Національний лісотехнічний університет України
(повне найменування вищого навчального закладу)

Навчально-науковий інститут деревообробних та
комп'ютерних технологій і дизайну
(повне найменування інституту, назва факультету (відділення))

Кафедра інформаційних технологій
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до дипломної роботи

другий (магістерський)
(рівень вищої освіти)

на тему: "Порівняння та аналіз сервісів розпізнавання природної мови"

Виконав: студент _6__ курсу групи _КНм-61_
спеціальності
122 "Комп'ютерні науки"
(шифр і назва напряму підготовки, спеціальності)

Ференц Роман Андрійович
(прізвище та ініціали)

Керівник к.т.н, доц. Крошний І.М.
(прізвище та ініціали)

Рецензент
(прізвище та ініціали)

Львів – 2021

Національний лісотехнічний університет України
(повне найменування вищого навчального закладу)

ННІ Деревообробних та комп'ютерних технологій і дизайну
Кафедра інформаційних технологій
Рівень вищої освіти другий (магістерський)
Спеціальність 122 – "Комп'ютерні науки"
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Крошний І.М.
"____" _____2021 року

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Ференц Роман Андрійович
(прізвище, ім'я, по батькові)

1.Тема роботи " Порівняння та аналіз сервісів розпізнавання природної мови "
керівник роботи к.т.н., доц. Крошний Ігор Миколайович,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від "31" грудня 2020 р. №С-593

2. Термін подання студентом роботи 10.12.2021 р.

3. Вихідні дані до роботи Формулювання задачі та її формалізація. Аналіз попередніх досліджень. Огляд програмних засобів для реалізації поставленого завдання.

4.Зміст пояснювальної записки (перелік питань, які потрібно розробити)

1) Стан проблемної області.
2) Інформаційне забезпечення
3) Метематичне забезпечення

4) Програмне забезпечення
5) Стартап проекту
4) Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
слайди доповіді, актуальність теми, постановка завдання, аналіз отриманих результатів.

6.Дата видачі завдання 18 грудня 2020

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд літературних та інших джерел згідно досліджуваної теми. Збір потрібних матеріалів.	18.12.20-05.04.21	
2	Постановка задачі та її формалізація.	05.04.21-12.05.21	
3	Вибір та обгрунтування методів і засобів розв'язання завдання.	12.05.21-02.06.21	
4	Програмна реалізація системи.	02.06.21-16.08.21	
5	Оформлення опису створеної програми.	16.08.21-20.09.21	
6	Аналіз отриманих результатів виконання програми.	20.09.21-20.10.21	
7	Здача пояснювальної записки на перевірку та виправлення виявлених помилок.	20.10.21-10.12.21	

Студент _____ Ференц Р.А.
(підпис) (прізвище та ініціали)

Керівник роботи _____ к.т.н., доц. Крошний І.М.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Дипломна робота містить 52 сторінки пояснювальної записки, 17 рисунків, 6 таблиць, 2 додатоки, 17 джерел.

Було досліджено і вивчено основні поняття теорії розпізнавання мови.

Оглянуто і аналізовано літературні джерела за відповідною темою і зроблено відповідні висновки про доцільність використання описаних алгоритмів та інструментів.

Розглянуто спектр задач, які ставить перед собою дана тема, а також шляхи до їх вирішення.

Оглянуто основні алгоритми розпізнавання мови, а також інструменти, які використовують ці алгоритми.

Також було детально досліджено роботу обробки звукових сигналів і методи маніпулювання ними.

Досліджено вплив якості аудіо потоку та особливості мовлення, що впливають на процес розпізнавання мови та засоби покращення даних показників.

Детально порівняно різні типи готових інструментів, які базуються на розглянутих алгоритмах розпізнавання мови.

Як результат вибрано один інструмент розпізнавання мови – Microsoft Speech Recognition API і обґрунтовано доцільність його використання на реальному прикладі.

Ключові слова: Microsoft Speech Recognition API, GoogleSpeechAPI, YandexSpeechKit, Julius.

ABSTRACT

Thesis contains 52 pages of explanatory note, 17 figures, 6 tables, 2 appendices, 17 sources.

The basic concepts of the theory of speech recognition were investigated and studied.

Literature sources on the relevant topic were examined and analyzed, so appropriate conclusions were made about the feasibility of using the described algorithms and tools.

The range of tasks that this topic poses, as well as ways to solve them are considered and the basic speech recognition algorithms and tools that use these algorithms are examined.

Also, the work of processing of sound signals and methods of their manipulation was studied in detail.

The influence of the quality of audio stream and features of speech affecting the process of speech recognition were investigated. Also, the means of improving these indicators are determined.

Different types of ready-made tools based on the considered speech recognition algorithms were compared in details.

As a result, Microsoft Speech Recognition API is selected and the expediency of its use on a real example is justified.

Keywords: Microsoft Speech Recognition API, GoogleSpeechAPI, YandexSpeechKit, Julius.

ТЕХНІЧНЕ ЗАВДАННЯ

1. Оглянути і аналізувати літературні джерела за темою «розпізнавання мови»;
2. Дослідити особливості аудіопотоку та особливості мовлення, що впливають на процес розпізнавання мови;
3. Провести попередній аналіз та обробку даних;
4. Дослідити інструменти та засоби розпізнавання мови для побудови та інтеграції їх у прикладне програмне забезпечення;
5. Реалізація програми голосового керування показом слайдів, на основі MicrosoftSpeechRecognitionAPI.

ЗМІСТ

ВСТУП

На сьогоднішній день, інформаційні технології відіграють все більш важливу роль для існування та розвитку людства. Як результат, левова частина процесів автоматизується, тим самим мінімізує необхідність людині приймати безпосередню участь у їх здійсненні. Наприклад, з'являється все більше нових додатків, які надають можливість відтворювати аудіо матеріал у вигляді тексту. Однак, у цьому випадку необхідно розуміти те, що якість визначення окремих слів у подібних програмах ще далека від ідеальної. Звісно, подібні програми намагаються покращувати, додавати новий функціонал тощо.

Кажучи глобально, взаємозв'язок між пристроєм та користувачем у вигляді обміну інформацією є досить важливим. Покращення алгоритмів визначення мови та формування повноцінного тексту надало б більше переваг та можливостей для подальшої автоматизації процесів. Наприклад, зазначені дії надали б змогу людині, що немає необхідних знань у програмуванні, повноцінно обмінюватися інформацією з машиною. Якщо розглядати це питання ще більш вузько, то у подальшому людство може дійти до такої стадії розвитку, коли користувач скаже персональному комп'ютеру команду, а він її виконає.

Актуальність теми, відштовхуючись від інформації, що наведена вище, полягає в тому що, вона відіграє важливу роль у подальшому розвитку не лише людства, в цілом, а й України. Покращуючи алгоритми розпізнавання людської мови, можливо більш продуктивно виконувати певні завдання, серед яких:

- взаємодію між людиною та персональним комп'ютером;
- Надсилання машині інформації щодо нових варіантів доступу;
- Мовне підтвердження проведення фінансової транзакції;
- Взаємодія з пристроєм, завдяки які аудіо інформація, що надходить від користувача, замінюється на текст;
- Повністю автоматизоване заповнення примитивних анкет, які не потребують прямої взаємодії з користувачем;

- Використання результатів розвитку IoT (технології Інтернет-речей);
- Використання необхідних програм людьми з обмеженими можливостями для виконання повсякденних дій, наприклад, використання функціоналом «розумного будинку».

При цьому, необхідно зазначити, що реалізації подібного програмного забезпечення не є досить легким завданням. Вона потребує від розробника досить великих знань та вмінь. Як результат, станом на 2021 рік, досить часто з'являються нові алгоритми розпізнавання. Наприклад, науковці по усьому світу формують нейронні мережі, які б мали змогу досліджувати інтонацію людини, перетворюючи мову на повноцінний текст. Також, необхідно зазначити використання Марковських моделей та експертних систем. Окремо потрібно виділяти фонемно-орієнтований метод.

Формування таких алгоритмів та їх подальше використання принесло певні результати. Наприклад, почали з'являтися системи, які можуть використовувати різні варіанти розпізнавання мови, що надходить від користувача. Також, необхідно зазначити, що сьогодні існує досить багато систем, головним завданням яких є розпізнавання мови. Найбільш відомим та популярним, скоріш за все, є програмне забезпечення для операційної системи iOS. Apple створили електронного помічника, більш відомого, як Siri, яка повинна не тільки перетворювати мову людини на текст, а надавати власнику пристрою релевантні відповіді на її питання. Також, яскравим прикладом є розробка міжнародної корпорації IBM, яка отримала назву Watson. Необхідно зазначити, що левова частина систем розпізнавання сьогодні є платними. Більш того, якість розпізнавання та перетворення мови є досить невеликою, бо вона не перевищує 70 % успішних розпізнавань.

При цьому, необхідно зазначити, що також є велика кількість безкоштовних програмних застосунків. Найбільш відомим та яскравим є GoogleSpeechAPI.

Об'єктом дослідження є процес розпізнавання голосових команд.

Предметом дослідження є порівняння найкращих сервісів розпізнавання природної мови та застосування їх для керування слайдшоу голосовими командами.

Практичне значення полягає тому, що було побудовано діаграми порівняння сучасних фреймворків для розпізнавання мови, здійснено кількісне та якісне порівняння даних бібліотек та детально проаналізовано алгоритм роботи розробленого програмного додатку для керування слайдшоу за допомогою голосу.

Наукова новизна. В процесі аналізу та розроблення додатку, використано сучасні підходи у створенні подібних застосунків. Знайдену перспективну область застосування даної системи.

Мета дипломної роботи:

1. Провести теоретичний аналіз інформації, пов’язаної з алгоритмами розпізнавання мови людини.
2. Визначати найбільш важливі чинники та ознаки, які безпосередньо впливають на розпізнавання пристроєм мови людини.
3. Дослідити різного роду інструменти, які можна використовувати для створення програмних застосунків, які нададуть можливість реалізації питань розпізнавання.

Завдання, які були поставлені до цієї дипломної роботи, розкривають усі вищезгадані поняття:

1. Оглянути і аналізувати літературні джерела за темою «розпізнавання мови».
2. Дослідити особливості аудіопотоку та особливості мовлення, що впливають на процес розпізнавання мови.
3. Дослідити інструменти та засоби розпізнавання мови для побудови та інтеграції їх у прикладне програмне забезпечення

РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

1.1. Основи теорії розпізнання мови

Для ефективного дослідження алгоритмів розпізнавання мови, необхідно надати загальне визначення цьому поняттю. Кажучи глобально, під дефініцією «розпізнавання мови» необхідно розуміти алгоритм перетворення мови людини, яка передається пристрою, на текст.

При цьому, зазначений термін не потрібно вважати синонімом до терміну «розпізнати мову». Останні лише вказує на можливість пристрою визначати, чи є отримана інформація людською мовою та до якої мови взагалі відноситься.

Також, потрібно розуміти, які саме технології використовуються у додатках, головним завданням яких є визначення людської мови. Як правило, це можливість безпосереднього розпізнавання голосу та мови, можливість формувати текст завдяки мові, визначати машиною інформацію, що надсилається її не власником, а через файли тощо.

