. – 592 с.

10. Лайонс, Р. Цифровая обработка сигналов / Лайонс Р; - 2-е изд. ; Пер. с англ. − М.: ООО "Бином-Пресс", 2006 − 656 с.: ил.

11. Марпл-мл, С.Л. Цифровой спектральный анализ и его приложения / Марпл-мл. С.Л.; Пер. с англ. −М.: Мир, 2005.

12. Михайлов В.Г., Златоусов Л.В. Измерение параметров речи/ В.Г. Михайлов, Л.В. Златоусова; Под.ред. М.А. Сапожникова. – Москва: Радио и связь, 1987. – 168с.: ил.

13. Рабинер Л. Р., Шафер Р.В. Цифровая обработка речевых сигналов/ Л.Р. Рабинер, Р.В. Шафер.; Пер. с англ.М.В. Назарова, Ю.Н. Прохорова; Под ред. М.В.Назарова, Ю.Н. Прохорова. – Москва: Радио и связь, 1981. – 496с.:ил.

15. Сергиенко А.Б. Цифровая обработка сигналов: учебное пособие. — 3-е изд.— М.: БХВ-Петербург, 2011. — 768 с.

16. Солонина, А.И. Основы цифровой обработки сигналов/ А.И. Солонина, Д.А. Улахович, С.М. Арбузов, Е.Б.Соловьева. – СПб.: БХВ- Петербург, 2005. – 768с.

17. Ю.Лабунец В. Г. Алгебраическая теория сигналов и систем. Красноярск: Изд-во Краснояр. ун-та, 1984.

18. Цемелъ Г. И. Опознавание речевых сигналов. М., Наука, 2006.

19. Шелепов В.Ю. Новые алгоритмы сегментации речевого сигнала и распознавания некоторых классов фонем / В.Ю. Шелепов, А.В. Ниценко // Искусственный интеллект. – 2007. – № 1. – С. 213-224.

20. Шелепов В.Ю. Новые алгоритмы распознавания фонем и их классов, поиск слова по его смешанной транскрипции при распознавании слов большого словаря / В.Ю. Шелепов, А.В. Ниценко, А.В. Жук // Искусственный интеллект. – 2007. – № 2. – С. 139-147.

21. Шелепов В.Ю. О распознавании фонем с помощью анализа речевого сигнала в частотной и временной областях. Приложение к распознаванию синтаксически связных фраз / В.Ю. Шелепов, А.В. Ниценко, А.В. Жук, Д.С. Азаренко // Речевые технологии. – 2008.– № 2. – С. 43-52

22. Жиляков Е.Г., Фирсова А.А.. Сегментация речевых сигналов на основе субполосного анализа // Вестник НТУ "ХПИ", № 39 (1012). - 2013г. - С.73-81.
23. Фирсова А.А. Разработка и исследование субполосных методов и алгоритмов сегментации речевых сигналов / Фирсова А.А.//Автореферат диссертации на соискание ученой степени кандидата технических наук. 17 мая 2013 г. - Белгород. - С.15-19.
24. Музычук Д.С. Сегментация, шумоподавление и фонетический анализ в задаче распознавания речи/ Музычук Д.С., Медведев М.С. // Молодой ученый. - 2013. - №6. - С. 86-96.
25. Дремин И.М., Иванов О.В., Нечитайло В.А. Вейвлеты и их использование. //Успехи физических наук, т. 171, №5 с. 465-500, 2001 г.
26. Ермоленко Т.Н. Алгоритмы сегментации с применением быстрого вейвлет- преобразования / Т.Н. Ермоленко, В.И. Шевчук // Статьи, принятые к публикации на сайте международной конференции Диалог 2003.
27. Шелухин О.И. Цифровая обработка и передача речи/ Лукьянцев Н.Ф..– М.: Радио и связь, 2005.– 454 с.
28. Жиляков, Е.Г. Вариационные методы частотного анализа звуковых сигналов/ Белов С.П., Прохоренко Е.И. // Труды учебных заведений связи. СПб, 2006.-№ 174.-С.163-170.
29. Жиляков Е.Г. Бабаринов С.Л. Чадюк П.В. Исследование сервиса компании Google Inc. по распознаванию русской речи / Жиляков Е.Г. Бабаринов С.Л. Чадюк П.В // Научные ведомости Белгородского государственного университета. Серия: История. Политология. Экономика. Информатика № 15-1 (158)/ - том 27. - 2013
30. Жиляков Е.Г. Вариационные методы анализа и построения функций по эмпирическим данным на основе частотных представлений. - Белгород: Изд-во БелГУ, 2007. - с. 160
31. Цыплихин А.И. Системный анализ, управление и обработка информации,- 2006.

