НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

«На правах рукопису»
УДК 004.522

«До захисту допущено»

Завідувач кафедри СПСКС

_____
В.П.Тарасенко_
(підпис) (ініціали, прізвище)

"___"_____2018р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 123 Комп'ютерна інженерія
(Спеціалізовані комп'ютерні системи)
на тему: «Метод розпізнавання команд голосового управління комп'ютерною системою»

Виконав: студент ІІ курсу, групи КВ-63м
(шифр групи)

Шуліка Владислав Павлович_____ _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник д.т.н., доцент Терейковський І.А. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – другий (магістерський)
Спеціальність 123 Комп'ютерна інженерія
(Комп'ютерні системи та компоненти)

ЗАТВЕРДЖУЮ
Завідувач кафедри СПСКС
_____
_В.П.Тарасенко__
(підпис) (ініціали, прізвище)
«___»_____2018р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Шуліка Владислав Павлович
(прізвище, ім'я, по батькові)

1. Тема дисертації «Метод розпізнавання команд голосового управління комп'ютерною системою»,
науковий керівник дисертації Терейковський І.А., д.т.н., доцент _____ ,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «22» березня 2018 р. №986-с

2. Термін подання студентом дисертації 11 травня 2018 р.

3. Об'єкт дослідження є процес розпізнавання голосової команди для керування комп'ютерної системи.
4. Предмет дослідження є способи розпізнавання голосової команди для керування комп'ютерної системи на основі методу динамічного програмування.

5. Перелік завдань, які потрібно розробити: проаналізовати існуючі системи та алгоритми розпізнавання голосових команд; запропонувати модифікований спосіб пошуку голосової команди; виконати(дослідження)

тестування системи розпізнавання голосових команд при різних умовах; розробити систему спілкування людини з комп'ютерною системою.

6. Перелік ілюстративного матеріалу блок-схема етапів попередньої обробки мовного сигналу, блок-схема загальної роботи з .wav файлами, блок-схема деталізованої роботи програми, UML діаграма класів, блок-схема загальної роботи з голосовою командою, схема компонентів систем розпізнавання мови.

7. Перелік публікацій: X наукова конференція магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2018 (Київ, 21-23 березня 2018 р.), I міжнародна науково-практична конференція «Проблеми кібербезпеки інформаційно-телекомунікаційних систем (PCSISTS)» (Київ, 05-06 квітня 2018 р.) Київський національний університет імені Тараса Шевченка, міжнародна науково-практична конференція студентів і молодих учених «Інформаційні технології в соціокультурній сфері, освіті, економіці та права» (Київ, 18-19 квітня 2018 р.) Київський національний університет культури і мистецта.

8. Дата видачі завдання 5 вересня 2016 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Огляд літератури по темі роботи	01.10.2016-05.12.2016	
2	Дослідження ефективності LBP операторів	05.02.2017-10.04.2017	
3	Розробка системи розпізнавання облич	10.08.2017-15.12.2017	
4	Тестування розробленої системи	20.12.2017-05.01.2018	
5	Оформлення пояснювальної записки	15.01.2018-10.04.2018	
6	Оформлення ілюстративного матеріалу	12.04.2018-24.04.2018	
	Попередній розгляд магістерської дисертації на кафедрі	26.04.2018	

Студент _____ _____
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ _____
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Актуальність теми. Мова є найбільш природною формою людського спілкування і тому реалізація інтерфейсу на основі аналізу мовної інформації є перспективним напрямком розвитку інтелектуальних систем управління.

Задача розпізнавання мовної інформації є складною задачею, яка використовує такі області науки як: цифрова обробка сигналів, розпізнавання образів та лінгвістика.

Діалог з комп'ютерами, роботами, автоматизованими системами управління за допомогою голосових повідомлень відкриває великі перспективи:

- простота спілкування з системою;
- доступність мовного інтерфейсу людям з порушеннями опорно-рухового та зорового апарату;
- можливість роботи користувачів в умовах перевантаженості тактильно-зорового каналу.

Об'єктом дослідження є процес розпізнавання голосової команди для керування комп'ютерної системи.

Предметом дослідження є способи розпізнавання голосової команди для керування комп'ютерної системи на основі методу динамічного програмування.

Мета роботи: прискорення процесу розпізнавання голосової команди для керування комп'ютерної системи на основі методу динамічного програмування, розробка більш точної системи розпізнавання голосової команди на основі додаткового аналізу.

Наукова новизна:

1. Проаналізовано існуючі системи та алгоритми розпізнавання голосових команд та показано, що ці алгоритми мають недоліки у їх

використанні за різними показниками: час, неточність та надлишкове використання системних ресурсів.

2. Запропоновано модифікований спосіб пошуку голосової команди по набору еталонних команд, який відрізняється від стандартного тим, що для однієї команди використовується цілий масив видозміненої промови певної команди.

3. Виконано(дослідження) тестування системи розпізнавання голосових команд при різних умовах: відстань, напрямок, шум, інтонація.

Практична цінність отриманих в роботі результатів полягає в тому, що розроблений модифікований спосіб дозволяє прискорити процес розпізнавання голосових команд для управління комп'ютерної системи. Крім того, запропонована система може бути використана у різних сферах для полегшення роботи з комп'ютером. Розроблена в роботі програмна реалізація системи для розпізнавання голосових команд може бути використана для побудови розумного будинку чи автомобіля.

Апробація роботи. Основні положення і результати роботи були представлені та обговорювались на X науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2018 (Київ, 21-23 березня 2018 р.), на I міжнародній науково-практичній конференції «Проблеми кібербезпеки інформаційно-телекомунікаційних систем (PCSISTS)» (Київ, 05-06 квітня 2018 р.) Київський національний університет імені Тараса Шевченка, на міжнародній науково-практичній конференції студентів і молодих учених «Інформаційні технології в соціокультурній сфері, освіті, економіці та права» (Київ, 18-19 квітня 2018 р.) Київський національний університет культури і мистецта.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, чотирьох розділів та висновків.

У вступі подано загальну характеристику роботи, зроблено оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень, сформульовано мету і задачі досліджень, показано наукову

новизну отриманих результатів і практичну цінність роботи, наведено відомості про апробацію результатів і їхнє впровадження.

У першому розділі розглянуто існуючі алгоритми для розпізнавання голосу, їхні особливості, недоліки та переваги.

У другому розділі розглянуто опис засобів розробки та організація програмних засобів.

У третьому розділі наведено особливості реалізації розробленої системи.

У четвертому розділі представлено підходи до тестування системи в цілому та окремих модулів.

У висновках представлені результати проведеної роботи.

Робота представлена на 80 аркушах, містить посилання на список використаних літературних джерел.

Ключові слова: розпізнавання голосової команди, метод динамічного програмування.

Реферат

Актуальность темы. Язык является наиболее естественной формой человеческого общения и поэтому реализация интерфейса на основе анализа речевой информации является перспективным направлением развития интеллектуальных систем управления.

Задача распознавания речевой информации является сложной задачей, которая использует такие области науки как: цифровая обработка сигналов, распознавания образов и лингвистика.

Диалог с компьютерами, работами, автоматизированными системами управления с помощью голосовых сообщений открывает большие перспективы:

- простота общения с системой;

- доступность речевого интерфейса людям с нарушениями опорно-двигательного и зрительного аппарата;

- возможность работы пользователей в условиях перегруженности тактильно-зрительного канала.

Объектом исследования является процесс распознавания голосовых команд для управления компьютерной системы.

Предметом исследования являются способы распознавания голосовых команд для управления компьютерной системы на основе метода динамического программирования.

Цель работы: ускорение процесса распознавания голосовых команд для управления компьютерной системы на основе метода динамического программирования, разработка более точной системы распознавания голосовых команд на основе дополнительного анализа.

Научная новизна:

1. Проанализированы существующие системы и алгоритмы распознавания голосовых команд и показано, что эти алгоритмы имеют недостатки в их использовании по разным показателям: время, неточности и избыточное использование системных ресурсов.

2. Предложен модифицированный способ поиска голосовой команды по набору эталонных команд, который отличается от стандартного тем, что для одной команды используется целый массив видоизмененной речи определенной команды.

3. Выполнено (исследование) тестирование системы распознавания голосовых команд при различных условиях: расстояние, направление, шум, интонация.