Звісно, кажучи глобально, ключовим завдання подібного роду технології є розпізнавання людської мови та надсилання їй інформації текстом. При цьому, цей текст повинен бути релевантним інформації, що була отримана від власника. Більш того, ця інформація повинна залишатися у пам’яті смартфону, персонального комп’ютеру тощо.

Як приклад використання подібних програм можна уявити факт формування текстового варіанту книги, яка на сьогодні існує виключно у вигляді нотатків. Звісно, якщо власноруч друкувати увесь текст на ПК, то на це необхідно буде витратити велику кількість часу. У цьому випадку оптимальним варіантом здається надсилання машині аудіо інформації, яку вона згодом перетворить в текстовий еквівалент. Згідно з цим, алгоритми систем розпізнавання мови є інтуїтивно зрозумілими. Програма отримує звук від власника пристрою, або будь-якої іншої людини, розпізнає мову, визначає структури речення та у результаті надсилання текстовий варіант.

Відштовхуючись від наведеної інформації можна прийти до висновку, головним джерелом інформації для додатків з розпізнавання мови є звук. Якщо надавати більш глобальне визначення поняттю звука, то це є хвилі різної частоти. На рисунку 1 графічно відображена інформація отримання звуку.

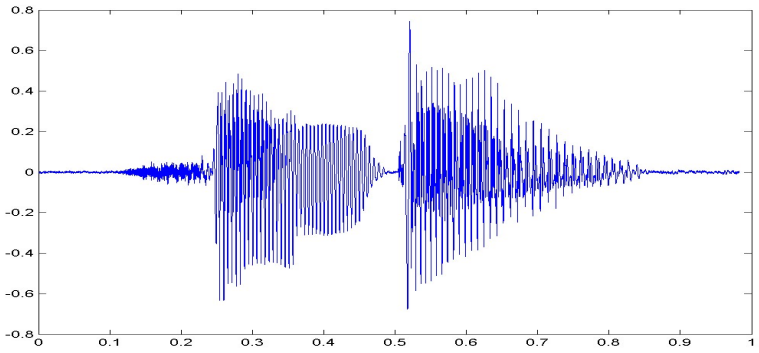

Рис.1 Звукова хвиля.

Щоб пристрій мав змогу визначити джерело звукового сигналу, то для нього необхідно розбити інформацію на певну кількість проміжків. При цьому, кожен з цих проміжків повинен мати певне значення. Ця інформація наведена на рисунку 2.

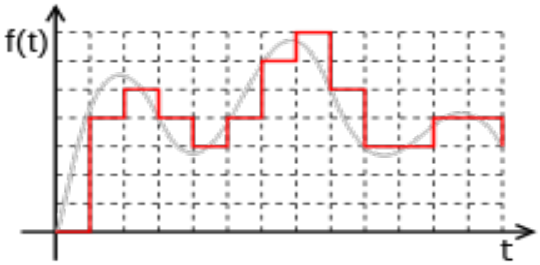

Рис.2 Усереднення сигналу.

Як результат наведеної вище інформації можна прийти до висновку, що персональні комп'ютери з необхідним програмним забепеченням використовують звукові сигнали у вигляді коливань, які згодом перетворюються в набір чисел. Саме вони є основою для формування тексту. Тобто, потрібно розуміти, що пристрій розпізнає мову не у вигляді тексту, а у вигляді чисел, які у подальшому перетворюються на текстовий варіант [1, с. 88-90].

1.2. Постановка задачі розпізнавання мови

Отримана вище інформація надає можливість визначити та сформувати ключову задачу з розпізнавання мови. Вона полягає у формуванні певного алгоритму, який має можливість найбільш ефективно розпізнавати звукові сигнали, що надходять від людини, та формувати з них текст на основі коливань. На сьогоднішній день, найбільш регулярно подібними додатками використовується першочергове порівняння програмою мови людини та її окремих частин з інформацією, що міститься у словниках застосунку. Як правило, ці словники містять декілька найбільш регулярно використовуваних слів. Тобто, у глобальному сенсі, не кожна сучасна програма може ефективно визначити мову людини через те, що певні її слова відсутні у словнику [1, с.12-17].

Програмний застосунок, що використовується для розпізнавання мови, після отримання звукового сигналу формує дерево рішень, завдяки якому можливо найбільш швидко та ефективно визначити мову людини та інформацію, що надходить від неї. Зокрема, використання дерево рішень полягає у визначення фонетичних складових слів, які відносяться до певного рівня дерева рішень, а також обрання найбільш релевантних варіантів, які визначаються за рахунок інтонації, що надходить від людини.

Якщо мова йде про процес розпізнавання мови, то головним завданням, а також об’єктом є безпосереднє перетворення звукової інформації, що була отримана персональним комп’ютером, на текст. Більш того, у своїх алгоритмах такого роду застосунок повинен вміти не лише аналізувати, а й обробляти отриману інформацію. Потрібно розуміти, що у отриманій інформації можуть мати місце фрагменти з шумом, який необхідно видаляти на програмному рівні. Це такоє елемент дослідження, що виконується пристроєм після отримання інформації.

Як результат, отримана персональним комп’ютером інформація повинна бути оброблена у сім етапів:

1. Стартова обробка отриманих звукових сигналів.
2. Видалення зайвої інформації, наприклад, шуму.
3. Сегментація отриманої інформації на рівні.
4. Визначення частоти основного тону внаслідок розподілу сигналу.
5. Визначення ключових параметрів отриманого звукового сигналу.
6. Сегментація отриманої інформації.
7. Виконання кореляційного аналізу.

1.3. Процес видалення шуму з мовних сигналів

Зазвичай, найбільш вагомою проблемою для застосунків, ключовим завданням яких є розпізнавання мови, є визначення шуму у звукових сигналах. Згідно з цим, на цьому етапі необхідно зробити ключові акценти.

Якщо розглядати цей етап більш глобально, то необхідно зазначити, що ключовим його завдання є визначення елементів мовного сигналу, які відносяться до фонового шуму, та їх подальше видалення. Необхідно зазначити, що подібного роду шум може мати декілька різновидів. Зокрема, він буває стаціонарним та нестаціонарним.

Яким чином алгоритми з розпізнавання мови визначають ділянки звукового сигналу, які відносяться до стаціонарного шуму? Як правило, вони використовують перетворення Фур'є. Для початку пристрій отримає інформацію, яка складається виключно з шуму. Згідно з отриманими даними, виконується Гауссівський розподіл. У подальшому, алгоритм програми повинен визначити ділянки мовного сигналу, в яких є частини, що складаються не лише з шуму, а й корисного сигналу. Для таких ділянок також визначаються параметри Гауссівського розподілу. Факт наявності у звуковому сигналі корисного сигналу визначається за допомогою теореми Байеса.

У результаті повинен залишитися виключно корисний сигнал. Для виконання цієї задачі використовується Марківська мережа.

Тобто, якщо казати глобально, то головним завданням обробки мовного сигналу є визначення безпосередньо корисного сигналу, а також подальше видалення наявних шумів.

Потрібно розуміти, що шум у звуковому сигналі может бути різного роду. Згідно з цим з’являється необхідність виконати наступні завдання:

1. Потрібно визначити, які елементи мовного сигналу відносяться до корисного сигналу. На цьому етапі необхідно визначити найбільш важливі параметри для фільтрації отриманої інформації [1,с.78-99];

2. Більш складним та важливим етапо є визначення мовних сигналів з елементів отриманих машиною даних, де є мово-подібний шум. Яскравим прикладом такого шуму є ситуація, у якій одночасно щось кажуть відразу два диктора, а завданням машини є визначення мовних елементів лише одного з них. Реально ефективних варіантів рішення подібних ситуацій, на сьогодні, не існує.

3. Також ключовим завдання є відновлення сигналів, які мали деякі спотворення у процесі надсилання. Цей етап є актуальним після появи у глбальній мережі так званих цифрових телефонних ліній. Глобальною проблемою є спотворення початкового сигналу.

Якщо розглядати особливості [1, с. 144-158], то у цій науковій роботі використовується фільтрація за допомогою не рекурсивних фільтрів. Потрібно розуміти, що подібного роду задача складається з двох класів задач. Їх використання залежить від певних складових. Наприклад, від того, чи має місце фізична модель нестаціонарного шуму.

Висновки до розділу

В даному розділі було розглянуто стан, проблеми та історію виникнення систем для взаємодії людини та комп’ютерних систем за допомогою голосових команд. Було проаналізовано стан області на даний час, та визначено що системи для голосової взаємодії набирають популярність та постійно вдосконалюються

РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1. Історія розвитку алгоритмів розпізнавання мови та їх сьогодення

Перші системи розпізнавання мови могли розуміти лише цифри(враховуючи складність будь-якої мови – це правильно, що інженери спочатку зосередились на цифрах).BellLaboratories в 1952 роціrозробили систему «Audrey», яка успішно розпізнавала цифри, промовлені одним і тим же голосом(однією і тією ж людиною). Через 10 років, в 1962 році, IBMпродемонструвала свій виріб – систему «Shoebox», яка розуміла 16 слів англійською мовою.

Лабораторії в США, Японії, Англії і тодішнього СРСР розробили ще декілька апаратів, які розпізнавали окремі вимовлені звуки, розширивши при цьому саму технологію розпізнавання мови підтримкою чотирьох голосних і дев'яти приголосних звуків будь-якої тональності і тембру. Звичайно працювали ці інструменти не завжди добре, але ці перші спроби дали вражаючий старт даній галузі інформаційних технологій, особливо, якщо врахувати той факт, що комп'ютери на той час були дуже примітивними[3,с.5-9].

Ці перші системи працювали за одним із найпростіших і перших алгоритмів розпізнавання мови – «взяв – порівняв». Його суть дуже проста. Спочатку людина, мову якої розпізнавали, говорила слова, які потрібно було розпізнати. Ці слова брали і записували на будь-який доступний накопичувач у вигляді аудіопотоку, який машина потім перетворювали в байти і біти. Таким чином машина вже знала, які конкретно слова вона мусить розпізнати(тональність і особливості голосу теж враховувались). І після таких маніпуляцій комп'ютер в більшості випадків успішно розпізнавав ці слова, сказані цією ж людиною. Схематично цей алгоритм можна зобразити так(див. рис.4):

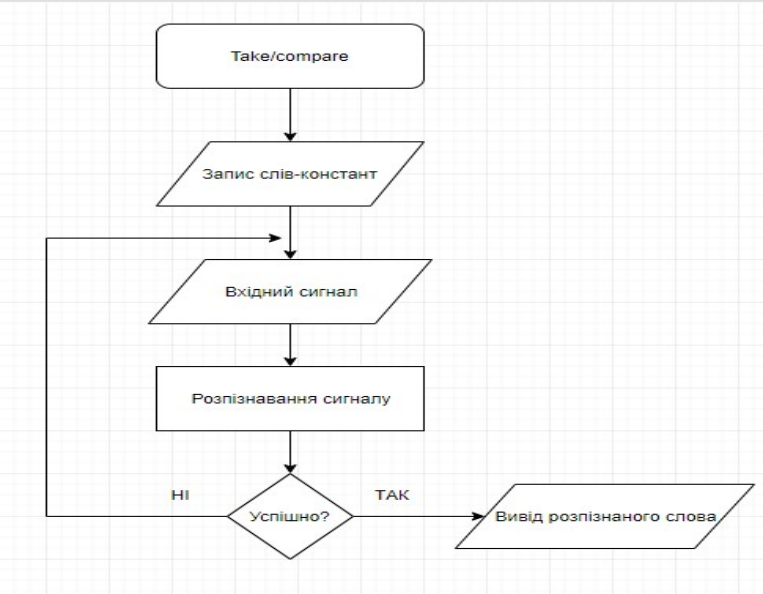

Рис.3 Блок-схема алгоритму "Взяв-порівняв"

У 1970 – 1990х роках був своєрідній «застій» розвитку систем автоматичного розпізнавання мови. Це протривало до 2000х років, покиGooglene анонсував одну з перших систем голосового пошуку в 2010 році для Android. З того часу тема автоматичного розпізнавання мови отримала потужний поштовх, який в кінцевому результаті вилився в те, що ми маємо сьогодні.