32. Фирсова А.А. О различиях распределения энергии звуков русской речи и шума / А.В. Болдышев, А.А. Фирсова // Материалы 12-ой Международной конференции и выставки "Цифровая обработка сигналов и еѐ применение. – "DSPA'2010". – Москва. – 2010. – С. 204–207.

33. Сорокин В.К. Синтез речи. М. : Наука, 1992. С. 392

34. Цыплихин А.И. Анализ и автоматическая сегментация речевого сигнала: дис. канд. тех. наук / ИППИ РАН. – М., 2006. – 149 с.

35. А. С. Колоколов. Обработка сигнала в частотной области при распознавании речи.- 2006. – с. 13–18

36. Конев А. А. Параметрическое описание сегментов речевого сигнала / В. И. Голубев, А. А. Конев // Научная сессия ТУСУР – 2005: Материалы Всероссийской научно-технической конференции студентов, аспирантов и молодых специалистов – Томск: Издательство ТУСУРа, 2005. – С. 113- 116.

37. Кочаров Д.А. Автоматическая интерпретация звуков речи // Диссертационная работа.- СПбГУ 2008

38. Утробин В.А., Гай В.Е. Алгоритм выделения вокализованных участков речевого сигнала // Вестник Нижегородского университета им. Н.И. Лобачевского, 2012, № 6 (1). С. 175–179

39. Мясникова Е.Н. Объективное распознавание звуков речи Л.: «Энергия», 1967. - 150 с.

40. Черник, Н. Н. Сегментация спонтанной речи в языках различных типов / Н. Н. Черник // Вестник Белорусского государственного экономического университета. - 2009. - N 4. - С. 101-107.

41. Ли У. А. Методы автоматического распознавания речи. М., Мир, 1983.

42. Огнев И.В., Огнев А.И., Парамонов П.А., Классификация речевых образов на основе анализа распределений их локальных экстремумов //131труды XXI международной научно-технической конференции"Информационные средства и технологии". - М.: МЭИ, 2013 - с. 53-57.

43. Винцюк Т.К., Анализ, распознавание и интерпретация речевых сигналов, — Киев: Наукова думка, 1987. – 264 стр.

44. Бондаренко Л. В., Вербицкая Л. А., Гордина М. В., Основы общей фонетики. – М.: Академия, 2004. – 160 с.

45. Шарий Т.В., О проблеме параметризации речевого сигнала в современных системах распознавания речи // Вісник Донецького Національного Универсітету, 2008, вып. 2, стр. 536-541

46. Маркел Дж. Д., Грэй А. Х., Линейное предсказание речи. – М.: Связь,1980. – 308 с.

47. Загоруйко Н. Г., Методы распознавания и их применение. – М.: Советское радио, 2006. – 208 с.

48. Агашин О.С., Корелин О.Н., Методы цифровой обработки речевого сигнала в задаче распознавания изолированных слов с применением сигнальных процессоров // Труды Нижегородского государственного технического университета им. Р.Е. Алексеева № 4, 2012, с. 32-44

49. Ронжин А.Л., Ли И. В. Автоматическое распознавание русской речи //Вестник Российской академии наук, 2007, том 77, № 2, с. 133-138.

50. Огнев И. В., Парамонов П.А. Исследование способов представления числа для реализации арифметических операций в ассоциативной среде с командным управлением // Информационные средства и технологии: труды Международной научно-технической конференции (19 – 21 октября 2010 г.): в 3 т. – М.: МЭИ, 2010. – 1 т. – с. 54-60.

51. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

52. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

ДОДАТКИ

Додаток А

ЗАТВЕРДЖУЮ

Завідувач кафедрою КСУ

д.т.н., проф.

Володимир Дубовой.

«30» 09 2021 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

Розробка застосунку для голосового управління типовими операціями системи планування ресурсів підприємства. Ч. 1. Підсистема параметризації.

08-01.МКР.003.00.000 ТЗ

Керівник: д.т.н., професор каф. КСУ

В'ячеслав Ковтун

«_____» _____ 202_р.

Виконав: студентка 2 курсу, групи 2АКІТ-20м

_____Людмила Дихніч

«_____» _____ 202_р.