Практическая ценность полученных в работе результатов заключается в том, что разработан модифицированный способ позволяет ускорить процесс распознавания голосовых команд для управления компьютерной системы. Кроме того, предложенная система может быть использована в различных сферах для облегчения работы с компьютером. Разработанная в работе программная реализация системы для распознавания голосовых команд может быть использована для построения умного дома или автомобиля.

Апробация работы. Основные положения и результаты работы были представлены и обсуждались на X научной конференции магистрантов и аспирантов «Прикладная математика и компьютинг» ПМК-2018 (Киев, 21-23 марта 2018), на I международной научно-практической конференции «Проблемы кибербезопасности информационно телекоммуникационной систем (PCSISTS) »(Киев, 05-06 апреля 2018) Киевский национальный университет имени Тараса Шевченко, на международной научно-практической конференции студентов и молодых ученых« Информационные технологии в социокультурной сфере, образовании, экономике и права »(Киев, 18-19 апреля 2018) Киевский национальный университет культуры и искусства.

Структура и объем работы. Магистерская диссертация состоит из введения, четырех глав и выводов.

Во введении представлена общая характеристика работы, произведена оценка современного состояния проблемы, обоснована актуальность

направления исследований, сформулированы цели и задачи исследований, показано научную новизну полученных результатов и практическую ценность работы, приведены сведения об апробации результатов и их внедрение.

В первой главе рассмотрены существующие алгоритмы для распознавания голоса, их особенности, недостатки и преимущества.

Во втором разделе рассмотрены описание средств разработки и организация программных средств.

В третьем разделе приведены особенности реализации разработанной системы.

В четвертом разделе представлены подходы к тестированию системы в целом и отдельных модулей.

В выводах представлены результаты проведенной работы.

Работа представлена на 80 листах, содержит ссылки на список использованных литературных источников.

Ключевые слова: распознавание голосовых команд, метод динамического программирования.

ABSTRACT

Actuality of theme. Language is the most natural form of human communication, and therefore the implementation of the interface based on the analysis of language information is a promising direction for the development of intelligent management systems.

The task of recognizing language information is a complex task that uses such fields of science as: digital signal processing, image recognition and linguistics.

Dialogue with computers, robots, automated control systems by means of voice messages offers great prospects:

- Simple communication with the system;

- availability of the language interface for people with musculoskeletal and visual disorders;

- the ability to work users in conditions of overload tactile-visual channel.

The object of the study is the process of recognizing a voice command to control the computer system.

The subject of the study is the methods of recognizing a voice command to control a computer system based on the method of dynamic programming.

The purpose of the work is to accelerate the recognition process of the voice command for managing the computer system on the basis of the dynamic programming method, the development of a more accurate voice recognition system based on additional analysis.

Scientific novelty:

1. The existing systems and algorithms of recognition of voice commands are analyzed and it is shown that these algorithms have disadvantages in their use on different indicators: time, inaccuracy and excessive use of system resources.

2. A modified method for searching a voice command for a set of reference commands is proposed, which differs from the standard one by using the whole array of modified command speech for a particular command.

3. Performed (research) testing of voice recognition recognition system under different conditions: distance, direction, noise, intonation.

The practical value of the results obtained in the work is that the developed modified method allows accelerating the process of recognizing voice commands for managing a computer system. In addition, the proposed system can be used in various areas to facilitate the work of the computer. The program implementation of the system for recognizing voice commands developed in the work can be used to build a smart home or car.

Test work. The main provisions and results of the work were presented and discussed at the Xth Scientific Conference of Master and Postgraduate Students "Applied Mathematics and Computer", PMK-2018 (Kyiv, March 21-23, 2018), at the I International Scientific and Practical Conference "Cybersecurity Information Issues Telecommunication Systems (PCSISTS) »(Kiev, April 5-5, 2018) Taras Shevchenko National University of Kyiv, International Scientific and Practical Conference of Students and Young Scientists« Information Technologies in Socio-Cultural, Education, Economics Law "(Kyiv, 18-19 April 2018) Kyiv National University of Culture and Arts.

Structure and scope of work. The master's thesis consists of an introduction, four chapters and conclusions.

The introduction gives a general description of the work, assesses the current state of the problem, substantiates the relevance of the research direction, formulates the purpose and objectives of the research, shows the scientific novelty of the results obtained and the practical value of the work, provides information on the approbation of the results and their implementation.

The first chapter deals with existing algorithms for voice recognition, their features, disadvantages and advantages.

The second section describes the description of development tools and the organization of software.

The third section presents the peculiarities of the implementation of the developed system.

The fourth section presents approaches to testing the system as a whole and individual modules.

The conclusions are the results of the work.

The work is presented on 80 sheets, contains a link to the list of used literary sources.

Keywords: recognition of voice command, dynamic programming method.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ПММ – приховані Марківські моделі.

АРМ – автоматичне розпізнавання мови.

MFCC – мел-кепстральні коефіцієнти.

URL (від англ. Uniform Resource Locator) – узагальнений покажчик місцезнаходження ресурсу.

ANN (від англ. Artificial Neural Network -) - штучні нейронні мережі.

HTTP (від англ. HyperText Transfer Protocol) – протокол передачі даних прикладного рівня.

ASR в(від англ. Automatic Speech Recognition) – автоматичне розпізнавання мови.

DWT (Dynamic Time Warping) – алгоритм динамічного програмування часу.

ВСТУП

З моменту появи перших ЕОМ одним з найбільш важливих питань розвитку комп'ютерної техніки був процес взаємодії людини з машиною. Довгий час це було доступно тільки вузьким фахівцям-технологам «спілкувалися» з машиною через посередника - програміста. Така ситуація проіснувала аж до появи діалогового інтерфейсу, коли користувач зміг особисто вводити з клавіатури адресовану машині команду і отримувати осмислений відповідь. Поява графічного інтерфейсу, при якому відпала необхідність в знанні людиною будь-яких команд, привела до повсюдного поширення персональних комп'ютерів.

Однак людина завжди прагнув до більш універсального і природного способу взаємодії з ЕОМ. Ще в епоху перфокарт в науково-фантастичних романах людина розмовляв з комп'ютером, як з рівним собі. Тоді ж були зроблені перші кроки по реалізації мовного інтерфейсу.

Проте, якщо порівняти показники сучасних систем розпізнавання з показниками систем часів початку зародження цієї галузі науки, то можна сказати, що за минулі десятки років дослідники недалеко просунулися. Це змушує деяких фахівців сумніватися щодо можливості реалізації мовного інтерфейсу в найближчому майбутньому. Інші вважають, що завдання вже практично вирішена. Більшість експертів сходиться на думці, що для розвитку розпізнавання мови буде потрібно якийсь час[1].

Останнім часом спостерігається стрімкий розвиток інформаційних технологій. Одним з пріоритетних напрямків досліджень в даній області є завдання зберігання, обробки і передачі даних мультимедіа. На жаль, до сих пір у багатьох задачах аналізу мультимедіа даних комп'ютер так і не зміг остаточно замінити експерта. Це такі завдання, як синхронний переклад, автоматична сегментація зображень і відеопослідовностей, автоматична стенографія. Однією з основних завдань обробки мультимедіа інформації є задача розпізнавання та аналізу природної мови людини.

У завдання аналізу мови входить широкий спектр завдань. Традиційно їх поділяють на три підкласи: завдання ідентифікації, класифікації та діагностики. До завдань ідентифікації відносять завдання верифікації та ідентифікації дикторів. До завдань класифікації відносять завдання розпізнавання ключових слів, розпізнавання злитого мовлення і завдання семантичного аналізу мови. До класу задач діагностики відносять завдання визначення психофізичного стану диктора. У багатьох з вище перерахованих завдань в останні роки було досягнуто значного прогресу. Скажімо, алгоритми ідентифікації або верифікації дикторів широко використовуються при проведенні криміналістичних процедур або для розмежування прав доступу, завдяки високій точності розроблених методів.

Як і раніше зберігає свою актуальність завдання розпізнавання мови. Область застосування отриманих рішень досить обширна: автоматичні стенографії, автоматизовані довідкові термінали з голосовим керуванням, синхронні перекладачі, системи стиснення і передачі мовного сигналу з високою якістю, системи сегментації, індексації та пошуку мультимедіа інформації.