На сьогоднішній день існує дуже багато голосових помічників Siri, Cortana, Lizai т.д. Кожен з яких працює на власній системі розпізнавання мови. Також створено безліч голових інструментів, які будь-який бажаючий(за оплату або безкоштовно) може інтегрувати в своє програмне забезпечення. І цей потік нових технологій лише збільшується.

Отже можна з впевненістю сказати, що така галузь інформаційних технологій як розпізнавання мови стала однією з провідних в теперішньому світі. І тому має великі перспективи для розвитку.

2.2 Алгоритми розпізнавання мови та їх класифікація.

Будь-який алгоритм розпізнавання мови має приблизно такий хід дій(див. рис.5):

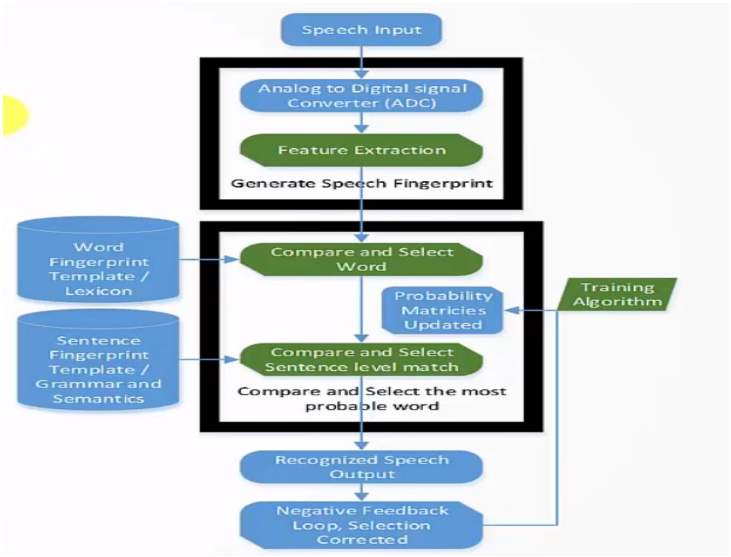

Рис.4 Загальний вигляд алгоритмів

На першому кроці ми маємо вхідні дані – голосовий потік (аудіопотік).

На другому кроці цей аудіопотік перетворюється на зрозумілі машині нулі та одиничкі і, крім того, більшість сучасних алгоритмів на цьому етапі максимально відфільтровують звук від різного роду дефектів(шуми, сторонні звуки і т.д.). Тобто тут формується основа для розпізнавання(чисті вхідні дані).

На третьому кроці використовуючи різні запрограмовані шаблони словосполучень та словники, а також спираючись на особливості вибраного лексикону система намагається спів ставити вхідні дані з вже існуючими словами(які вона вміє розпізнавати) і, якщо схожість є більшою за 81%(зазвичай використають саме цю константну величину), повідомляє результати розпізнавання. За такою ідеєю працюють майже усі існуючі на даний момент системи розпізнавання мови[2, 23-44с].

Відповідно до згаданого стандарту, систем розпізнавання мови(СРМ) розрізняють за наступними ознаками:

- інтервал між окремими словами;
- залежність від диктора;
- ступінь деталізації при завданні еталонів;
- розмір словника.

Системи розпізнавання, яким властива відносна незалежність від диктора, дозволяють користувачу працювати без попередньої настройки, однак поліпшують надійність розпізнавання після навчання. Незалежність від диктора таких систем звичайно досягається за рахунок збереження звукових еталонів для усіх найбільш типових голосів носіїв даної мови. Це, безумовно, вимагає в кілька разів більшої продуктивності й обсягу пам'яті. Настроювання на голос диктора дикторозалежних систем займає звичайно від 30 хвилин до декількох годин. Це складає головну незручність для користувача. Звичайно дикторозалежні системи дозволяють працювати з відносним ступенем надійності без попереднього настроювання на голос конкретного користувача. Третім різновидом систем за цією ознакою є системи, що автоматично настроюються на голос диктора в міру їх використання. Системи останнього типу мають дві основні особливості. По-перше, їм потрібно знати, чи зробив користувач помилку, вимовивши конкретне слово (інакше навчання буде невірним). А по-друге, після настроювання на одного диктора такі системи перестають надійно працювати з іншими дикторами.

На сьогодній день в світі найбільшого визнання та поширення здобули такі програмні продукти з розпізнавання мови,такі як Dragon Naturally Speaking, IBM Voice, OfficeTALK, KVWin, Micro-Introvoice, Speech Magic та інші.Дані програмні продукти використовуються переважно в медицинських закладах, юриспруденції та роботи в офісах.

Всі ці системи цікаві тим, що в собі містять певний алгоритм розпізнавання мови, зазвичай не один, а два і навіть більше. Тому слід перейдемо до детальнішого опису найвідоміших із них.

Висновки до розділу

В даному розділі було проведено аналіз основного алгоритму розпізавання мови, проблеми та історію виникнення систем розпізнавання голосових команд, сучасних засобів для розпізнавання мовлення та класифікацію таких систем.

РОЗДІЛ 3.МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1. Математична модель процесу розпізнавання мови

Для побудови математичної моделі з розпізнавання мови необхідно розуміти, що мовни сигнал перетворюється у програмі на певну послідовність акустичних векторів ознак. Їх можна визначити, як $Y_{1,T} = (y_1, y_2, ..., y_T)$. Вони формуються у результатів обробки. Далі, алгоритми програмного застосунку повинні перетворити отриманий сигнал у код. Для цього він виконує пошук мовних сегментів, які можна визначити, як $w_{1,L} = (w_1, w_2, ..., w_L)$. Вони повинні відповідати Y та визначатися за формулою:

$$\hat{w} = \arg\max_{w} P(w \mid Y) \cong \arg\max_{w} p(Y \mid w) P(w).$$

Права частина вираза в зазначеній вище формулі застосовує правило Байєса. Кажучи глобально, вона формує модель розпізнавання системою застосунку мовлення. При цьому, акустична- $p(Y \mid w)$ та лінгвістична – $P(w)$- елементи цієї моделі описуються власними стохастичними гараматиками.

Розглянемо більш глобально акустичну модель. Вона формується зі слів w та використовує базові мовні елементи, до складу яких потрібно відносити різного роду фонеми. Зазначені фонеми повинні, у свою чергу, формувати фонемну транскрипцію слова, яка визначається за наступною формулою $q_{1,K}^{(w)} = (q_1, q_2, ..., q_{K_w})$. Більш того, враховуючи ймовірність виявлення алгоритмами спонтанного мовлення, у алгоритм додатково додаються неінформативні звуки, що можуть з’являтися у подібному випадку.

Найбільш популярні та ефективні системи розпізнавання мови використовують саме алфавіт фонем. Зазначені фонеми можуть бути як контекстно-залежними, так й незалежними. Саме завдяки їх використанню формуються мовленні образи слів.

На рисунку 3 відображена структура розпізнавання злитого мовлення. Вона використовуються у різного роду технологій розпізнавання мови, наприклад, в клієнт-серверній та ізольованій технології.

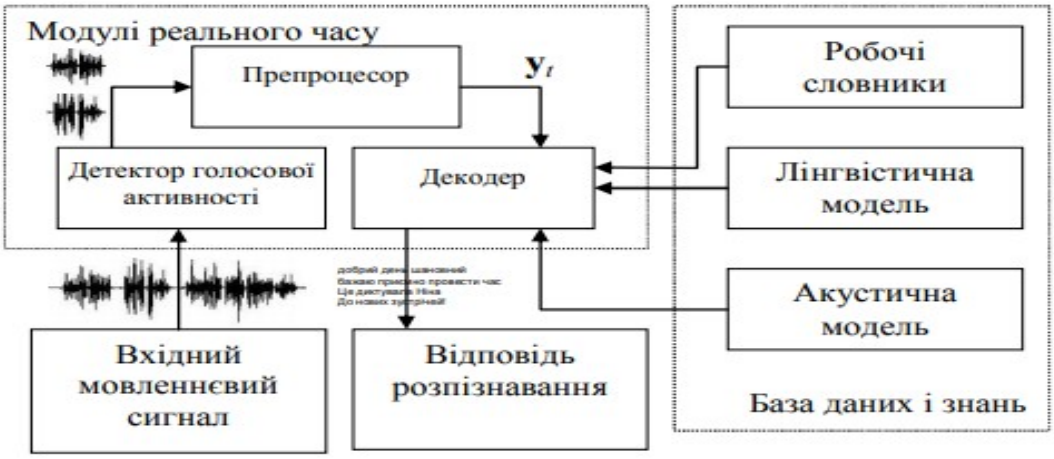

Рис.5 Структура автоматичного розпізнавача злитого мовлення

Модулі, які використовуються у застосунках з розпізнавання мови, використовують для отримання звукової інформації певні механізми. Наприклад, мікрофон за допомоги якого людина надсилає звуковий сигнал системи. Також, це може бути файлова система, за допомогою якої застосунком отримується інформація, яка повинна бути перероблена на текст.

Необхідно зазначити, що портативні пристрої отримуються усю необхідну інформацію, яка використовуються в ізольованих системах. При цьому, клієнт-серверні системи розпізнавання мови використовують елемент розпізнавання звукових сигналів на пристрої [4, 2-4с.].

3.2. Приховані Марковські моделі

Для дослідження використання прихованих Марковських моделей у процесі розпізнавання мови, необхідно надати зазначеній дефініції визначення.

Якщо розглядати моделі Маркова в площині теорії ймовірності, то вони є стохастичними моделями, які використовуються для формування системи, яка може змінюватися у певний проміжок часу. У цьому випадку, майбутній стан

системи безпосередньо пов’язаний з тим, який стан вона має у певному проміжку часу. Тобто, стан системи взагалі не залежить від дій, які виконувалися до їх появи. Ця особливість отримала назву властивість Маркова. Саме завдяки цим припущенням можливо виконувати різного роду обчислення моделі. Без їх використання ці дії були б неможливими.

Необхідно зазначити, що на сьогодні існує чотири моделі Маркова. Вони досліджуються та використовуються у випадках, коли має місце певний стан системи. Також важливим фактором для використання моделей Маркова є те, чи будуть внесені зміни до системи, на підставі отриманих спостережень. Ці чотири моделі Маркова отримали наступні назви:

- Ланцюг Маркова;
- Марковський процес отримання рішень;
- Прихована модель Маркова;
- Марковський процес ухвалення рішень.