Вінниця 2021

1. Назва та галузь застосування

Розробка застосунку для голосового управління типовими операціями. Ч. 1. Підсистема параметризації.

Розроблена архітектура, модель і система для підвищення ефективності параметризації моного сигналу.

Розробка призначена для використання в галузях інформаційних технологій.

2. Підстави для розробки

Розробку системи здійснювати на підставі наказу по університету №____ від _____ та завдання до магістерської кваліфікаційної роботи складеного та затвердженого кафедрою КСУ.

3. Мета та призначення розробки

Метою дипломної дисертації є підвищення ефективності параметризації моного сигналу. Для досягнення поставленої мети необхідно розробити архітектуру, модель і саму відповідну систему.

4. Джерела розробки

1. CO-ResNet: Optimized ResNet model for COVID-19 diagnosis from X-ray images / S. Bharati et al. International Journal of Hybrid Intelligent Systems. 2021. Vol. 17. P. 71–85.
2. Nibali A., He Z., Wollersheim D. Pulmonary Nodule Classification with Deep Residual Networks // International Journal of Computer Assisted Radiology and Surgery. 2017. Vol. 12. P. 1799–1808.
3. Bottou L. Large-Scale Machine Learning with Stochastic Gradient Descent // Proceedings of COMPSTAT2010. 2010. P. 177–186.
4. He S., Wu Q.H., Saunders J.R. A Group Search Optimizer for Neural Network Training // Computational Science and Its Applications - ICCSA 2006 Lecture Notes in Computer Science. 2006. P. 934–943.

5. Rozložník M. Solution Approaches for Saddle-Point Problems // Nečas Center Series Saddle-Point Problems and Their Iterative Solution. 2018. P. 33–39.

5. Показники призначення

Вихідними даними для обробки є результати аналізу об’єкта дослідження.

Результатом роботи методу є ефективності параметризації мовного сигналу.

6. Стадії розробки

1. Огляд предметної області має бути виконаний до «22» 09 2021 р.
2. Розробка математичного апарату має бути виконана до «30» 10 2021 р.
3. Розробка програмного забезпечення та експериментальні дослідження має бути виконана до «12» 11 2021 р.
4. Підготовка економічної частини має бути виконана до «08» 12 2021р.
5. Оформлення пояснювальної записки і графічного матеріалу має бути виконано до «17» 12 2021р.

7. Порядок контролю та приймання

1. Рубіжний контроль. Провести до «___» _____202_ р.
2. Попередній захист магістерської кваліфікаційної роботи. Провести до «17» 12 2021 р.
3. Захист магістерської кваліфікаційної роботи. Провести «22» 12 2021 р.