Мова є найбільш природною формою людського спілкування і тому реалізація інтерфейсу на основі аналізу мовленнєвої інформації є перспективним напрямком розвитку інтелектуальних систем управління. Однією з актуальних невирішених проблем у галузі інформаційно-вимірювальних систем є побудова систем автоматичного розпізнавання мовленнєвих сигналів, інваріантних до диктора. Її вирішення дало б змогу розширити коло користувачів таких систем і значно підвищити ефективність обміну інформацією в людино-машинних системах. Реалізація мовного інтерфейсу є дуже складною технічною задачею, розв’язання якої знаходиться на стику багатьох галузей науки. Так при сприйнятті мови людина використовує механізми асоціативного аналізу, не просто розбираючи і порівнюючи почуті звуки, але збираючи фонеми в словесні образи, підбираючи найбільш відповідні слова не тільки по звуковій

подібності, але і по інтонації, емоційному забарвленню, контексту слова, фрази, речення і навіть всього тексу. Тому, людина здатна розпізнавати мову навіть при великому браку несучої інформації. Наприклад, людина набагато вимогливіша до якості звуку при прослуховуванні тексту на чужій мові яку вона погано знає, ніж при сприйнятті рідної мови. Ще не настав час безпосереднього впровадження мовного інтерфейсу в повсякденне життя кінцевого користувача, однак наявний на даний час прогрес важко переоцінити. Програми і системи, що володіють засобами мовного введення інформації, одержують усе більше поширення, але, з огляду на всі їх недоліки, варто розглядати перспективи розвитку вузькоспеціалізованих систем, що мають чітке застосування[2]:

— системи контролю присутності людини;

— аутентифікація та контроль доступу;

— телефонний банкінг;

— голосовий ввід інформації, що замінює текстовий набір;

— автоматизовані системи заповнення анкет та шаблонних інформаційних листів;

— біометрична реєстрація в різноманітних і різнонаправлених телефонних системах;

— розробка інформаційно-довідкових служб різного призначення, в яких клієнт запитує довідки, данні, що його цікавлять й одержує інформацію в мовній або іншій формі; телефоні лінії підтримки клієнтів, електронна комерція;

— управління системами життєзабезпечення для людей з обмеженими фізичними можливостями та побудови систем інтелектуалізації житла, так звані «розумні дома»[3];

— управління освітленням, водопостачанням, опалюванням, кондиціонуванням повітря тощо;

— створення індивідуальних автоматичних систем перекладу з одної мови на іншу, що працює у реальному часі;

— судові експертизи, зокрема системи відтворення спотворених та зашумлених мовних повідомлень;

— в майбутньому - пошук в інформаційних мережах мовної інформації за заданими ключовими словами або проблематикою.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

1.1. Класифікація існуючих комп'ютерних систем

Автоматичне розпізнавання мови - це процес автоматичної трансляції комп'ютерами або іншими машинами усної людської мови в текстовий формат. Незважаючи на таке просте формулювання завдання, над її вирішенням вже більше пів століття борються дослідники з усього світу. Людська мова сама по собі дуже складний об'єкт, на який впливають такі чинники, як освіта, фізичний стан і психологія людини, діалект або специфіка вимови людини, контекст, в якому вимовляється мова, а також специфіка самої мови, включаючи її фонетику, фонологію і граматику. Крім того, при аналізі мовних сигналів потрібно враховувати і зовнішні чинники такі, як фонові шуми, приміщення, відстань до пристроїв захоплення мови, зміни в каналі зв'язку та інше. У зв'язку з цим, вивчення мови і проведення досліджень в області її автоматичного розпізнавання - це міждисциплінарний процес, який вимагає знань з таких областей, як[4]:

- цифрова обробка сигналу, де потрібно витягти максимально корисну інформацію з сигналу з урахуванням умов, в яких було отримано даний сигнал;
- фізика, яка дає зрозуміти зв'язок між отриманим цифровим сигналом і процесами вимови і сприйняття мови;
- фонетика і фонологія, що описують систему звуків даної мови і взаємозв'язок між цими звуками;
- лінгвістика, що дає розуміння зв'язків між звуками і словами, їх значеннями і будівлями всередині пропозицій;
- теорія інформації, яка дозволяє оцінити наявність мови в сигналі
- представити її в компактному і закодованому вигляді, зручному для подальшої машинної обробки;

- розпізнавання образів, область науки, яка виявляє закономірності в мовних сигналах, дозволяє класифікувати і розрізняти між тими чи іншими звуками, словами та іншими об'єктами;
- інформатика, наука про ефективні комп'ютерні алгоритми і способи їх програмної і апаратної реалізації, які можуть бути використані в системах розпізнавання мови;
- математика, що є невід'ємною теоретичною основою, методів і алгоритмів, що розробляються.

Системи розпізнавання мови можна класифікувати за рядом ознак, в залежності від яких можуть змінюватися застосовувані підходи вирішення задачі розпізнавання мови. Наведемо лише основні типи ознак[5].

За типом мови, що розпізнається:

- окремі слова;
- пов'язані слова і фрази;
- злита мова.

Ключова різниця розпізнавання окремих слів від інших типів заключається у можливості визначеняя початку і кінця слова, що набагато складніше в інших ситуаціях. Відповідно, перша задача спрощується і не потребує використання занадто складних підходів[6].

За залежності від диктора:

- дикторо-залежні;
- дикторо-незалежні.

Варіації в вимові між дикторами створюють додаткову проблему для систем розпізнавання мови. Тому найбільш складним завданням з двох типів завдань є створення диктор-незалежної системи, в якій потрібно застосовувати методи нормалізації міждикторських варіацій. Рішення ж першого завдання зводиться або до навчання системи на основі голосу одного диктора або до адаптації існуючої дикторо-незалежної системи до нового диктора.

За якістю розпізнавання мови[7]:

- чиста мова (SNR> 40дБ);
- слабо зашумлена мова (SNR ~ 20-40дБ);
- сильно зашумлена мова (SNR <20 дБ).

Під чистою мовою розуміється аудіо сигнал, що містить тільки людську мову без будь-яких сторонніх звуків, що не відносяться до мови. На практиці ж зазвичай немає ідеально чистих від шумів аудіо даних через наявність зовнішніх перешкод (відлуння, сторонні звуки, та ін.), Технічних перешкод, що утворюються в процесі захоплення і передачі аудіо сигналу, а також перешкод, що виникають під час вимови самої мови . Розпізнавання мови в умовах шумів - це складна і не завжди розв’язувана задача.

За способом розбиття мови на елементарні одиниці[8]:

- розпізнавання по фонемам;
- розпізнавання по частинам слів (склади, інтервали);
- розпізнавання по словам.

Людська мова формується шляхом вимови окремих звуків, які, об'єднуючись, утворюють осмислені слова і речення. Так, при створенні систем розпізнавання мови можна виділити такі неподільні природні елементарні одиниці, як фонеми, склади і слова або ж визначити інші штучно створені одиниці. Якість розпізнавання від вибору тих чи інших типів елементарних одиниць залежить не однозначно, тому тут питання вибору залежить більше від завдань, потреб і внутрішньої інтуїції розробників.

За розміром словникової бази:

- з малим словником близько 100 слів;
- з невеликим словником близько 1000 слів;
- з великим словником близько 5000 і більше слів.

Розмір словникової бази визначає підходи для вирішення задачі розпізнавання мови. Зокрема, системи розпізнавання мови з великим

словником часто використовують фонеми або склади для розбиття слів, а для малих розмірів словника може бути достатнім використання слова як елементарної одиниці.

За типом словникової бази[9]:

- певна (замкнута) словникова база;
- невизначена (необмежена) словникова база.

Розпізнавання мови може проводитися в просторі слів, які наперед задані і не змінюються, або на всьому просторі природної мови, допускаючи слова, відсутні в словнику. Останнє являється складним завданням, як в обчислювальному, так і в алгоритмічному відношенні.

За типом граматики[10]:

- фіксована граматика;
- природна граматика.

Крім словникової бази розпізнавання мови пов'язано і з граматикою або структурою виразів, які необхідно розпізнавати. Це можуть бути вирази з певною структурою і синтаксисом, тобто з фіксованою граматикою, або із загальною структурою, властивою природній мові.

Відповідно до положення пристрою захоплення звуку:

- близьке розташування (до 20 см);
- далеке розташування (більше 20 см).

Важливим фактором при побудові систем розпізнавання мови є відстань від диктора до пристрою захоплення мови (мікрофона, телефону та ін.), Тому що від цього може залежати ступінь присутності сторонніх шумів або ехо, що безпосередньо впливає на ставлення «сигнал-до-шуму» (SNR)[11].