Найбільш інтуїтивно зрозумілим є ланцюг Маркова. У процесі його використання визначається певний стан системи, до складу якого входить випадкова змінна, що у подальшому трансформується у часі. Марков зназначає, що розподіл обраної зміни безпосередньо пов’язаний з минулим станом системи. У якості прикладу можна використати ланцюг Маркова Монте-Карло. У ланцюзі використоується властивість Маркова. Вона необхідна для того, щоб виконати розподіл системи на певні частини [3, с.12-19].

Якщо надавати визначення поняттю «модель Маркова», то під ним потрібно розуміти ланцюг Маркова, у випадку з яким стан спостерігається лише частково. На сьогодні, існує декілька найбільш популярних алгоритмів, які можна використовувати разом з прихованими моделями Маркова. Одним з таких алгоритмів є алгоритм Вітербо. З його допомогою можна визначити найбільш релевантну послідовність станів. Крім цього, алгоритм Баума-Уелча проводить оцінку стартових ймовірностей.

Одним з ключових напрямків використання цих алгоритмів є розпізнавання мови. При цьому, спостережувальними даними повинні бути звукові сигнали, що надходять від користувача. Приховані стани, у цьому випадку, є проголошенням тексту.

Кажучи глобально, під «прихованою Марковською моделлю» потрібно розуміти граф, де з кожною його вершиною пов'язана наступна функція:

$$M \overset{def}{=} < G, \{b_j\} > .$$

Марковська модель у загальному випадку відображена на рисунку 6.

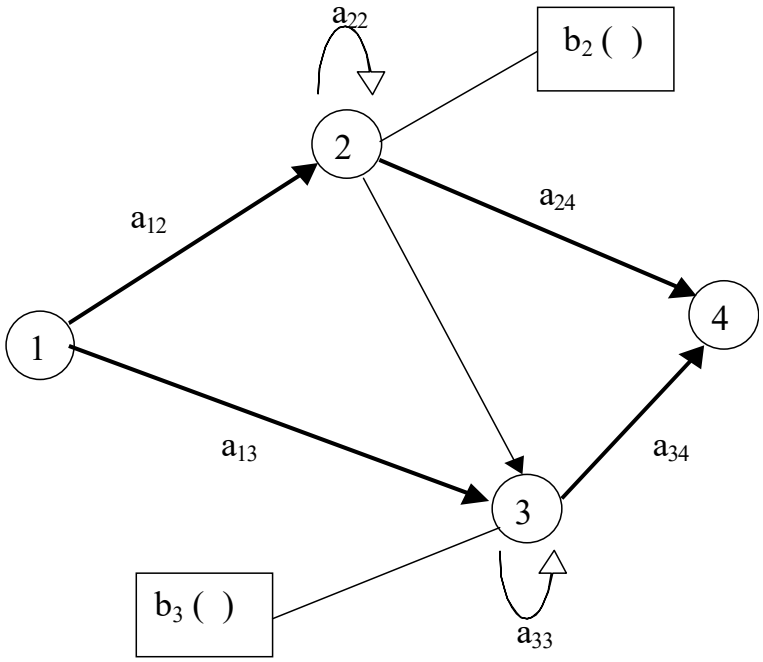

Рис.6 Приклад ПММ

Потрібно зазначити, що для вирішення завдання розпізнавання мови, вершини ПММ отримують навантаження, пов'язаного з нормальним законом розподілу.

Першим етапом розпізнавання мови є визначення частин мови, до яких відноситься кожне зі слів, що зустрічаються у тексті.

Якщо аналізувати це питання з точки зору англійської мови, то можна одразу визначити певну проблему. Наприклад, слово «can» у левовій частині речень відіграє роль дієслова. При цьому, у певних випадках воно може бути також й іменником. Саме для рішення подібних проблем була створена нова

модель, де додатково аналізується яка частина мови буде наведена після артикля, прикметник чи іменник. Ця формула виглядає наступним чином:

$$\arg\max_{t_1 \ldots t_n} \prod_{t=1}^{n} p(w_i \mid t_i) p(t_i \mid t_{i-1}),$$

де :

t - мітка(іменник, прикметник і т.д.);

w- слово в тексті(rust, can …);

$p(w \mid t)$ - ймовірність того, що слово w відповідає мітці t ;

$p(t_1 \mid t_2)$ - ймовірність того, що t_1 слідує після t_2.

З зазначеної вище формули можна прийти до висновку, що кожна з наступних міток повинна відповідати попередній. Саме завдяки її використанню, якщо повертатися до слова «can» можна визначити, яку частину мови він відображає – іменника або модального дієслова.

Потрібно зазначити що такого роду модель может бути описана як ергодична ПММ. Вона відображена на рисунку 7.

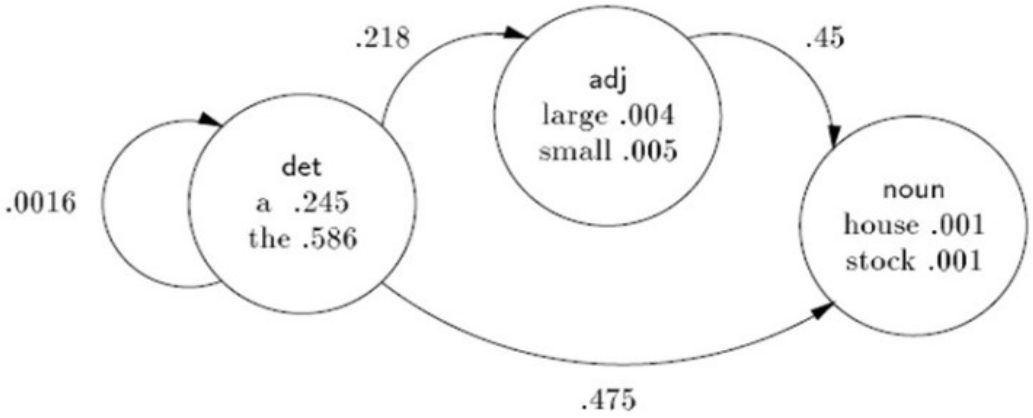

Рис.7 Ергодична ПММ.

Вершини відображені на рисунку 7 відповідають за певну частину мови, навколо яких описуються слова та словосполучення. При цьому, ребра між вершинами відображають ймовірність використання однієї частини мови за іншою. Як приклад, ймовірність двох артиклів, що будуть йти у реченні один за одним дорівнює лише 0,0016. Потрібно зазначити, що цей етап є максимально

важливим, так як він мінімізує ймовірність помилок у відображенні тексту після отриманого аудіозапису.

Крім того, на сьогодні, існують n-грамні моделі. Згідно з особливостями цієї моделі можна прийти до висновку, що ймовірність появи певного слова в реченні залежить від певної кількості слів, яка розраховується по формулі n-1. На сьогодні, найбільш популярними є двох та трьох гранні моделі мови. З іншої сторони, цей алгоритм не дає можливість визначити синтаксичні та семантичні зв'язки, у випадку, коли залежні одні від інших слова знаходяться на відстані 5 слів. При цьому, використання моделей, де n буде більше п'яти, потребує досить великі фінансові та технічні витрати [2, с.75-99].

3.3. DWT алгоритм

DWT або Discrete wavelet transform, на сьогодні, вважається найбільш популярним та продуктивним алгоритмом, головним завданням якого є розпізнавання мови та перетворення її на текст.

Потрібно навести особливості роботи зазначеного алгоритму. Звук повинен проходити через певне середовище з певною швидкістю. Вона безпосередньо залежить від щільності середовища, в якому знаходиться. Необхідно розуміти, що найбільш інтуїтивно зрозумілим способом отримання системою звуків є синусоїдальний графік.

Форма отриманої на синусоїдальному графіку хвилі безпосередньо залежить від трьох складових – фази, амплітуди та частоти.

Теорема Фур'є надає можливість перетворити звукові хвилі на синусоїдальні криві. Потрібно зазначити, що особливість цієї теореми полягає у тому, що кожна періодична хвиля можу бути у подальшому розібрана завдяки синусоїдальній кривій з різними амплітудами, фазами та частотами. Потрібно відзначити, що ця процедура називається аналізом Фур'є. У результаті зазначеного аналізу буде отримані певні амплітуди, частоти та фази для

кожного компоненту хвилі. В результаті, після складання цих складових в одне ціле, буде отримана необхідна звукова хвиля.

У цьому випадку спектром буде названа точка частоти, яка була узята з амплітудою. При цьому, кожен з періодичних сигналів, який відображає рекурсивну модель часу, має назву головної частоти. Потрібно також зазначити, що основна частота також може бути отримана з мовного сигналу. Це можливо через розрахунку періоду коливань біля нулевої осі. Якщо необхідно відобразити частоту послідовності звуків протягом певного періоду часу, то оптимальним варіантом для цього буде спектограма. Вона формується за рахунок двох вимірів, а саме частоти й часу. Особливу увагу необхідно приділити кольору точки. Він вказує на амплітуду інтенсивності. Якщо колір темного відтінку, то вона сильна, якщо світлого, то слабка. Потрібно зазначити, що спектограми відіграють важливу роль у розпізнаванні людської мови. Згідно з цим, фахівці у цій галузі можуть визначити певні особливості після аналізу звукової спектограми [9, с.334-356].

На сьогоднішній день, різного роду методи надають можливість без проблем визначати точку початку та кінця певного слова в контексті глобального звукового потоку.

Необхідно відзначити, що формування початкової та кінцевої точки можна визначити без проблем лише у тому випадку, коли запис звуку виконується в оптимальних умовах. Звісно, що в реальних умовах виконати це досить складно. Це безпосередньо пов’язано з тим, що різного роду фоновий шум може значно ускладнювати процес відокремлення певних слів.

Визначення слів згідно з зазначеним алгоритмом виконується шляхом порівняння певних сигналів.

Існують 2 особливості застосування алгоритму.

Перша особливість в тому, що пряме порівняння числових форм сигналів. У цьому випадку, для кожної числової послідовності створюється нова послідовність, розміри якої значно менші. Алгоритм має справу з цими

послідовностями. Числова послідовність може мати кілька тисяч числових значень, в той час як підпослідовність може мати кілька сотень значень. Зменшення кількості числових значень може бути виконано шляхом їх видалення між кутовими точками. Цей процес скорочення довжини числової послідовності не повинен змінювати свого представлення. Безсумнівно, процес призводить до зменшення точності розпізнавання. Однак, беручи до уваги збільшення швидкості, точність, по суті, підвищується за рахунок збільшення слів у словнику.

Друга ж полягає в тому, що потрібно виконуватиподання сигналів спектрограм і застосування алгоритму DTW для порівняння двох спектрограм. Метод полягає в розділенні цифрового сигналу на деяку кількість інтервалів, які будуть перекриватися. Для кожного імпульсу, інтервали дійсних чисел (звукових частот), буде розраховуватись швидким перетворення Фур'є, і буде зберігатися в матриці звукової спектрограми. Параметри будуть однаковими для всіх обчислювальних операцій: довжин імпульсу, довжини перетворення Фур'є, довжини перекриття для двох послідовних імпульсів. Перетворення Фур'є є симетрично пов'язаним з центром, а комплексні числа з однієї сторони пов'язані з числами з іншого боку. У зв'язку з цим, тільки значення з першої частини симетрії можна зберегти, таким чином, спектрограма буде представляти матрицю комплексних чисел, кількість ліній в такій матриці є рівній половині довжини перетворення Фур'є, а кількість стовпців буде визначатися в залежності від довжини звуку. DTW буде застосовуватися на матриці дійсних чисел в результаті сполучення спектрограми значень, така матриця називається матрицею енергії[10, с.216-244].