Додаток Б

Фрагменти програмного коду

```
import speech_recognition as sr from gtts import gTTS

import pygame
from pygame import mixer mixer.init()
import os import sys import time import datetime import
logging
import webbrowser import subprocess

class Speech_AI: def       init         (self):
self._recognizer = sr.Recognizer() self._microphone =
sr.Microphone()

now_time = datetime.datetime.now()
self._mp3_name =
now_time.strftime("%d%m%Y%I%M%S") + ".mp3"
self._mp3_nameold = '111'

def work(self):
print("Минутку тишины, пожалуйста...") with
self._microphone as source:
self._recognizer.adjust_for_ambient_noise(source)

try:
while True:
print("Скажи что - нибудь!") with self._microphone as
source:
audio = self._recognizer.listen(source) print("Понял, идет
распознавание...") try:
statement = self._recognizer.recognize_google(audio,
language="ru_RU")
statement = statement.lower()

# Команды для открытия различных внешних
приложений

if ((statement.find("калькулятор") != -1) or
(statement.find("calculator") != -1)):
self.osrun('calc')



!= -1)):
```

```
1)):


!= -1)):




!= -1) or (

-1)) and (




!= -1) or (

if ((statement.find("блокнот") != -1) or
(statement.find("notepad") self.osrun('notepad')
if ((statement.find("paint") != -1) or (statement.find("паинт")
!= - self.osrun('mspaint')
if ((statement.find("browser") != -1) or
(statement.find("браузер") self.openurl('http://google.ru',
'Открываю браузер')
# Команды для открытия URL в браузере
if (((statement.find("youtube") != -1) or
(statement.find("youtub") statement.find("ютуб") != -1) or
(statement.find("you tube") != statement.find("смотреть")
== -1)):
self.openurl('http://youtube.com', 'Открываю ютуб')

if (((statement.find("новости") != -1) or
(statement.find("новость")

statement.find("на усть") != -1)) and (
(statement.find("youtube") == -1) and

(statement.find("youtub") != -1) and (
statement.find("ютуб") == -1) and (statement.find("you
tube")

== -1))):


self.openurl('https://www.youtube.com/user/rtrussian/videos',

'Открываю новости')
```

```
почту')

if ((statement.find("mail") != -1) or (statement.find("майл")
!= -1)): self.openurl('https://e.mail.ru/messages/inbox/',
'Открываю



if ((statement.find("вконтакте") != -1) or (statement.find("в
контакте") != -1)):
self.openurl('http://vk.com', 'Открываю Вконтакте')

# Команды для поиска в сети интернет



1) or (

1) or (

if ((statement.find("найти") != -1) or
(statement.find("поиск") != - statement.find("найди") != -1)
or (statement.find("дайте") != -
statement.find("mighty") != -1)): statement =
statement.replace('найди', '') statement =
statement.replace('найти', '') statement = statement.strip()
self.openurl('https://yandex.ru/yandsearch?text=' + statement,
"Я

нашла следующие результаты")




1))):

if ((statement.find("смотреть") != -1) and (
(statement.find("фильм") != -1) or (statement.find("film") !=
-

statement = statement.replace('посмотреть', '') statement =
statement.replace('смотреть', '') statement =
statement.replace('хочу', '') statement =
statement.replace('фильм', '') statement =
statement.replace('film', '') statement = statement.strip()



self.openurl('https://yandex.ru/yandsearch?text=Смотреть+о
нлайн+фильм+' + statement,
```

```
"Выберите сайт где смотреть фильм")



-1) or (

if (((statement.find("youtube") != -1) or
(statement.find("ютуб") !=

statement.find("you tube") != -1)) and

(statement.find("смотреть") != -1)):
statement = statement.replace('хочу', '') statement =
statement.replace('на ютубе', '') statement =
statement.replace('на ютуб', '') statement =
statement.replace('на youtube', '') statement =
statement.replace('на you tube', '') statement =
statement.replace('на youtub', '') statement =
statement.replace('youtube', '') statement =
statement.replace('ютуб', '') statement =
statement.replace('ютубе', '') statement =
statement.replace('посмотреть', '') statement =
statement.replace('смотреть', '') statement = statement.strip()

self.openurl('http://www.youtube.com/results?search_query='
+ statement, 'Ищу в ютуб')



-1)):

if ((statement.find("слушать") != -1) and
(statement.find("песн") !=

statement = statement.replace('песню', '') statement =
statement.replace('песни', '') statement =
statement.replace('песня', '') statement =
statement.replace('песней', '') statement =
statement.replace('послушать', '') statement =
statement.replace('слушать', '') statement =
statement.replace('хочу', '') statement = statement.strip()
self.openurl('https://my.mail.ru/music/search/' + statement,

"Нажмите плэй")

# Поддержание диалога

if ((statement.find("до свидания") != -1) or
(statement.find("досвидания") != -1)):
answer = "Пока!" self.say(str(answer))
while pygame.mixer.music.get_busy(): time.sleep(0.1)
sys.exit()
```

```
print("Вы сказали: {}".format(statement)) except
sr.UnknownValueError:
print("Упс! Кажется, я тебя не поняла, повтори еще раз")
except sr.RequestError as e:
print("Не могу получить данные от сервиса Google Speech
Recognition; {0}".format(e))
except KeyboardInterrupt: self._clean_up() print("Пока!")

def osrun(self, cmd):
PIPE = subprocess.PIPE
p = subprocess.Popen(cmd, shell=True, stdin=PIPE,
stdout=PIPE, stderr=subprocess.STDOUT)

def openurl(self, url, ans): webbrowser.open(url)

self.say(str(ans))
while pygame.mixer.music.get_busy(): time.sleep(0.1)

def say(self, phrase):
tts = gTTS(text=phrase, lang="ru") tts.save(self._mp3_name)

# Play answer mixer.music.load(self._mp3_name)
mixer.music.play()
if (os.path.exists(self._mp3_nameold)):
os.remove(self._mp3_nameold)

now_time = datetime.datetime.now() self._mp3_nameold =
self._mp3_name
self._mp3_name =
now_time.strftime("%d%m%Y%I%M%S") + ".mp3"

def _clean_up(self): def clean_up():
os.remove(self._mp3_name)


def main():
ai = Speech_AI() ai.work()


main()



Тестувальна частина import
os
import unittest


import speech_recognition as sr
```

```
class TestRecognition(unittest.TestCase): def setUp(self):

self.AUDIO_FILE_EN =
os.path.join(os.path.dirname(os.path.realpath(        file
        )), "english.wav")
self.AUDIO_FILE_FR =
os.path.join(os.path.dirname(os.path.realpath(        file
        )), "french.aiff")
self.AUDIO_FILE_ZH =
os.path.join(os.path.dirname(os.path.realpath(        file
        )), "chinese.flac")


def test_sphinx_english(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_EN) as source: audio =
r.record(source)
self.assertEqual(r.recognize_sphinx(audio), "wanted to
three")


def test_google_english(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_EN) as source: audio =
r.record(source)
self.assertIn(r.recognize_google(audio), ["1 2 3", "one two
three"])


def test_google_french(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_FR) as source: audio =
r.record(source)
self.assertEqual(r.recognize_google(audio, language="fr-
FR"), u"et c'est la dictée numéro 1")


def test_google_chinese(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_ZH) as source: audio =
r.record(source)
self.assertEqual(r.recognize_google(audio, language="zh-
CN"), u"砸自己的脚")


@unittest.skipUnless("WIT_AI_KEY" in os.environ,
"requires Wit.ai key to be specified in WIT_AI_KEY
environment variable")

def test_wit_english(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_EN) as source: audio =
r.record(source)
self.assertEqual(r.recognize_wit(audio,
key=os.environ["WIT_AI_KEY"]), "one two three")
```