- За типом вирішуваних завдань:
- розпізнавання окремих команд;
- розпізнавання злитого мовлення;
- розпізнавання телефонного мовлення;

- розпізнавання новин, лекцій, та ін .;
- розпізнавання діалогів;
- розпізнавання спонтанного мовлення.

Дана класифікація не є класифікацією як такою, а швидше позначає коло основних завдань, які представляють певний інтерес з боку наукової спільноти через їх практичну цінність і труднощі рішення.

Сучасні системи автоматичного розпізнавання мови засновані на застосуванні статистичного підходу, суть якого полягає в обробці великого обсягу мовних даних з метою побудови адекватної акустичної моделі [12]. Колекція таких мовних даних разом з їх орфографічними транскрипціями називається акустичним корпусом. За типом озвучування тексту існує два види акустичних корпусів - корпуси, що містять начитаний матеріал (новини, статті, слова та ін.) і спонтанну мова (діалоги, лекції, ін.).

Більшість систем розпізнавання мови (Automatic Speech Recognition - ASR) складається з процесу аналізу і обробки аналогового сигналу і процесу розпізнавання. При аналізі аналогового сигналу з промови виділяються властивості, які використовуються далі в процесі розпізнавання для того, щоб визначити, що було сказано. Розглянемо коротку історію розвитку систем ASR в контексті цих двох процесів [13].

Найперші спроби створення ASR систем здійснювалися в 1950-х роках. Була побудована залежна від диктора система, розпізнавати цифри [14]. Як властивості сигналу використовувалися спектральні резонанси голосних в словах. У 1959 році був створений модуль, здатний розпізнавати десять голосних незалежно від диктора [15].

У 60-х роках в Японії було побудовано кілька машин, які розпізнавали голосні звуки, використовуючи спеціальний спектральний аналізатор [16]. Також було створено пристрій, що розпізнає фонеми [17].

У 70-х роках в області розпізнавання мови було скоєно два значних відкриття: використання методу динамічного програмування (Dynamic Time

Warping - DTW) [18], засноване на тимчасовому вирівнюванні мовних діалектів, і метод кодування лінійного передбачення (Linear Predictive Coding - LPC) [19], який успішно використовувався в розпізнаванні сигналів з низьким бітрейтом (кількість бітів інформації, переданих в секунду). У AT & T Bell Laboratories були побудовані системи розпізнавання, обробка акустичного сигналу в яких була заснована на LPC аналізі, а процес розпізнавання на DTW [20].

У 80-х рр від підходів, заснованих на застосуванні шаблонів, дослідження в області розпізнавання мовлення перейшли до методів статистичного моделювання. Використовувалися приховані моделі Маркова (Hidden Markov Models - HMM). Роботи Бейкера [21] були одними з перших, в яких для вирішення задачі розпізнавання мови були застосовані HMM. В кінці 80-х рр до проблеми розпізнавання був застосований метод, заснований на штучних нейронних мережах (Artificial Neural Network - ANN). У наші дні більшість ASR систем в процесі розпізнавання використовують HMM.

З 90-х років розпізнавання мови кілька вдосконалювалося. Словник було розпізнати слів виріс до кількох десятків тисяч. Використання швидких алгоритмів декодування дозволило виробляти розпізнавання в реальному часі. У сучасних дикторозалежних системах, які розпізнають окремі слова, кількість яких досягає двадцяти тисяч слів, помилки складають менше 0.1% [22]. І близько 5% помилок в незалежних від диктора системах, які розпізнають злиту мова з тисячі слів [23].

Розпізнавання мови в реальному часі за допомогою сучасних методів вимагає великих обчислювальних ресурсів, обсяг яких часто буває обмежений. Неможливість широкого застосування багатьох алгоритмів сьогодні, наприклад, в мобільних пристроях, змушує дослідників шукати більш ефективні і оптимізовані методи. За рахунок своєї простоти і невеликої кількості операцій на кожній ітерації розглянутий в дипломній

роботі алгоритм може бути запропонований як альтернатива існуючим підходам для розпізнавання мови в реальному часі.

Найбільш яскравими прикладами, акустичних корпусів для англійської мови, які використовувалися в задачах розпізнавання мови, є такі. TIMIT - фонетично представницький мовної корпус злитої мови, призначений для розпізнавання злитого мовлення і проведення фонетичних досліджень [24].

Switchboard - акустичний корпус телефонного спонтанної мови, розроблений для розпізнавання телефонних діалогів [25]. TIDIGITS - велика мовна база, орієнтована для задач дикторо-незалежного розпізнавання послідовностей цифр Aurora2 - акустичний корпус, який містить зашумлені версії мовної бази [25].

1.2. Підходи до рішення задачі розпізнаванню мови

Підходи рішення задачі розпізнавання мови можна розбити на три класи [26]:

- акустико-фонетичні підходи;
- підходи розпізнавання образів;
- підходи штучного інтелекту.

Акустико-фонетичні підходи використовують властивості звуків мови (фонем) для подання і розпізнавання мовного сигналу. Всі фонеми можуть бути описані за допомогою відмінних характеристик, які і дозволяють розрізняти фонеми між собою. Прикладами таких характеристик для розрізнавання голосних звуків від приголосних є участь / відсутність голосу, наявність / відсутність формантной структури, «періодичність» і амплітуда сигналу. Голосні звуки між собою відрізняються формантною структурою. Приголосні можна розрізняти по тимчасовому сигналу, так і по спектральній структурі. Таким чином, маючи набір відмінних

характеристик, нескладно побудувати класифікатор у вигляді дерева рішень [26], який дозволить розмітити мовний сигнал по фонемам. Наступний етап - це визначення слів, що представляють послідовність фонем. Послідовність фонем може бути декодована шляхом прямого порівняння зі словами з словника. Тут можуть виникнути основні проблеми з декодуванням через те, що визначення і класифікація фонем - це процес неоднозначний з причини високої варіативності мови і складності точного вимірювання акустичних характеристик. Тому даних підхід вимагає глибокого розуміння звуків мови і способів їх опису, що не завжди неможливо.

Підходи розпізнавання образів полягають в знаходженні в мовному сигналі певних образів без явного їх виділення і сегментації. Всі підходи також включають два етапи: навчання та розпізнавання образів. На першому етапі системі надається набір мовних сигналів, з яких вона повинна виявити певні закономірності і навчитися образам, що адекватно описують мову. Далі, на етапі розпізнавання, система повинна зіставляє представлені їй мовні сигнали з навченими на першому етапі образами, і класифікує по ним невідому мову відповідно до деякої міри подібності.

Підходи штучного інтелекту використовують методи і алгоритми двох попередніх підходів. Системи штучного інтелекту намагаються приймати рішення аналогічно тому, як це роблять люди. Прикладом може бути експертна система, здатна, застосовуючи акустико-фонетичний підхід, сегментувати і розмічати мовної сигнал, далі, навчаючись і адаптуючи.

Базові ознаки акустичного сигналу - характеристики основного елемента периферичної слухової системи - равлики, і тонотопічна організація слухової системи [26] - припускають використання спектра Фур'є як основи для побудови ознак більш високого рівня. Під тонотопічною організацією розуміється поширення спектральних компонент акустичного сигналу в периферичній системі аж до слухової кори практично без перемішування. При цьому суміжні частоти поширюються в топологічно суміжних нейронних каналах. Для отримання

спектру використовують «віконний» аналіз з довжиною вікна в 15-25 мс з будь-яким згладжуючим вікном (Хеммінга, Ханнінг і ін.) [27] або рекурсивні фільтри з таким же часом загасання. Довжина вікна обумовлена характером зміни мовного сигналу або силабічного частотного мовлення, яке знаходиться в діапазоні 8-12 Гц. Вікна аналізу зазвичай зсуваються на 10 мс, забезпечуючи частотне отримання векторів-ознак в 100 Гц. У наборі популярних крейда-частотних кепстральних ознак (mel-frequency cepstral coefficients, MFCC) [28] над спектром виробляють маніпуляції, що імітують особливості обробки слухових систем: збирають компоненти спектра відповідно до частотної шкали крейда і логарифмують значення енергії в кожному каналі. Шкала крейда являє собою псевдологарифміну шкалу частот, експериментально отриману в психоакустичних експериментах. Її важливість полягає не тільки в тому, що вона відповідає нашим уявленням про роботу слухової системи, але і в тому, що, об'єднуючи спектральні компоненти високих частот у все більш широких зонах, вона дозволяє істотно знизити розмірність вектора ознак. Логарифмування моделює амплітудну компресію, характерну для поширення сигналів по нервових каналах. Додатково роблять протилежне косинусне перетворення, переходячи до кепстра, і залишають тільки перші 12 компонент. Косинусне перетворення призводить до декореляції спектральних компонент аналогічно перетворенню Карунена-Лоева [29]. Власне, характер вагових функцій Карунена- Лоева і наштовхнув на думку використовувати косинусне перетворення. Косинусне перетворення не має аналога в механізмах обробки сигналів нервовою системою.