Висновки до розділу

В розділі III (математичне забезпечення) було описано алгоритм аналізу звукового потоку, проаналізовано в якому напрямку розвиваються та як вдосконалюються системи для розпізнавання мовлення, зокрема Discrete wavelet transform алгоритм та Приховані Марківські моделі.

РОЗДІЛ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

4.1.Загальний огляд готових систем та засобів розпізнавання мови

Станом на 2021 рік створена велика кількість систем, які використовуються людиною у якості голосових помічників. За їх допомогою можна отримати інформацію щодо останніх новин, прогнозу погоди тощо. Найбільш популярними голосовими помічникам є Siri, що створена компанією Apple та Google Assistant від однойменної системи.

Аналізуючи особливості роботи Siri можна прийти до висновку, що вона є повноцінною інформаційною системою. Її функціонал не обмежується виключно перетворенням звукових сигналів на текст. Вона також може самостійно формувати відповідь на питання, що було отримане від людини. З самого початку відповідь від Siri відображається у вигляді тексту. Через декілька секунд система надсилає голосовий поток. Зокрема, для підвищення ефективності надання відповідей користувачу, до алгоритмів роботи відповідних застосунків додаються Марківські моделі, про які зазначено вище.

Sirita Cortana мають дуже подібну програмну архітектуру. Обидві платформи створені на своїх SiriSpeechRecognitionEnginei CortanaCloudAPI,яка розміщена в AzureCloudStorage. Ці дві системи нажаль є закритими до комерційного використання за межами компаній. І їх детальний розгляд є неможливим.

Надзвичайно цікавим і потужним інструментом є когнітивний сервіс від компанії IBM – Watson. Дана система використовує обробку природної мови і машинне навчання для того, щоб забезпечити зручну, природню взаємодію людини з комп’ютером.

Дана система включає в себе такі підсистеми як Conversation, Watson Virtual Agent, Watson Discovery, Natural Language Understanding та багато інших. Тобто цей сервіс є комплексним інструментом за допомогою якого можна вирішувати велику кількість прикладних задач. Крім того його досить

легко інтегрувати в будь-яке програмне забезпечення і адаптувати під ваші потреби.

Під час тестування IBM Watson показував стійкі і постійні результати. Приблизний рівень успішного розпізнавання становить приблизно вражаючих 81%. Здебільшого це зумовлено тим, що система має дуже хороший алгоритм навчання нейронної мережі, яка фільтрує і обходить всі вхідні дефекти. Тобто в кінцевому результати на вхід до системи поступає лише «чистий» аудіо потік [17].

Вказані вище досягнення є прогресивними, але вони мають й обмеження. Наприклад, отримати доступ до подібних інструментів можливо після оплати відповідної суму. Для того, щоб ця робота була більш практичною та доступною кожному користувачу, нижче проведений аналіз безкоштовних варіантів програмного забезпечення.

4.2. Дослідження інтегрованості та показників успішності розпізнавання мови в системах розпізнавання мови

Для дослідження я вибрав чотири готових системи розпізнавання мови. Це Speech Recognition API, Microsoft Speech Recognition API, Yandex Speech Kit і Julius. Одна з цих систем є системо з відкритим вихідним кодом(Julius) а інші три є системами з закритим вихідним кодом. Також дуже важливим є той факт, що Julius є платним, а Google, Microsoft і Yandex пропонують безкоштовний продукт для використання.

Суть досліду полягає в тому, щоб розпізнати короткий вірш про життя англійською мовою, текст якої є наступним:

Simple Sam was a simple man.
He lived each day by a simple plan.
Enjoy your life and live while you can.
Make each day count and take a stand.

В експерименті з визначення якості розпізнавання мовлення цих систем взяли участь 20 осіб, у тому числі 14 чоловіків та 6 жінок, які читали текст в мікрофон. Всі промовці не є носіями англійської мови, але мають хороші навички розмовної англійської (B1+). Середній вік учасників експерименту становив 22 роки. У групі не було професійних спікерів. Для оцінки результатів автоматичного розпізнавання обрано такий показник, як відсоток правильно розпізнаних слів WCR (WordCorrectlyRecognized). Цей коефіцієнт обчислювався як для кожного спікера окремо, так і для кожного продукту загалом.

$$WCR = \frac{H}{T} * 100\%$$

де H–кількість правильно розпізнаних слів, T–загальне число слів для розпізнавання. Отже, ця формула є дуже простою до використання, але водночас і дуже ефективною.

Першою було протестовано систему GoogleSpeechRecognitionAPI. Це продукт Google, який дозволяє здійснювати голосовий пошук за допомогою технології розпізнавання мови. Ця технологія інтегрована в мобільні телефони і комп'ютери, де можна ввести інформацію в машину за допомогою голосу. З 14 червня 2011 року компанія Google оголосила про інтеграцію цього мовного «двигуна» в свій Google Search і з тих пір система працює в стабільному режимі[16].

Для інтеграції цього інструменту в своє програмне забезпечення вам потрібно зробити наступне:

1. Треба зробити POST запит з аудіо данимив форматі FLAC або Speexна адресу, яку вам надішле GoogleAPIManager.
2. Отримати HTTPresponse і «дістати» звідти розпізнаний текст.

Ви можете реалізувати подібний приклад розпізнавання мови за допомогою мови програмування C#. Кількість обмежень запитів в день не було помічено.

В цілому ця система легко інтегрувалася в невеликі додатки WinForms. Результати були наступними(див. рис.8-9):

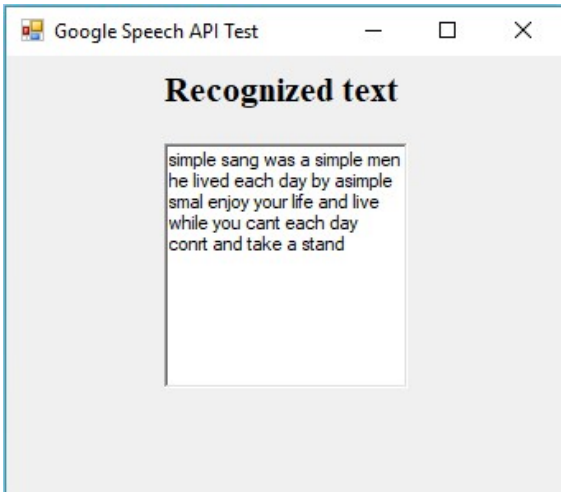

Рис.8 Поганий результат розпізнавання

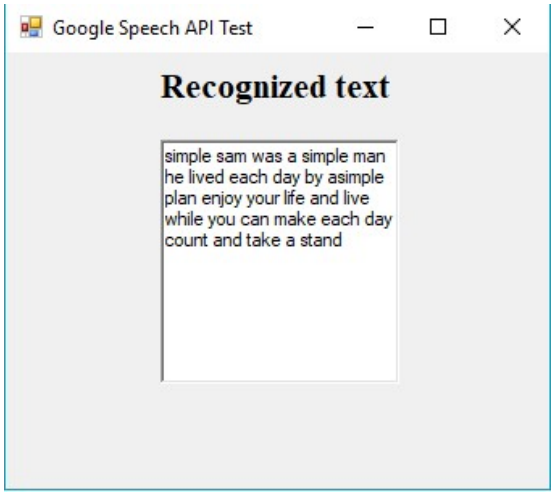

Рис.9 Успішне розпізнавання

Як ви можете бачити, іноді цей інструмент працює дуже добре, а іноді зазнає невдачі. Я показавлише два з 20 результатів, які у мене є, але WCR для цього двигуна дорівнює 81,92 %.

Наступною була система YandexSpeechKit. В цілому ця система працює так само, як і попередня, і якщо ви хочете дізнатися більше, ви можете прочитати документацію продукту на веб-сайті API Яндекса[15]. Цей інструмент дуже легко інтегрується у всі застосунки, які ви схочете з допомогою C#, PHP і багато інших мов програмування.

Також дуже цікавий той факт, що система голосового пошуку Яндекса створена на основі цього API. Таким чином, ми протестували наш вірш за допомогою цього сервісу. Більшість результатів були дуже схожі на наступний (див. рис.10).

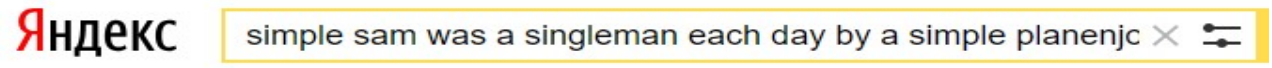

Рис.10 Результат роботи YandexSpeechKit

В загальному ця система отримала показникWCR = 64,81 %.

Третьою системою, яку було протестовано стала MicrosoftSpeechRecognitionAPI. Цей API має свій власний SDK (SoftwareDevelopmentKit), тому цей інструмент дуже легко інтегрувати у ваш додаток. Але він також має деякі обмеження. Перше і найголовніше, це той факт, що нам потрібно налаштувати словник слів для розпізнавання системою вручну. Цей словник називається "Choices". Потім необхідно додати мовні налаштування і створити граматику на основі варіантів і налаштувань.

Для тестування було створено програму, яка цілком аналогічна до тієї, що використовувалась в першому тесті GoogleSpeechAPI. Різниця лише в тому, що не потрібно було посилати POST-запити на сервер, а лише описати словник потрібних слів, які треба розпізнати. Результати, які показала ця система наступні(див. рис.11-12).

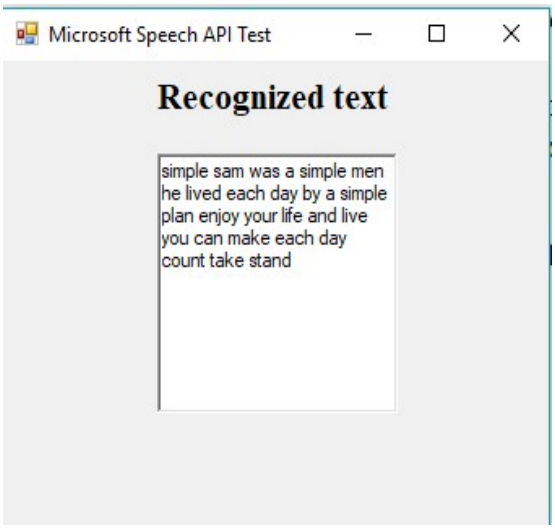

Рис.11Результат роботи MS Speech Recognition API.

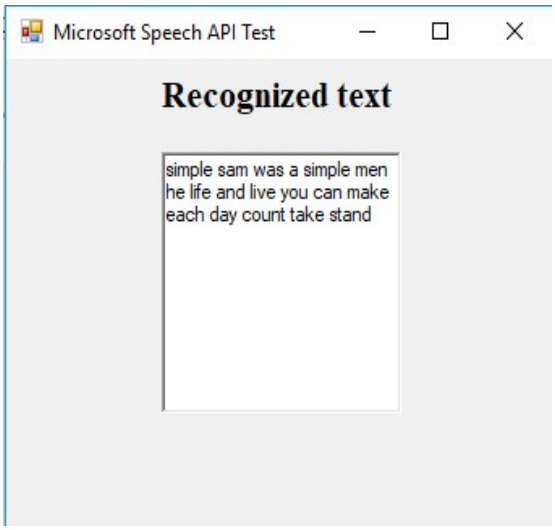

Рис.12Результат роботи MS Speech Recognition API.