@unittest.skipUnless("BING_KEY" in os.environ, "requires Microsoft Bing Voice Recognition key to be specified in BING_KEY environment variable")
def test_bing_english(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_EN) as source: audio = r.record(source)
self.assertEqual(r.recognize_bing(audio, key=os.environ["BING_KEY"]), "123.")

@unittest.skipUnless("BING_KEY" in os.environ, "requires Microsoft Bing Voice Recognition key to be specified in BING_KEY environment variable")
def test_bing_french(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_FR) as source: audio = r.record(source)
self.assertEqual(r.recognize_bing(audio, key=os.environ["BING_KEY"], language="fr-FR"), u"Essaye la dictée numéro un.")

@unittest.skipUnless("BING_KEY" in os.environ, "requires Microsoft Bing Voice Recognition key to be specified in BING_KEY environment variable")
def test_bing_chinese(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_ZH) as source: audio = r.record(source)
self.assertEqual(r.recognize_bing(audio, key=os.environ["BING_KEY"], language="zh-CN"), u"砸自己的脚
。")

@unittest.skipUnless("HOUNDIFY_CLIENT_ID" in os.environ and "HOUNDIFY_CLIENT_KEY" in os.environ, "requires Houndify client ID and client key to be specified in HOUNDIFY_CLIENT_ID and HOUNDIFY_CLIENT_KEY environment variables")
def test_houndify_english(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_EN) as source: audio = r.record(source)
self.assertEqual(r.recognize_houndify(audio, client_id=os.environ["HOUNDIFY_CLIENT_ID"], client_key=os.environ["HOUNDIFY_CLIENT_KEY"]), "one two three")

@unittest.skipUnless("IBM_USERNAME" in os.environ and "IBM_PASSWORD" in os.environ, "requires IBM Speech to Text username and password to be specified in IBM_USERNAME and IBM_PASSWORD environment variables")

```
def test_ibm_english(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_EN) as source: audio =
r.record(source)
self.assertEqual(r.recognize_ibm(audio,
username=os.environ["IBM_USERNAME"],
password=os.environ["IBM_PASSWORD"]), "one two three
")


@unittest.skipUnless("IBM_USERNAME" in os.environ and
"IBM_PASSWORD" in os.environ, "requires IBM Speech to
Text username and password to be specified in
IBM_USERNAME and IBM_PASSWORD environment
variables")
def test_ibm_french(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_FR) as source: audio =
r.record(source)
self.assertEqual(r.recognize_ibm(audio,
username=os.environ["IBM_USERNAME"],
password=os.environ["IBM_PASSWORD"], language="fr-
FR"), u"si la dictée numéro un ")


@unittest.skipUnless("IBM_USERNAME" in os.environ and
"IBM_PASSWORD" in os.environ, "requires IBM Speech to
Text

username and password to be specified in
IBM_USERNAME and IBM_PASSWORD environment
variables")
def test_ibm_chinese(self): r = sr.Recognizer()
with sr.AudioFile(self.AUDIO_FILE_ZH) as source: audio =
r.record(source)
self.assertEqual(r.recognize_ibm(audio,
username=os.environ["IBM_USERNAME"],
password=os.environ["IBM_PASSWORD"],
if        name__ == "       main      ": unittest.main()
```