Автоматичне розпізнавання мови є унікальним завданням моделювання системи, що розвинулася в процесі філогенезу за кілька сотень тисяч років. У цій системі «передавач» і «приймач» сигналів управляються одним органом - мозком, і протягом цих тисячоліть вони знайшли «спільну мову», яку і треба розшифрувати. Дуже сумнівно, що розшифрування допускає альтернативні варіанти, тим більше сумнівно, що

вони краще «натуральних». Очевидним наслідком цих міркувань є також те, що перспективні системи розпізнавання мови повинні в максимальному ступені використовувати досягнення фізіології в області слухового аналізу. Однак слід мати на увазі, що сліпе копіювання відкритих механізмів сприйняття може навіть погіршити розпізнавання, оскільки в живих системах механізми обробки рідко функціонують ізольовано один від одного. Скоріше мова може йти про загальні принципи обробки інформації в живих системах - багатоетапної ієрархічної обробки з використанням великої кількості нейронів і зворотними зв'язками. Відповідно до вищенаведених міркуваннями будемо оцінювати методи і досягнення в системах розпізнавання з точки зору їх «біологічності».

Нижче розглянемо три найбільш успішних методи, що використовуються в сучасних системах розпізнавання мови.

Перший підхід, який використовується для поліпшення показників розпізнавання мови, ґрунтується на виділенні векторів властивостей з сигналу з урахуванням особливостей сприйняття звуку людським вухом. Він включає в себе аналіз несучих частот і вирівнювання сигналу по гучності. Найбільш поширеними технологіями, які використовують такий підхід, є метод кепстральних коефіцієнтів тоновой частоти (Mel Frequency Cepstral Coefficients, MFCC, Davis & Mermelstein, 1980) і метод коефіцієнтів лінійного передбачення (Perceptual Linear Prediction, PLP, Hermansky, 1990). Одночасне і випереджальне зіставлення з шаблоном (маскування) (Paliwal & Lilly, 1997), характерне для людського сприйняття, може бути змодельоване і використано для виділення властивостей, що забезпечують більшу стійкість від шумів. З цією метою було створено метод варіювання розмірностей кадрів (Variable Frame Rate analysis, VFR, Zhu & Alwan, 2000). З огляду на специфіку роботи нервових клітин, що відповідають за слухові рецептори, був запропонований метод діапазонної автокореляції (Subband-Autocorrelation, SBCOR, Kajita & Itakura, 1994) [30].

Інший підхід заснований на аналізі звукових сигналів. Різниця надходять в систему зашумлених сигналів від шаблонів, отриманих в ході навчання «чистими» сигналами, є основною причиною нестійкості роботи систем розпізнавання. Метою підходу є зменшення цієї різниці. Передбачається, що шум у звукових сигналах адитивний і стаціонарний. Оцінки середнього значення усередненого шуму віднімаються з кепстра (Cepstral Mean Subtraction, CMS, Furui, 1981) або спектра (Spectral Subtraction, SS, Virag, 1999), обчисленого по зашумленими даними. Деякі модифікації таких методів включають в себе нелінійне спектральне віднімання (Non-linear Spectral Subtraction, NSS, Lockwood & Boudy, 1992), які використовують спектральні огинаючі. Такі техніки вимагають гарної оцінки шуму, яку на практиці буває складно отримати, особливо в разі нестаціонарного фонового шуму [31].

Ще одним способом боротьби з різницею між отриманими властивостями з зашумлених і чистих сигналів є використання високочастотного фільтра. Передбачається, що шум в сигналах не стаціонарний, а повільно змінюється в часі. Метод RASTA (Relative Spectral Analysis, Hermansky & Morgan, 1994) представлений таким чином, що відносні спектральні зміни фіксуються. І ті повільні зміни, які були викликані шумом, видаляються. У цьому випадку відпадає необхідність у явному оцінюванні шуму.

Третій підхід заснований на використанні багатовимірних просторів (Ephraim & Trees, 1994). Основною ідеєю цього підходу є знаходження лінійного відображення, яке мінімізує функцію вартості. Часто в якості такого відображення береться множення вектора властивостей на матрицю перетворення. Прикладами такого підходу можуть служити основний компонентний аналіз (Principal Component Analysis, PCA) і незалежний компонентний аналіз (Independent Component Analysis, ICA, Koscor, 2000), а також проектування на багатовимірні підпростори (Gales, 2002) [32].

Іншим підходом в задачі розпізнавання мови стали методи і алгоритми, засновані на порівнянні мовних сигналів із зразками [10]. Ідея полягає в наступному. Є набір еталонних сигналів, які можуть бути закодовані в тимчасовій або в частотній області, і які представляють словник для розпізнавання. Еталони можуть бути сформовані за допомогою усереднення однотипних сигналів та подання їх в деякому закодованому вигляді, наприклад кодової книги. Саме розпізнавання відбувається шляхом порівняння нового сигналу з усіма зразками і визначення найбільш підходящого кандидата відповідно до деякої метрики або мірою подібності.

Найбільш популярним серед таких підходів є алгоритм динамічного деформування часу (Dynamic Time Warping), або скорочено DTW-алгоритм, висхідний до Т.К. Вінцюка [14]. Алгоритм дозволяє ефективно виміряти схожість двох часових рядів на основі динамічного програмування (звідки і назва).

До середини XX століття, з розвитком ЕОМ стає можливим розпізнавати обмежений набір команд в практично реальному часі (комфортному для користувача). Кожна команда представлена одним або декількома зразками - наборами спектральних векторів. Кількість векторів в наборі для кожної команди і кожної її реалізації, взагалі кажучи, різниться і залежить від тривалості проголошення. Отриманий мовний сигнал 'X', який треба віднести до однієї з команд, представлений в такому ж вигляді. Таким чином, треба зіставити різні за змістом і тривалості набори еталонів з набором 'X' і вирішити, до якого ідеалу 'X' ближче. Основною трудністю в даному завданні була різна кількість векторів в порівнюваних наборах. Очевидний алгоритм, заснований на градієнтному методі, як і годиться «жадібному» алгоритму, давав дуже нестабільні результати навіть для одного і того ж диктора і був абсолютно непрацездатний навіть в невеликих шуми. Першим рішення, засноване на алгоритмі динамічного програмування, запропонував в 1968 р Т.К. Вінцюк [13]. Цей рік можна вважати кордоном, з якого стало можливим практичне застосування систем

розпізнавання мови. Незалежно, але дещо пізніше, цей же метод запропонували В.М. Величко та Н.Г. Загоруйко [14]. На Заході метод був запропонований також незалежно, але 10 років по тому [15]. Оскільки цей алгоритм не тільки зіграв надзвичайно важливу роль на початковому етапі розвитку систем розпізнавання мови, а й продовжує використовуватися в сучасних системах в іншій формі або під іншою назвою, а також, щоб його можна було судити з точки зору «біологічності», дамо його опис. Ідея методу проста і допускає розгляд на якісному рівні. Завдання полягає в тому, щоб порівняти дві сукупності векторів різної довжини, причому на просторі векторів є метрика або міра близькості. Уявімо, що ми порівнюємо еталон сам з собою: відкладемо вектори ознак еталона по осі X і Y. На площині XY на перетині координат, відповідних векторах з номерами i та j, побудуємо вертикальний відрізок (по осі Z), що дорівнює відстані (ступеня близькості) між цими векторами. Тоді на квадраті зі стороною, що дорівнює кількості векторів в ідеалі (N), виникне «гористий ландшафт», симетричний діагоналі (0,0) (N, N), однак по діагоналі пролягатиме абсолютно пряма «долина» з висотою, що дорівнює 0 (оскільки відстань від вектора до самого себе дорівнює 0). Якщо ми порівнюємо два різних еталона, що належать одному і тому ж слову, то «картина місцевості» спотвориться, однак, якщо використовувані ознаки адекватно відображають процес сприйняття, можна сподіватися, що деяка долина як і раніше буде пролягати по ламаній, близькою до діагоналі, тепер вже прямокутника N, M, де M - довжина другого зразка. Метод динамічного програмування дозволяє порахувати мінімальну суму висот або накопичену відстань, що набирається при русі з точки (0,0) в точку (N, M) і, якщо це потрібно для сегментації, відновити шлях, по якому ця відстань набрана. Отриману суму зазвичай нормують або на кількість пройдених вузлів, або на суму довжин слів або довжину більш короткого слова і розглядають як відстань між двома проголошеннями. Звичайно, використовувані в практичних системах реалізації мають безліч керованих параметрів, що оптимізують якість розпізнавання і зменшують

час рахунку. Розглянутий метод дозволяє в дікторозалежному варіанті розпізнавати 100-300 слів в ідеальних умовах з ймовірністю 90-98% [34].