Через словник «Choices», якщо система не розпізнає слова, зазначені у словнику, то вона не пропонує ніяких інших слів. Вона просто ігнорує цей вхідний сигнал, як ви можете бачити на малюнку 11 і 12. Але в інших випадках

WCR процесу досить великий. В цілому цей API отримує значення WCR = 79,03%[11].

Останньою системою, яка пройшла тестування, став Julius.

Julius - високопродуктивний розпізнавач безперервної мови з великим словниковим запасом. Це програмне забезпечення декодера для досліджень у галузі розпізнавання мови та розвитку цього напрямку. Система ідеально підходить для розпізнавання майже в реальному часі на більшості існуючих комп'ютерах. ЇЇ словник містить близько 60 тисяч слів і сама система працює з використанням задачі "слово тіаграми" і контекстно-незалежної прихованої Марківської моделі.

Головною особливістю продукту є повна інтегровність. Також тут застосовується безпечна модуляція, яка може бути незалежною від модельних структур і різних типів прихованих Марковських моделей, які підтримують загальний стан трифонів і пов'язаних з ними змішаних моделей з різними зіллями, фонемами і висловлюваннями.

Даний інструмент є повністю інтегровним в будь-яке програмне забезпечення. Система стабільно працює як і на UNIX-подібних ОС, так і на Windows та MACOS. Julius – система з відкритом вихідним кодом і розповсюджується через ліцензію Berkeley Software Distribution(BSD)[14].

Як я вже згадував, ця система платна. Але для тестування ми отримали безкоштовну пробну версію на 7 днів. З допомогою Julius SDK для JavaScript, ми створили простий додаток, який записує вхідний сигнал. Потім програма відправляє його на сервери Julius, де виконується розпізнавання мови. Сама система показала наступні результати(див. рис.13-14).

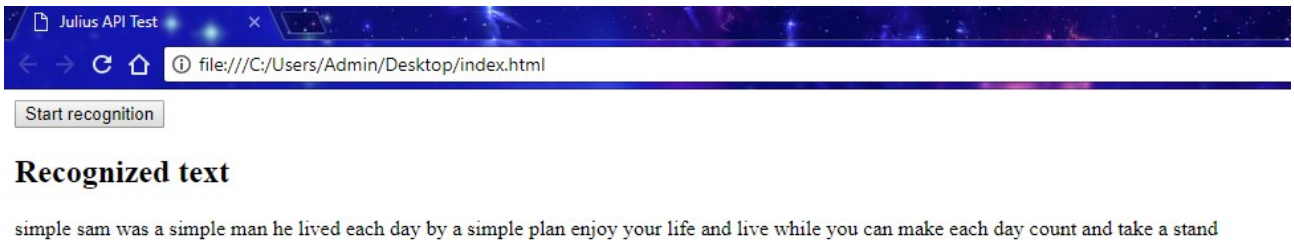

Рис.13JuliusAPI - результати тестування.

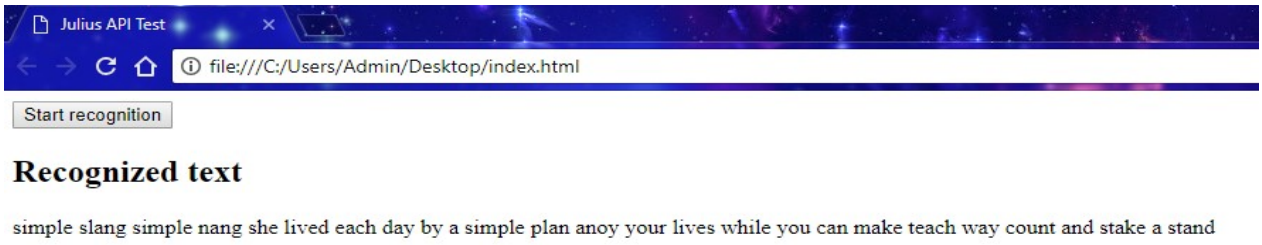

Рис.14 JuliusAPI -результати невдалого розпізнавання.

На рисунку 14 видно, що деякий текст замінений системою Julius, так як він не може правильно розпізнати вхідну мову і інструмент сам намагається вибрати оптимальний варіант для заміни.

Але в цілому ця система має дуже великий індекс WCR-89,67%.

Як результат, я хочу показати вам результати дослідження в наступних таблицях(див. рис.15, табл.1).

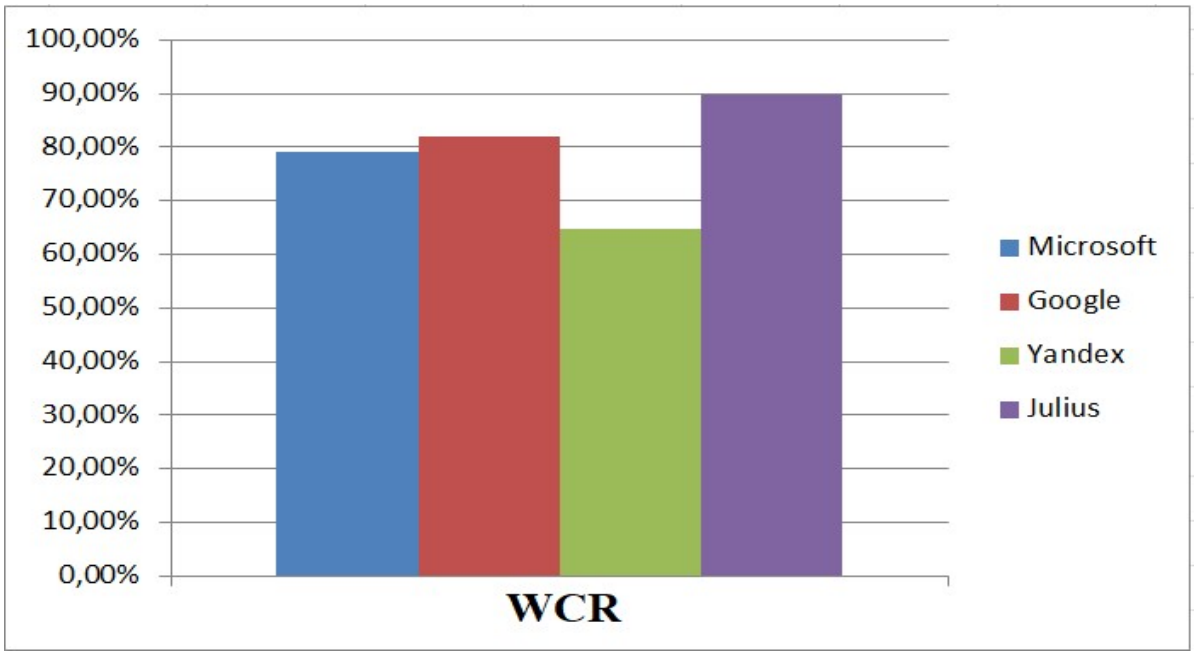

Рис.15 Графік WCR показників.

	Інтегровність (5 балів - max)	Безкоштовність	Open source	WCR значення
Microsoft API	5	+	-	79,03%
Google API	4	+	-	81,92%
Yandex Kit	3	+	-	64,81%
Julius	4	-	+	89,67%

Таблиця 1. Порівняльна таблиця систем розпізнавання мови.

Підводячи підсумок, можна виділити Julius, який показав найкращі результати, і ця система легко інтегрується. Але ця система платна. Серед безкоштовних, можна виділити Microsoft Speech API, а також Google API. Ці два API показують хороший індекс WCR. Продукт Microsoft краще, тому що його легко інтегрувати у ваші програми. Але Google не прив'язаний до певного набору слів і має свій словник. Яку систему обрати - вирішувати тільки вам.

4.3. Microsoft speech recognition API

Як один з найбільш актуальних варіантів, було розглянуте безкоштовне API, створене компанією Microsoft. Воно отримало назву Microsoft Speech Recognition API. Потрібно зазначити, що це API є складовою однойменної платформи, головним завданням якої є створення додатків та застосунків з розпізнавання мови. Як результат функціонування цієї платформи, наприклад, був створений голосовий пошук для Windows 10.

Ефективність цього механізму пов’язана з використанням більше ніж 10 алгоритмів, пов’язаних з розпізнаванням мови людини. Серед іншого, використовуються й приховані Маркові моделі.

Зазначене API також використовують граматику та словники для найбільш швидкого розпізнавання мови. У цьому випадку необхідно надати визначення дефініції граматика згідно з яким вона є лінгвістичною моделлю, яка

використовується під час процесу розпізнавання. Потрібно зазначити, що такою моделлю може бути, зокрема, й українська мова. Більш того, зазначене API дозволяє використовувати й власну граматику. Її використання надасть можливість додати до системи розпізнавання різного роду діалекти. Граматика у цьому сенсі вирішує такі види проблем:

- значно збільшує ефективність розпізнавання мови людини;
- надає можливість комп'ютеру розуміти інформацію, що буде отримана під час взаємодії з користувачем;
- надає можливість сформувати систему, яка буде розпізнавати різного роду лексеми та фонеми.

Якщо давати визначення дефініції «словник», то під нею потрібно розуміти певний набір слів, які у подальшому повинна отримувати та розпізнавати система. Потрібно зазначити, що усі слова повинні бути описані на тій самій мові, що й у граматиці. Зазначені слова будуть у першу чергу порівнюватися з отриманим системою звуковим сигналом[4].

Подією розпізнавання є період, протягом якого, людина надає інформацію системі завдяки тому, що каже певні слова або фрази у мікрофон [4].

Окремої уваги заслуговує шум. Надаючи визначення цієї дефініції у площині розпізнавання мови необхідно розуміти, що це певний об'єкт, до складу якого входять різного оду перешкоди, які не дають можливість ефективно розпізнати людську мову. Зазначене API видаляє відрізки з шумом завдяки порівнянню їх з шаблонами, які містить застосунок [3].

4.4. Приклад застосування Microsoft Speech Recognition API в програмному забезпеченні

Якщо казати глобально, то особливості використанння Microsoft Speech Recognition API можна описати певним алгоритмом дій, що складається з наступних етапів [11]:

1. В рамках програми зазначаємо граматику, яка буде досліджуватися та розпізнаватися системою.

2. Додаємо до системи словник, який необхідно використати для розпізнавання системою слів.

3. Виконуємо форматування словника таким чином, щоб він був відображений у зрозумілій для програмного забезпечення формі, а саме складався з байтів та бітів, з урахування існуючої граматики. У результаті виконання цього етапу отримуємо шаблони зі словами та словосполученнями.

4. Починаємо процес надання доступу програмі до необхідної інформації.

5. Користувач повинен почати говорити у мікрофон або будь-який інший інструмент для розпізнавання мови для того, щоб програма мала можливість розпізнати її.

6. API зазначеного застосунку починає процес розпізнавання отриманих звуків як частини певної мови. На цьому етапі має місце зв’язок з пунктами 1 та 2.