Додаток В

ІЛЮСТРАТИВНА ЧАСТИНА

РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ГОЛОСОВОГО УПРАВЛІННЯ ТИПОВИМИ ОПЕРАЦІЯМИ СИСТЕМИ ПЛАНУВАННЯ РЕСУРСІВ ПІДПРИЄМСТВА. Ч. 1. ПІДСИСТЕМА ПАРАМЕТРИЗАЦІЇ.

Перелік ілюстративних матеріалів:

1. Компоненти систем розпізнавання мови.
2. Етапи попередньої обробки мовного сигналу
3. Вікно Хеммінга та його спектр
4. Сегментація мовного сигналу
5. Визначення відстані між двома послідовностями
6. Результати розпізнавання
7. Загальна схема роботи з .wav файлами
8. Деталізована схема роботи програми
9. Взаємодія класів
10. Загальна схема роботи програми з голосовою командою

Виконав: студентка 2-го курсу, групи 2АКІТ-20м
спеціальності 151 –Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва спеціальності)

_____Людмила Дихніч
(ім'я та прізвище)

Керівник: д.т.н., професор каф. КСУ

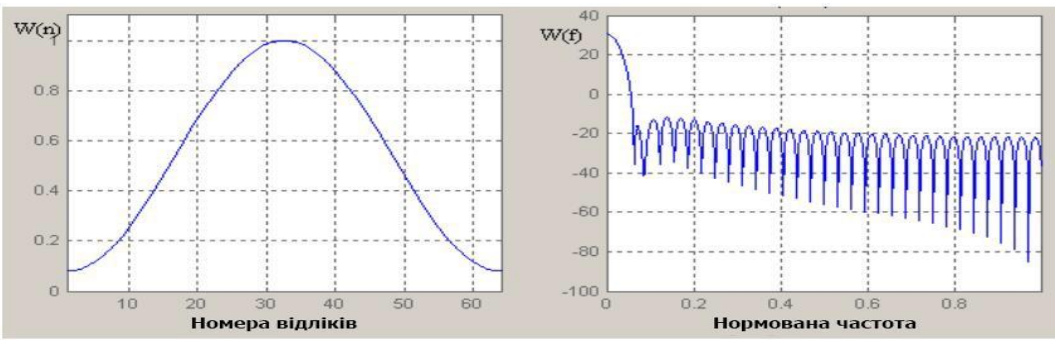
_____В'ячеслав Ковтун
(ім'я та прізвище)
«____» _____ 2021 р.