Для додання системі дикторонезалежних якостей для кожного слова записують кілька еталонів від різних дикторів (в процесі навчання додають еталон від нового диктора, якщо він не був розпізнаний). Крім того, існують схеми нормалізації еталонів щодо дикторів, а також кластеризації дикторів. Цілком очевидно, що даний метод не має ніяких аналогій в роботі живих систем, більш того, він має ряд недоліків, які роблять його непридатним до розпізнавання великих словників, великої кількості нових дикторів і, звичайно, злитої промови. Перш за все, відзначимо довільний характер міри близькості в просторі векторів-ознак. В якості запобіжного близькості для спектральних векторів використовувалися квартально-блокова (сума модулів різниць компонент), евклидова, Махаланобіса. Для кепстральних коефіцієнтів використовувалася метрика Кульбака-Лейблера [16] або проекційна [17], для коефіцієнтів лінійного передбачення - метрика Ітакура-Саїто [18], що не сильно позначалися на якості розпізнавання. Оскільки загальний спектр мови спадає приблизно зі швидкістю 6 дБ / окт [19], внесок високих частот в відстань між векторами дуже малий у порівнянні з низькими. Для боротьби з цим явищем мовний сигнал диференціювали, хоча, з огляду на штучний характер методу, слід було б ввести множник для кожної спектральної компоненти і оптимізувати їх все за результатами тестів, що вже вимагало великих баз мовних даних, які в ті роки тільки починали формуватися. Яку б метрику і які б коефіцієнти не використовували, відносний внесок різних спектральних компонент в відстань залишається постійним і знову-таки довільним, в той час як слухова система виділяє з спектра потрібні компоненти, ігноруючи інші. Однак головним недоліком методу є його «ієрогліфічний» характер, тобто уявлення слів словника цілісними об'єктами без внутрішньої структури, що робить неможливим нарощування словника. Хоча нарощування словника більш 100-300 слів вже не мало сенсу через невисоку дискримінантну силу

методу (слова просто плуталися). Звернемо увагу, що під словами «розпізнавання методом динамічного програмування» по суті розуміють сукупність алгоритму динамічного програмування та подання мови за допомогою ланцюжка векторів-ознак без структурування слів, що може вводити в оману щодо цінності ідеї застосування алгоритму динамічного програмування. Як вже говорилося вище, алгоритм живе в більш сучасних системах розпізнавання, незважаючи на свою штучність, тільки замість мінімуму накопиченого відстані підраховується максимум накопиченого логарифма ймовірності [35].

Нехай $X = \{x_1, x_2, ..., x_N\}$ і $Y = \{y_1, y_2, ..., y_M\}$ - дві послідовності, що представляють дискретні (тимчасові) ряди або набір векторів. Позначимо відстань між компонентами (векторами) x_i і yj як $D_{ij} = d(x_i, y_j)$, яке може бути задано як середньоквадратична відстань чи інша метрика на безлічі векторів. Визначимо C_{ij} як відстань між послідовностями $X_i = \{x_1, x_2, ..., x_i\}$ і $Y_j = \{y_1, y_2, ..., y_j\}$ наступним рекурсивним чином:

$C_{11} = D_{11}$, $C_{i1} = D_{i1} + C_{i-11}$, $C_{1j} = D_{1j} + C_{1j-1}$,

$C_{ij} = D_{ij} + \min \{C_{i-1j}, C_{ij-1}, C_{i-1j-1}\}$, $1 \le i \le N$, $1 \le j \le M$.

Тоді очевидно, що величина CNM буде відстанню (у відповідно до даного визначення) між вихідними послідовностями X і Y. Однак, в загальному випадку дана відстань не буде метрикою в звичайному розумінні, тому що для неї не буде виконуватися нерівність трикутника.

Перевагою алгоритму є його здатності порівнювати мовні сигнали, які мають різні швидкості вимови дикторів. Крім того, DTW-алгоритм не вимагає великих обчислювальних ресурсів і пам'яті, що зробило його популярним у вбудованих системах і мобільних телефонах. Недоліком даного алгоритму, як і всього підходу на основі еталонів, є обчислювальна складність при наявності дуже великого словника близько тисячі слів [36].

Інший алгоритм – приховані Марківські моделі. Для створення систем розпізнавання мови з великим словником потрібно провести навчання на представницьких даних. У таких випадках часто застосовують статистичні

підходи машинного навчання, які здатні витягувати закономірності з невизначеної і неповної інформації. У мовних сигналах джерелами невизначеності та неповноти є варіації в звуках і дикторских вимовах, зовнішнього середовища і каналах зв'язку.

Найбільш успішними і популярними серед статистичних методів в задачах розпізнавання мови стали приховані Марковские моделі (Hidden Markov Models) через природну здатність описувати як тимчасові, так спектральні характеристики мови. Теоретичні основи прихованих Марковських моделей були дані в класичних роботах Баума (Baum) і його колег в кінці 1960-их і початку 1970-их років, практичне застосування в задачах розпізнавання мови було здійснено Бейкером (Baker) з CMU, і Желінеком (Jelinek) і його колегами з IBM в 1970-их роках [37].

У додатку до мовного сигналу ланцюг Маркова будується як односпрямований процес переходу між станами в дискретні моменти часу, при цьому ймовірність переходу в наступний стан залежить тільки від поточного стану і не залежить від того, в яких станах перебував процес в попередні моменти часу [22] (рис. 1). Ця вимога, однак, призводить до неправильних гістограми часу життя станів [23-26], і від нього в сучасних системах відмовилися. Моделі, в яких імовірність переходу в наступний стан залежить від часу перебування в поточному стані, називаються неоднорідними марківськими або напівмарківськими.

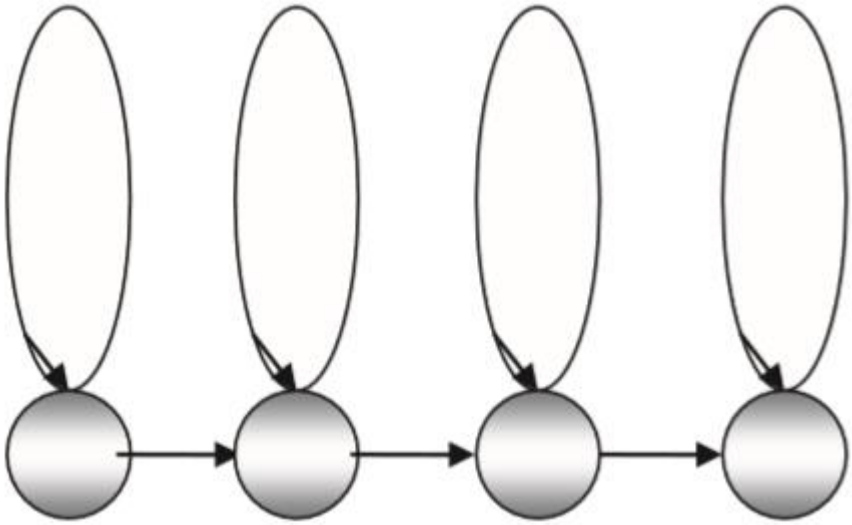


Рис. 1 Марківський ланцюг із чотирма станами

Таким чином, марківська модель деякого звуку або слова являє собою одне або декілька послідовних станів, для яких визначені функції щільності ймовірності в просторі ознак і ймовірності переходів. Функції щільності ймовірності можуть бути представлені в дискретному вигляді для проквантованого простору ознак або в безперервному вигляді. У другому випадку їх зазвичай апроксимують сумою гаусових функцій з діагональними матрицями коваріації. Діагональність матриць коваріації зменшує кількість навчених параметрів і спрощує деякі алгоритми, забезпечуючи аналітичні рішення деяких проблем, які для повних матриць коваріації допускають тільки чисельні рішення.