7. У випадку, якщо API розпізнало більше 78 % отриманої звукової інформації, то користувач може отримати підсумковий текст. Якщо відсоток розпізнавання менший, ніж 78 %, то система надає сповіщення про невдалу спробу.

8. Підсумковий текст видається користувачу на екрані пристрою, що він використовує для розпізнавання мови.

Використання зазначеного API можна використовувати під кожну операційну систему. Він однаково ефективно буде виконувати розпізнавання мови як у рамках веб-додатку, так й десктоп застосунку. Більш того, це API підходить під усі найбільш популярні мови програмування, серед яких JavaScript, Python, C++ тощо.

Для того, щоб відобразити якість роботи зазначеного API потрібно сформувати певну задачу. Наприклад, виконати розпізнавання людської мови

та визначити одне з чисел, що знаходяться у діапазоні від 0 до 10. У результаті роботи була отримана необхідна програма, яка може з ймовірністю 78 % розпізнати число, що знаходиться у діапазоні від 0 до 10. Результат виконання цього завдання системою відображено нижче:

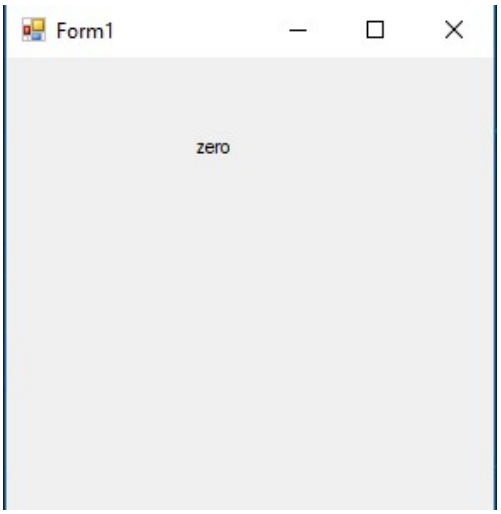

Рис.16 Реакція програми на слово "Zero"

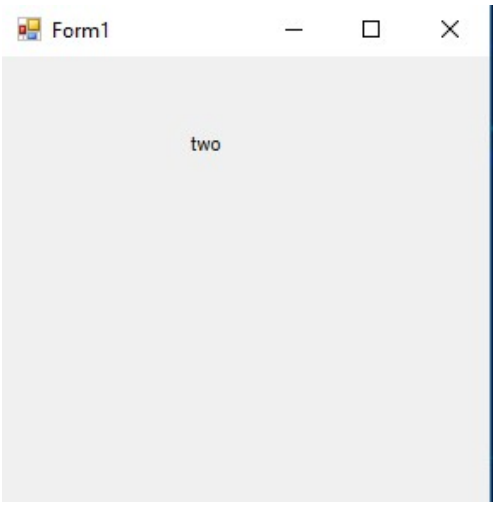

Рис.17 Реакція програми на слово "Two"

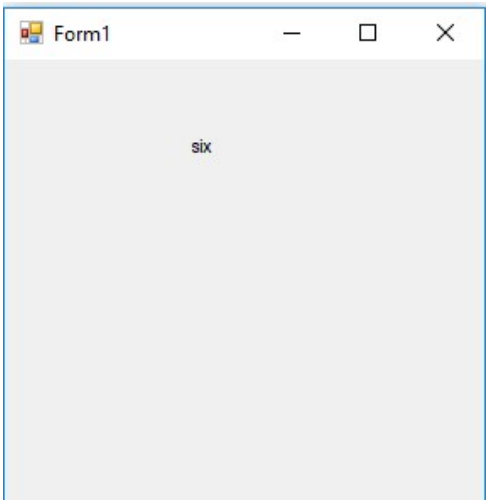

Рис.18 Реакція програми на слово "Six"

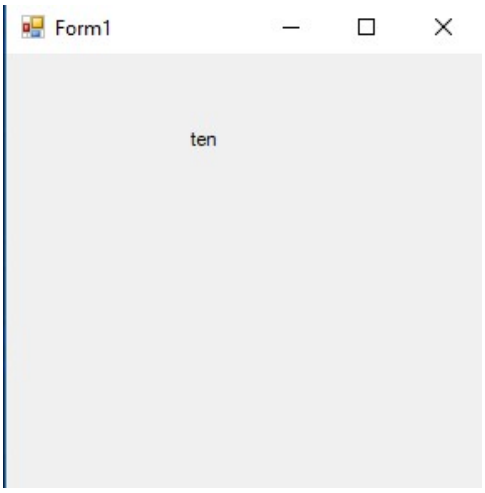

Рис.19 Реакція програми на слово "Ten"

Наприклад, якщо власник застосунку для розпізнавання мови та її подальшого перетворення на текст каже, наприклад: «zero», то відразу починають виконуватися закладені у систему алгоритми роботи. Зокрема,

виконується порівняння отриманої інформації з словником, що доданий до програми. Якщо відповідне слово знайдене, то воно автоматично відображається на екрані персонального комп'ютеру або смартфону.

Код програми можна переглянути в «Додатку 1».

Як ще один приклад інтеграції цього інструменту, я створив додаток, який допомагає перемикати слайди на презентації PowerPoint за допомогою голосу. Він також був реалізований мовою програмування C# і його вихідний код можна переглянути в «Додатку2».

Якщо детальніше, то додаток працює наступним чином:

1. Відкриває обраний файл-презентацію через COM-об’єкт .NET Framework, такі як MicrosoftOfficeInterop.
2. Запускає логіку розпізнавання мови, аналогічним чином, як це зроблено в додатку з розпізнаванням чисел.
3. Реагує на такі голосові команді як «nextslide», «previousslide», «go»та «exit» і виконує відповідні дії. Таким чином вирішуються одразу дві проблеми. Перша – для ведення презентації не потрібна мишка. Друга-додаток реагує лише на потрібні голосові команди, пропускає всі інші слова, тобто можна не думати про вибір слів, коли ведеш презентацію.
4. Закриває додаток, коли презентація закінчилась.

Таких прикладів для інтеграції можна придумати безліч. Об’єднує їх лише той факт, що з застосуванням MicrosoftSpeechRecognitionAPI більшість таких процесів можна звести до банально виконання голосових команд.

Висновки до розділу

В розділі IV (програмноге забезпечення) було побудовано діаграми порівняння сучасних фреймворків для розпізнавання мови, здійснено кількісне та якісне порівняння даних бібліотек та детально проаналізовано алгоритм роботи розробленого програмного додатку для керування слайдшоу за допомогою голосу.

РОЗДІЛ 5. РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

Ціль даного розділу розглянути можливість ринкового впровадження розробленого програмного продукту. Для цього ми провели аналіз стартап-проекту і визначили напрямки розвитку маркетингової стратегії.

5.1. Опис ідеї проекту

В даному підпункту розглядаються наступні питання:

- зміст ідеї (що пропонується);
- можливі напрямки застосування;
- основні вигоди, що може отримати користувач товару (за кожним напрямком застосування);
- чим відрізняється від існуючих аналогів та замінників;

Три перші пункти представлено у вигляді таблиці (табл.5.1.).

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	*Напрямки застосування*	*Вигоди для користувача*
Система розпізнавання природної мови	1. В навчальних цілях для учнів/студентів.	1.Простота у використанні
	2. В комерційних цілях	2. Персоналізація
	3. Крос-платформність	3. Можливість перенесення на іншу платформу, де присутній .NET

5.2. Технологічні стратегії проекту

В даному підрозділі проведено аудит технології, за допомогою якої реалізувано ідею проекту. Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (табл. 5.2):

- за якою технологією буде виготовлено товар згідно ідеї проекту?
- чи існують такі технології, чи їх потрібно розробити/доробити?
- чи доступні такі технології авторам проекту?

Таблиця 5.2. – Технологічна здійсненність ідеї проекту

№ п/п	*Ідея проекту*	*Технології її реалізації*	*Наявність технологій*	*Доступність технологій*
1	Створення програмного забезпечення	Visual Studio	Наявна	Безкоштовна, доступна
		.NET	Наявна	Безкоштовна, доступна
		WPF	Наявна	Безкоштовна, доступна

5.3. Аналіз ринкових можливостей запуску стартап-проекту

Визначимо ринкові можливості нашого проекту сплануємо напрямки розвитку програмного продукту із урахуванням стану ринку і потреб потенційних покупців та пропозицій конкурентних проектів. Для початку проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 5.3).

Таблиця 5.3 Попередня характеристика потенційного ринку стартап проекту

№ п/н	*Показники стану ринку (найменування)*	*Характеристика*
1	Кількість головних гравців, од	5
2	Загальний обсяг продаж, грн/ум.од	2250грн./ум.од
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі , %	R = (30000 * 100) / (1000000 * 12) = 23%

5.4. Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 5.4).

Таблиця 5.4 Вибір цільових груп потенційних споживачів

№ п/н	*Опис профілю цільової групи потенційн х клієнтів*	*Готовність споживачів сприйняти продукт*	*Орієнтовний попит в межах цільової групи (сегменту)*	*Інтенсивність конкуренції в сегменті*	*Простота входу у сегмент*
1.	Студенти	Високо	Середній	Існує декілька конкурентів, які надають	Перевага у тому, що застосунок є зручним для

				схожі, але менш швидкі та якісні рішення.	користування,
2.	Подорожуючі	Висока	Високий		Не велика конкуренція.

Визначено стратегію охоплення ринку-стратегія диференційованого маркетингу (компанія концентрується на декількох сегментах).

5.5. Розроблення маркетингової стратегії та маркетингового плану стартапу

Сформуємо маркетингову концепцію товару, який отримає клієнт. Для цього у табл. 5.5 підсумовано результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.5 Визначення ключових переваг концепції потенційного товару

№ п/н	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Ціна	Безкоштовний	Користувачу не потрібно платити зайві гроші.
2.	Спрощення інтерфейсу користувача	Пришвидшення роботи з ПЗ	Користувачам не потрібно замислюватись над тим, як зручно класифікувати інформацію

Розроблена трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання.

1-й рівень-При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язаний з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень-Цей рівень являє рішення того, як буде реалізований товар в реальному/ включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень-Товар з підкріпленням(супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін).

Фінальним етапом є захищення програмного продукту інтелектуальним правом від копіювання і встановлення. Визначено цінові межі в залежності від рівня доходів цільової групи споживачів.

Висновки до розділу

Дана система може бути цілком придатним першим варіантом для подальшої комерційної реалізації. Головною її перевагою для цього є низький рівень конкуренції через новизну ідеї системи. При правильному використанні реклами для просування системи можна досягнути значного комерційного успіху та високого доходу.

ВИСНОВКИ

Як результат роботи, було виконано наступні завдання:

1. Оглянути і аналізувати літературні джерела за темою «розпізнавання мови».
2. Дослідити особливості аудіопотоку та особливості мовлення, що впливають на процес розпізнавання мови.
3. Дослідити інструменти та засоби розпізнавання мови для побудови та інтеграції їх у прикладне програмне забезпечення

Підводячи підсумки дипломної роботи, необхідно зазначити, що головним елементом формування систем розпізнавання мови є формування алгоритмів для визначення вимовлених слів. Також, потрібно розуміти, що на сьогоднішній день, у Світі створено більше 2000 мов, а тому для кожної з них необхідно сформувати відповідні мовні моделі. Для того, щоб подібна система була максимально ефективною, велика кількість фахівців в галуз інформаційних технологій повинні якісно виконувати покладені на них обов’язки. До таких спеціалістів потрібно відносити нейрофізіологів, інженерів, фахівців по створенню мовних технологій тощо. Саме завдяки їх взаємодії між собою можна утворити якісну та продуктивну систему з розпізнавання людської мови.