Опонент: к.т.н., доцент каф. АІІТ

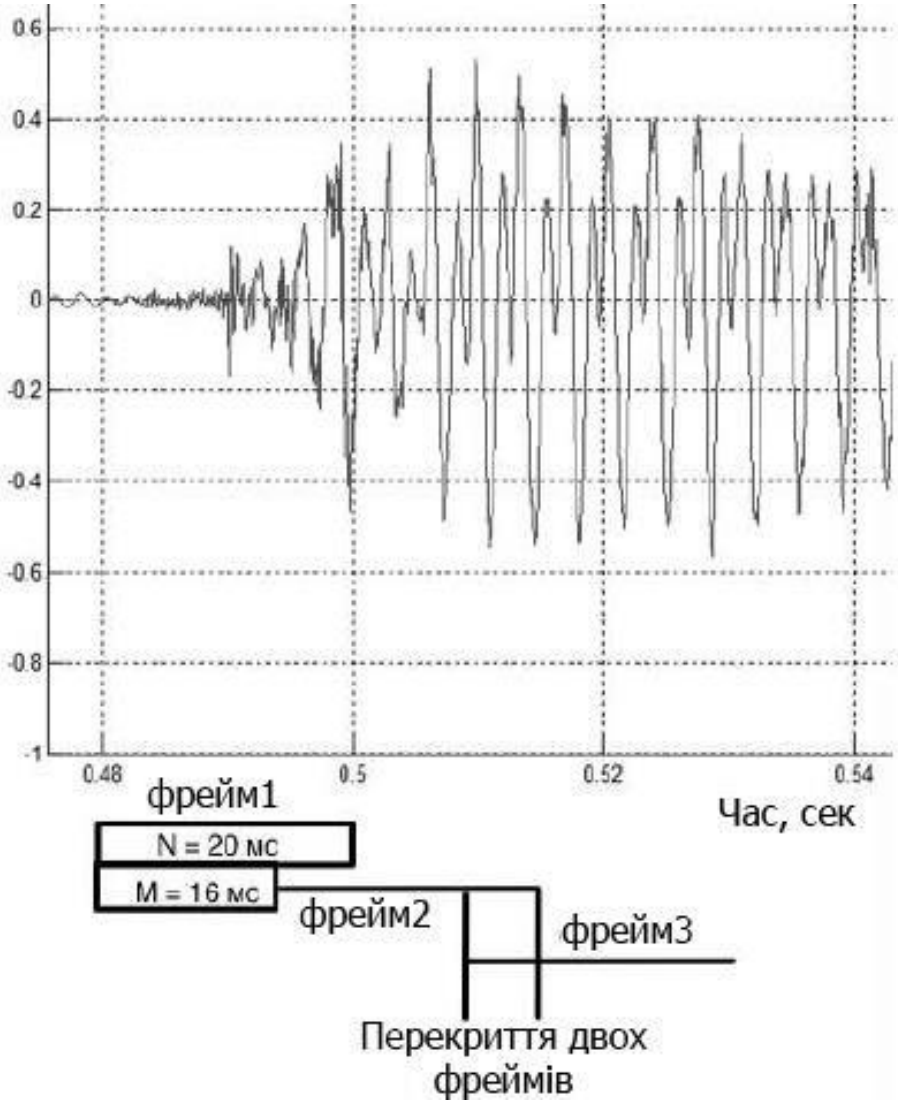
_____Володимир Гармаш
(ім'я та прізвище)
«____» _____ 2021 р.

Компоненти систем розпізнавання мови

Етапи попередньої обробки мовного сигналу



Вікно Хеммінга та його спектр

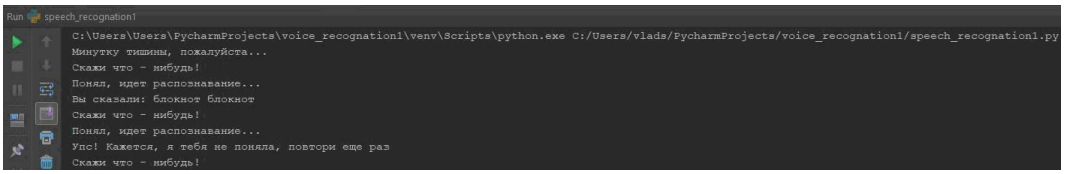


Сегментація мовного сигналу

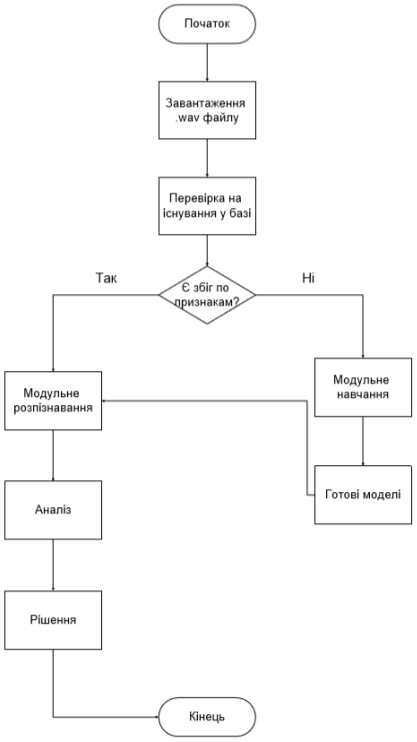
	-2	10	-10	15	-13	20	-5	14	2
3	5	12	25	37	53	70	78	89	90
-13	16	28	15	43	37	70	78	105	104
14	32	20	39	16	43	43	62	62	74
-7	37	37	23	38	22	49	45	66	71
9	48	38	42	29	44	33	47	50	57
-2	48	50	46	46	40	55	36	52	**54**