Наведемо на якісному рівні процес навчання, тобто метод отримання функцій щільності ймовірності станів і ймовірності переходу в наступний стан.

Для навчання використовують мовну базу даних (запис мовних сигналів і відповідних їм текстів), частина якої відсегментована (розмічена) досвідченими лінгвістами на одиниці або фрагменти, для яких ми збираємося будувати марківскі моделі (зазвичай це фонеми, не вдаючись у тонкощі визначення цього складного поняття). Після того, як речовий матеріал переведений в послідовність векторів-ознак, програма,

використовуючи границі, проставлені експертами-лінгвістами, збирає вектори ознак для кожної фонеми в окремі множини, для яких уже неважко побудувати функції щільності ймовірності, апроксимованим набором гаусових функцій. Також програма аналізує тривалості фрагментів і будує їх гістограми. Знаючи гістограми тривалості для кожної фонеми, неважко обчислити вірогідність виходу зі стану, що відповідає даній фонемі, після перебування в ньому якийсь час.

Отримавши перші оцінки параметрів станів (функцій щільності ймовірності і ймовірностей переходу), використовують алгоритм Баума-Велша [22, 27] або Вітербо [22, 27] для переоцінки параметрів з метою максимізації ймовірностей породження послідовностей векторів-ознак бази даних ланцюжками станів.

На наступному етапі використовують невідсегментовану частину мовної бази даних, яка залишилася. Справа в тому, що, хоча отримані стани ще недостатньо точні для розпізнавання мовлення, вони можуть дуже точно відсегментувати мовний матеріал, коли текст сказаного фрагмента відомий. Цей метод називається «примусовим вирівнюванням» (forced alignment), він дозволив використовувати дуже великі бази даних, а, як буде видно з подальшого, обсягу мовних баз завжди не вистачає.

Однак опис слів словника станами, відповідними фонем, не привело до істотного поліпшення якості розпізнавання в порівнянні з методом динамічного програмування.

Цьому є просте пояснення. Те, що ми представляємо як незмінну фонему, насправді являє собою ціле сімейство звуків, іноді, які сильно відрізняються за складом векторівознак. Адже мовний апарат не застигає в якихось положеннях, щоб зафіксувати чергову фонему, а безперервний рух мовотворчих органів породжує безперервну траєкторію в просторі ознак. Таким чином, сусідні фонеми впливають на проголошення даної фонеми. Цей ефект називається «коартикуляція». Інакше кажучи, наша функція щільності ймовірності складається з обрізків траєкторій в просторі ознак,

які перетинають її в різних напрямках. Функції щільності ймовірності для різних фонем істотно перетинаються в просторі ознак, що і викликає великі помилки.

Таким чином, для більш точного опису слід розглядати всі поєднання даної фонеми з попередніми і наступними звуками як окремі акустичні об'єкти, для яких потрібно будувати свої стани. Такі об'єкти називаються «Трифони», оскільки вони пов'язують три послідовних фонеми. Аналогічно визначаються «Біфони», які описують фонему в поєднанні з попередньою або наступною фонемами. Біфони використовуються при описі початку або кінця мовного фрагмента, а також, коли для побудови станів Трифона не вистачило даних. Фонеми без урахування контексту, які розглядалися досі, за аналогією називаються монофонами.

Розглянемо траєкторію в просторі ознак, що перетинає область функції щільності ймовірності даної фонеми - обривок траєкторії являє собою певний протяжний об'єкт, вектори ознак на початку і кінці якого визначаються попередніми та подальшими фонемами і можуть істотно відрізнятися один від одного. Описувати такий об'єкт одним станом не доцільно, оскільки це викличе додаткові помилки через перетини з іншими такими ж об'єктами. Зазвичай для опису трифона використовують три стани. Крайні стани описують ділянки сигналу, що знаходяться під впливом сусідніх фонем, а центральні - ту частину центральної фонеми, яка піддалася найменшому впливу сусідів. Однак кількість станів не повинно збігатися з глибиною контекстної залежності - можна розглядати пентафони [28, 29] і моделювати їх трьома станами або моделювати кількома станами монофонів.

Необхідність будувати стани для трифонів, тобто враховувати контекст, викликала нові труднощі – кількість фонетичних одиниць настільки зростає, що навіть дуже великих баз даних не вистачає для оцінки їх статистики. Наведемо дані з роботи [30], що відноситься до англійської мови і широко використовуваної базі даних Wall Street Journal Pronunciation

Lexicon. Для англійської мови кількість фонем становить близько 50 (кількість не є фіксованим - ряд поширених біфонів або трифонів можна заздалегідь віднести до окремих фонем). Тоді загальна кількість трифонів становить $50^3 = 125000$. Частина цих трифонів заборонена фонетичними правилами даної мови і ніколи не зустрічається, залишається 95221 трифонів. У згаданій базі даних, яка становить понад 57 годин мовлення і містить понад 36000 пропозицій, зустрічається тільки 22804 трифона, з них тільки 14545 трифонів, які зустрічаються більше 10 разів. Зрозуміло, що для навчання станів прихованої марківської моделі потрібна значна кількість зразків модельованого об'єкта. Число 10 можна визнати мінімально достатнім. Таким чином, понад 80000 трифонів є невидимими або незустрічаємими (unseen), але можуть зустрітися при експлуатації системи розпізнавання.

Кількість параметрів для однієї марківської моделі може досягати 1000-2000 (сюди входять матриці переходів і параметри гаусових функцій, апроксимуючих функції щільності ймовірності). Якщо помножити це число на кількість трифонів (50000-100000), то загальна кількість параметрів, яку треба оцінити в процесі навчання, виявляється порядку 10^8-10^9. Таким чином, постає нетривіальне завдання – оцінити мільйони параметрів, більшість з яких в навчальній базі даних не проявляється. Цю проблему вирішують шляхом зв'язування станів [30, 31]. Пов'язують або об'єднують ті стани функції, щільності ймовірності яких перекриваються найбільш сильно. Процес починають знизу, від монофнів, розщеплюючи монофони на трифона з найменш перекриваємими функціями щільності ймовірності, і закінчують, коли для навчання нових трифонів вже не вистачає даних. Таким чином, створюють тільки такі трифони, які розбивають функцію щільності даного монофона на великі, мало пересічні частини і які можна ефективно навчити.

Отримані системи розпізнавання вже значно перевищували системи, засновані на методі динамічного програмування. Однак для нового диктора

або іншого каналу передачі, якість розпізнавання суттєво падало. Необхідно було адаптувати якимось чином систему розпізнавання до нового диктора на основі дуже невеликого мовного матеріалу або в процесі роботи.

Вирішенню цих проблем було присвячено величезну кількість робіт, починаючи з дев'яностих років минулого століття.

Розрізняють нормалізацію ознак і адаптацію моделей.

Під нормалізацією ознак розуміють спотворення вхідного мовного сигналу або його векторів ознак з метою зближення з середнім характеристикам з векторами, складовими баз даних. З цією метою використовують віднімання середнього кепстра і нормалізацію по довжині голосового тракту [32, 33].

Під адаптацією розуміють зсув і спотворення моделей системи розпізнавання, тобто функцій щільності ймовірностей станів, щоб вони найкращим чином відповідали мовним даними нового диктора. Використовують Байєсову адаптацію або максимізацію апостеріорної ймовірності [34] і лінійну регресію максимуму правдоподібності [35-37].

З області розпізнавання дикторів і осіб прийшов метод адаптації за допомогою власних дикторів [38, 39]. Для адаптації моделей до шуму використовували векторні ряди Тейлора [9, 40].

Всі ці хитрощі дозволили підняти якість розпізнавання на досить високий рівень, так що системи розпізнавання стало можливим використовувати в системах голосового самообслуговування (IVR) та системах, призначених для кооперативного диктора.