Найбільш актуальним є статистичний підхід, якій використовує приховані Марківські моделі. Саме завдяки зазначеним моделям формується певна послідовність відтворення звукових сигналів у тексті. З іншої сторони, цей підхід має декілька недоліків, пов’язаних з наявністю обмежень. Як правило, сучасні науковці намагаються мінімізувати ці недоліки за допомоги використання різного роду нейромереж. Як вже зазначалося вище, яскравим прикладом такої системи є помічник Siri, який можуть використовувати власники сучасних смартфонів. Завдяки Siri користувачі можуть без проблем форматувати власний аудіо запис у текст.

Згідно з наведеною вище інформацією, можна прийти до висновку, що

найбільш продуктивна система розпізнавання мови повинна складатися з декількох етапів. Обробка вхідного сигналу, тобто тієї інформації, що надійшла від людини до машини. У процесі цієї обробки виконується не лише видалення шуму, але й сегментація аудіо ряду на певні сегменти. Додатково, виконується коригування отриманої інформації згідно даних, що мають місце у словниках, якими користується система для розпізнавання мови.

Необхідно розуміти, що не один з існуючих на сьогодні методів не може покрити абсолютно усі етапи. Для того, що система розпізнавання мови була найбільш ефективною для людини, вона повинна використовувати найбільш продуктивні методи на кожному з етапів, в тому числі й різного роду переваги. Яскравим прикладом є застосунок Watson, що створений компанією IBM. Кожна з систем, що була досліджена у цій дипломній роботи може використовуватися, як один з варіантів додатку для розпізнавання людської мови та перетворення її на текст.

Необхідно також зазначити особливості отримання інформації людиною. Вона не завжди сприймає ту інформацію, що чує. Як правило, майже половину мовних сигналів вона домислює. Згідно з цим, якщо інформаційні технології з розпізнавання людської мови отримають подальший розвиток, то вони повинні виходити за рамки безпосередньо мови, так як у процесі також використовуються інші складові. У подальшому обов’язково виникне необхідність створювати системи, які будуть не лише передавати мову через текст, а й коректно аналізувати її, продуктивно передаючи її зміст. Звісно, стрімкий розвиток інформаційних технологій в останні роки вказує на те, що алгоритми розпізнавання людської мови у подальшому будуть продовжувати покращуватися. Це безпосередньо пов’язано з тим, що людська спільнота намагається максимально автоматизувати свою життєдіяльність, передаючи частину робіт інформаційним системам. Поява якісної системи розпізнавання мови значно б пришвидшила сроки відправки текстових файлів.

СПИСОК ЛІТЕРАТУРИ

1. Плотников В. Н. Речевой диалог в системах управления / В. Н. Плотников, В. А. Суханов, Ю. Н. Жигулевцев. — М. : Машиностроение, 1988. — 223 с. — ISBN 5-217-00148-8.

2.Аграновский А. В. Теоретические аспекты алгоритмов обработки и классификации сигналов / А. В. Аграновский, Д. А. Леднов. — М. : Радио и связь, 2004. — 164 с.

3.Itakura F. Minimum Prediction Residual Principle Applied to Speech Recogition : праці наук. конф., Лютий 1975, IEEE Trans. Acoustics, Speech, and Signal roc, 1975. — Т. 23, № 1. — 72с.

4. N.B. Vasylieva, D.Ja. Fedoryn International Research and Training Center of Information Technologies and Systems, Kiev, Ukraine – 10 с.

5.Рабинер Л. Теория и применение цифровой обработки сигналов / Л. Рабинер, Б. Гоулд. — М. : Мир, 1978. — 848с.

6. Биков М. Дикторонезалежне описання образів в системах розпізнавання сигналів мови / Микола Биков, Абдурахман Раїмі, Максим Биков // Вимірювальна техніка та метрологія. Збірник наукових праць. — 2006. — № 66. — С. 13—17.

7.Waheed K. A robust algorithm for detecting speech segments using an entropy contrast : праці міжн. конф. 45th IEEE InternationalMidwestSymposiumonCircuitsandSystemsMWSCAS'2002,4-7серп.2002,Oklahoma(USA).—С.328—331,III.

8.Shen J.-L., Robust Entropy-based Endpoint Detection for Speech Recognition in Noisy Environments : праці міжн. конф., 30 лист. — 4 груд. 1998, 5th International Conference on Spoken Language Processing, Sydney(Australia).

9.M. Fujimoto. Evaluation of noisy speech recognition based on noise reduction and acoustic model adaptation on the AURORA2tasks:працімiжн.конф.,вер.2002,SpokenLang.Processing'ICSLP'2002,Denver(USA),2000—С.465—468,I.

10.Рабинер Р. Л. Цифровая обработка речевых сигналов / Р. Л. Рабинер. — М. : Радио и связь, 1981. — 495с.

11.Офіційний інформаційний веб-ресурс компанії Microsoft: https://msdn.microsoft.com

12.Офіційний інформаційний веб-ресурс продукту «CreedVocal» :http://www.creaceed.com/ceedvocal/about

13.Офіційний інформаційний веб-ресурс продукту «VoConHybrid» : http://www.nuance.com/for-business/by-product/automotiveproducts-services/vocon-hybrid/

14. Офіційна GitHubсторінка системи Julius : https://github.com/julius-speech/julius

15.Офіційний інформаційний веб-ресурс продукту «YandexSpeechKit» : https://tech.yandex.ru/speechkit/

16.Офіційний інформаційний веб-ресурс продукту «GoogleSpeechAPI» :https://cloud.google.com/speech-to-text/

17. Офіційний інформаційний веб-ресурс продукту IBM«Watson» :https://www.ibm.com/watson/

ДОДАТКИ

Додаток 1. Фрагмент вихідного коду приклад-програми «Розпізнай число».

```
//Задаємо мовні особливості граматики.
CultureInfo ci = new CultureInfo("en-US");
SpeechRecognitionEngine sre = new SpeechRecognitionEngine(ci);
sre.SetInputToDefaultAudioDevice();

//Підписуємо на нову подію розпізнавання мови.
sre.SpeechRecognized += new EvenHandler<SpeechRecognizedEventArgs>(sre_SpeechRecognized);


//Описуємо словник бажаних слів.
Choices numbers = new Choices();
numbers.Add("zero","one", "two", "three", "four", "five", "six", "seven", "eight",
"nine","ten");

//Створюємо граматику на основі мовної моделі і словника.
GrammarBuilder gb = new GrammarBuilder();
gb.Culture = ci;
gb.Append(numbers);

Grammar g = new Grammar(gb);
sre.LoadGrammar(g);

//Запускаємо процедуру розпізнавання.
sre.RecognizeAsync(RecognizeMode.Multiple);

//Обровник події розпізнавання мови.
staticvoid sre_SpeechRecognized(object sender, SpeechRecognizedEventArgs e)
{
if (e.Result.Confidence > 0.78) l.Text = e.Result.Text;
}
```

Додаток 2. Фрагмент вихідного коду програми «Помічник ведення презентації».

```
using System;
using System.Threading;
using System.Windows.Forms;

using Microsoft.Office.Interop.PowerPoint;
using Microsoft.Office.Core;
using System.Diagnostics;

using Microsoft.Speech.Recognition;


namespace DiplomaProject
{
publicpartialclassForm1 : Form
    {
public Form1()
        {
            InitializeComponent();
        }

static Label l;

conststring path = @"C:\Users\Admin\Desktop\Test.pptx";
static Microsoft.Office.Interop.PowerPoint.Application ppApp;
static Presentations objPresSet;
static Presentation objPres;


//speech handler
staticvoid sre_SpeechRecognized(object sender, SpeechRecognizedEventArgs e)
        {
if (e.Result.Confidence > 0.82)
            {
                l.Text = e.Result.Text;

if(l.Text == "go")
                {
                    ppApp = new Microsoft.Office.Interop.PowerPoint.Application();

                    ppApp.Visible = MsoTriState.msoTrue;

                    objPresSet = ppApp.Presentations;

                    objPres = objPresSet.Open(path, MsoTriState.msoFalse,
MsoTriState.msoTrue, MsoTriState.msoFalse);

                    objPres.SlideShowSettings.Run();
                }
elseif(l.Text == "end")
                {
                    objPres.Close();
                    ppApp.Quit();
                    System.Runtime.InteropServices.Marshal.ReleaseComObject(ppApp);

conststring processToKill = "POWERPNT";

var processes = Process.GetProcessesByName(processToKill);
foreach (Process p in processes)
                    {
```

```
                        p.Kill();
                    }
                }
elseif(l.Text == "next slide")
                {
                    objPres.SlideShowWindow.View.Next();
                }
elseif(l.Text == "previous slide")
                {
                    objPres.SlideShowWindow.View.Previous();
                }
            }

        }

//Power Point Open Event
privatevoid button1_Click(object sender, EventArgs e)
        {
            Microsoft.Office.Interop.PowerPoint.Application ppApp = new
Microsoft.Office.Interop.PowerPoint.Application();
            Presentations objPresSet;
            Presentation objPres;

            ppApp.Visible = MsoTriState.msoTrue;

            objPresSet = ppApp.Presentations;

            objPres = objPresSet.Open(path, MsoTriState.msoFalse, MsoTriState.msoTrue,
MsoTriState.msoFalse);

            objPres.SlideShowSettings.Run();

            Console.WriteLine("Staring presentation...");

            Thread.Sleep(1000);

for (var i = 0; i < objPres.Slides.Count - 1; i++)
            {
                objPres.SlideShowWindow.View.Next();
                Thread.Sleep(1000);
            }


            objPres.Close();

//ppApp.Visible = MsoTriState.msoFalse;

            ppApp.Quit();
            System.Runtime.InteropServices.Marshal.ReleaseComObject(ppApp);

conststring processToKill = "POWERPNT";

var processes = Process.GetProcessesByName(processToKill);
foreach (Process p in processes)
            {
                p.Kill();
            }


        }

//here we initialize speech recognition objects and logic
```

```
privatevoid Form1_Shown(object sender, EventArgs e)
        {

            l = recognizedText;

//language configuration
            System.Globalization.CultureInfo ci = new System.Globalization.CultureInfo("en-
US");
            SpeechRecognitionEngine sre = new SpeechRecognitionEngine(ci);
            sre.SetInputToDefaultAudioDevice();

            sre.SpeechRecognized += new
EventHandler<SpeechRecognizedEventArgs>(sre_SpeechRecognized);

//choices configuration
            Choices choices = new Choices();
            choices.Add(newstring[] {"_","go","end","next slide","previous slide" });

//grammar builder
            GrammarBuilder gb = new GrammarBuilder();
            gb.Append(choices);

            Grammar g = new Grammar(gb);
            sre.LoadGrammar(g);

            sre.RecognizeAsync(RecognizeMode.Multiple);
        }
    }
}
```