Визначення відстані між двома послідовностями

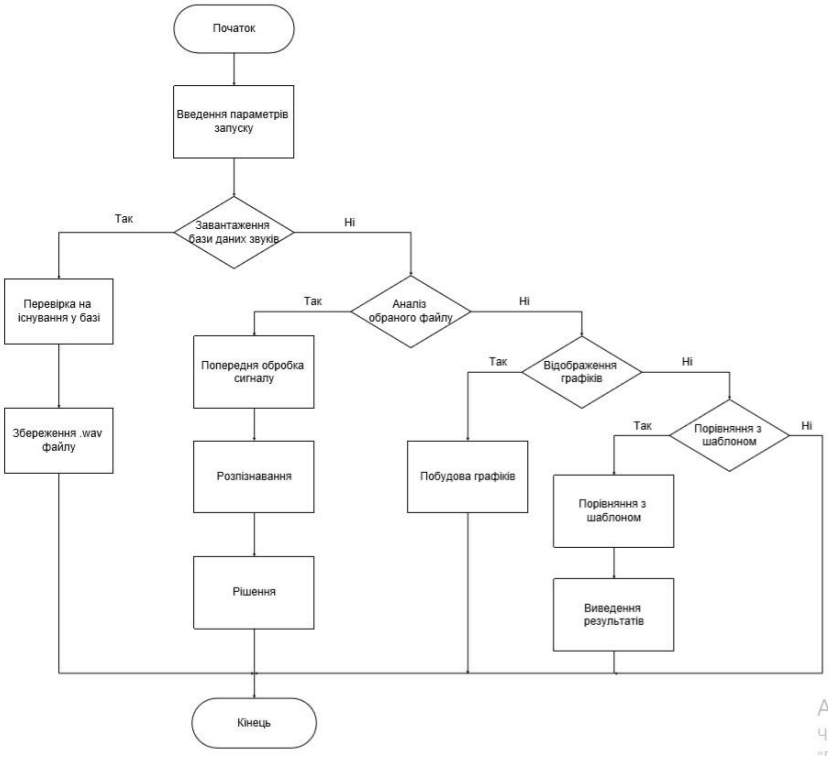
Команда	Відстань	Кут	Шум	Коефіцієнт розпізнавання
«Блокнот»	5м	90°	100%	49%
«Ворд»	5м	90°	100%	32%
«Ком’ютер»	5м	90°	100%	45%
«Пейнт»	5м	90°	100%	43%



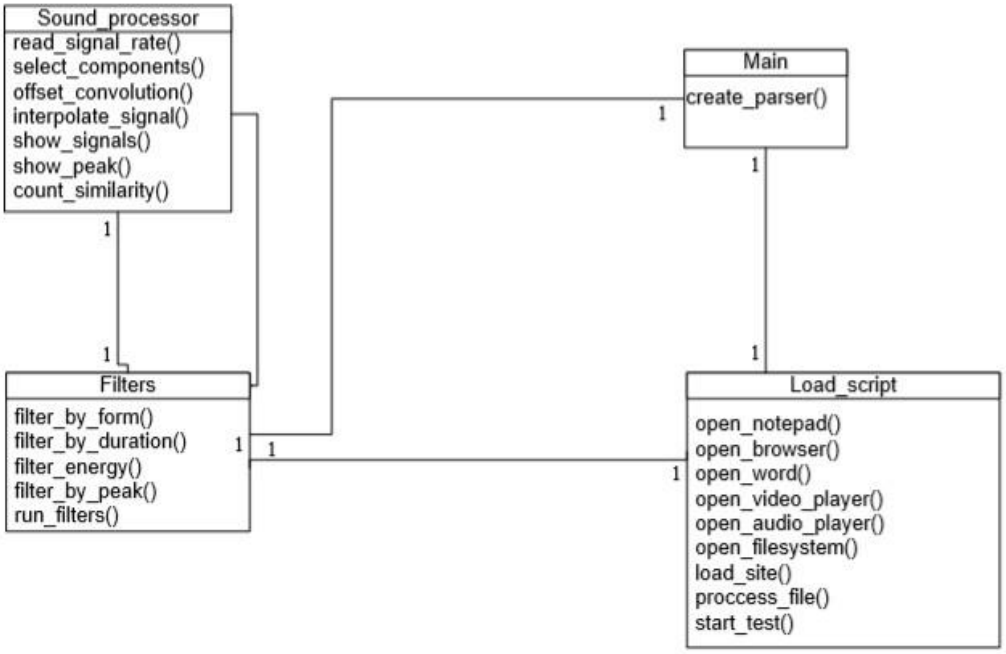
Результати розпізнавання



Загальна схема роботи з .wav файлами



Деталізована схема роботи програми



Взаємодія класів

Загальна схема роботи програми з голосовою командою