Для розпізнавання злитого мовлення додатково використовуються моделі мови. Довільна модель мови дозволяє формально описати мову, а точніше, ті з її аспектів, які необхідні для підвищення якості автоматичного розпізнавання мови. Визначаючи можливу послідовність слів, ми піднімаємося на вищі рівні опису мови в порівнянні з фонетичним і, як наслідок, повинні враховувати системні відносини вищих порядків. Використовувана модель опису слова в реченні може бути складною, що

враховує синтаксичну та семантичну структуру висловлювання, а може бути дуже простою, яка вважає, що поява будь-яких слів рівноймовірна (в такому випадку ми, по суті, відмовляємося від лінгвістичного аналізу та обліку закономірностей і особливостей природної мови). Мовна модель дозволяє дізнатися, які послідовності слів в мові більш вірогідні, а якісь менш. На жаль, всі численні моделі мови для української мови дають найменший внесок в розпізнавання через досить вільний порядку слів у реченні і його синтетичного характеру, що виражається в численних словоформах, які до того ж погано розпізнаються через традиційне зниження гучності проголошення до кінців фраз. Повертаючись до оцінок методів з точки зору «біологічності», відзначимо абсолютну штучність методу. Він з усією необхідністю повинен був впертися в певні межі, що, власне, і зробив.

У сучасних системах розпізнавання мови широко застосовуються, так звані безперервних приховані Марківські моделі, функція розподілу ймовірності спостережень яких представляється у вигляді суміші нормальних розподілів (Gaussian Mixture Model).

Безперервна прихована Марківська модель з N станами {1, 2, ..., N} і M сумішами (для кожного стану) визначається трійкою $\lambda = \{A, B, \pi\}$, де

1) $A = \{a_{ij}\}$ - ймовірність розподілу переходів з одного стану в інший, тобто

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

2) B - ймовірність розподілу отримання вектора спостережень, тобто

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

де X - вектор спостережень, c_{jk} - вага к-ої суміші для стану j, і - функція нормального розподілу із середніми значеннями μ_{jk} і матрицею коваріацій $\sum_{jk}$. На практиці матриця коваріацій представляється у вигляді діагональної матриці з міркувань швидкості обчислень.

3) $\pi = \{\pi_i\}$ - ймовірність розподілу початкового стану моделі,

тобто

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

Виділяють три основні завдання, які вирішують в процесі застосування ПММ [38].

Завдання 1. Нехай дана послідовність спостережень $O = \{o_1, o_2, ..., o_T\}$ і модель $\lambda = \{A, B, \pi\}$, тоді потрібно ефективно обчислити умовну ймовірність $P(O \mid \lambda)$, тобто ймовірність появи цієї послідовності спостереження при заданій моделі.

Завдання 2. Нехай дана послідовність спостережень $O = \{o_1, o_2, ..., o_T\}$ і модель $\lambda = \{A, B, \pi\}$, потрібно визначити оптимальну в деякому сенсі послідовність станів $Q = \{q_1, q_2, ..., q_T\}$, найкращим чином відповідає даній послідовності спостереження.

Завдання 3. Необхідно знайти алгоритм, що дозволяє ефективно знаходити параметри моделі $\lambda = \{A, B, \pi\}$ так, щоб максимізувати ймовірність $P(O \mid \lambda)$.

Перше завдання є завданням оцінки, яка дозволить здійснити вибір оптимальної моделі серед кількох конкурентних моделей. Для її вирішення використовуються процедури прямого або зворотного підрахунків [1].

Друге завдання є завданням декодування, тобто знаходження оптимальної в деякому сенсі послідовності станів, яку могла б породити дана послідовність спостережень. Критерій оптимальності може бути довільним в залежності від розв'язуваної задачі. Завдання декодування зазвичай вирішується за допомогою алгоритму Вітербі або його модифікацій.

Третє завдання є завданням навчання моделі, тобто знаходження оптимальних параметрів моделі для максимізації ймовірності $P(O \mid \lambda)$. тут застосовують ітеративний метод Баум-Уелча (Baum-Welch) або метод максимізації очікування (Expectation-Maximization) .

Крім алгоритмічного вирішення цих трьох завдань необхідно вибрати архітектуру ПММ. Основними параметрами є кількість станів, зв'язок між ними, а також наявність зв'язкових станів і облік тривалості станів.

Переваги прихованих Марківських моделей полягає в їх адекватному моделюванні часових характеристик мови. До недоліків можна віднести складність розуміння процесу розпізнавання, що не дозволяє проаналізувати природу помилок, щоб поліпшити якість розпізнавання мови. Крім того, використання сумішей нормальних розподілів також має свої недоліки, які полягають у нездатності ефективно описувати дані, що лежать навколо деякого різноманіття. Зокрема, показано, що мова може бути змодельована деякою динамічною системою з невеликим числом параметрів, означаючи, що мовні характеристики лежать в набагато меншій розмірності, ніж тій, яка зазвичай використовується в сумішах нормальних розподілів.

Технології розпізнавання мови є досить молодими, але дуже перспективними у комерційному сенсі, що передбачає значне фінансування і бурхливе зростання. Однак, незважаючи на хороше фінансування, з часу, коли було запропоновано використовувати марківскі моделі (середина шістдесятих років XX століття), прогрес в якості розпізнавання на протязі близько 40 років був досить малим. Новим методам не вдавалося подолати результати, які були досягнуті безперервною марківською моделлю з гаусівською апроксимацією функцій щільності ймовірностей станів, або поліпшення було настільки незначним, що не варто істотного ускладнення систем. При цьому досягнуті результати не дозволяли використовувати системи розпізнавання мови як масовий комерційний продукт, хоча конкретні програми в вузьких предметних областях вже давно працювали.

Багатьом дослідникам представлялося, що характер завдання відповідає можливостям штучних нейронних мереж. Спроби використання нейронних мереж почалися досить давно. Як приклад можна навести статтю 1990 року [41], в якій було запропоновано багато перспективних ідей.

Зокрема, використовувалися довготривалі ознаки в вигляді одного супервектора, що складається із 9 послідовних векторів крейда-спектра, і рекурентний зв'язок між вихідним і вхідним шарами, що дозволяли враховувати контекстні залежності. Відзначимо, що довготривалі ознаки цілком «біологічно» описують фрагменти траєкторій в просторі ознак. Незважаючи на те, що в цій системі фактично використовувалися ті ж ознаки, що і в стандартній марківській моделі, плюс згадані удосконалення, перевершити стандартну систему на основі гаусових сумішей не вдалося. Цей факт викликав таке здивування в науковому середовищі, що в 1996 році вийшла стаття з промовистою назвою «Towards increasing speech recognition error rates» [42], в якій була зроблена спроба пояснити тривалу відсутність прогресу в створенні систем розпізнавання мови. Автори пояснювали відсутність прогресу тим, що марківська модель на основі гаусових сумішей була прийнята в якості базової в десятках наукових центрів у всьому світі і протягом декількох років була гранично оптимізована, так що будь-якій новій, сирій системі на початковому етапі перевершити її майже неможливо.

Незважаючи на те, що наведений аргумент важко заперечити, останні роботи, які використовують багатошарові нейронні мережі різних типів, доводять, що була ще одна, елементарна причина – нейронні мережі не мали достатньої інформаційної потужністю, оскільки потужність комп'ютерів не дозволяла використовувати мережі з декількома шарами і вихідним шаром, що складається із кількох тисяч нейронів, що відповідають трифонам (а не кільком десяткам монофонів, як в ранніх системах).

Нейронна мережа або перцептрон з будь-якою кількістю прихованих шарів є універсальним апроксиматором [42], тобто навіть мережі з одним прихованим шаром, що використовувалися до цього етапу, можуть апроксимувати будь-яку поверхню в просторі ознак. Однак успіх в розпізнаванні мови прийшов тільки з використанням багатошарових мереж. Це пояснюється неможливістю або крайніми труднощами створення

розумної методики ініціалізації ваг для мереж з одним прихованим шаром, що призводить до далекого від оптимуму набору ваг при навчанні.

Використання багатошарових нейронних мереж поставило нове завдання - розробку нових алгоритмів навчання, що, можливо, буде трендом робіт, пов'язаних з використанням нейронних мереж в майбутньому.

Одним з методів є ініціалізація за допомогою пошарового навчання, починаючи з нижніх шарів [41, 42]. У якості цільової функції для першого прихованого шару розглядається вхідний вектор ознак. Вихідний вектор може містити кілька послідовних MFCC або мелспектральних векторів-ознак. Щоб уникнути тотожного перетворення, вхідний вектор зашумлять. Наступний шар нейронної мережі навчають таким же чином відтворювати вихідні сигнали попереднього шару. Всього, таким чином, навчають до 5-7 шарів. Після того, як ініціалізація перших шарів проведена, включають стандартний алгоритм зворотного поширення помилки для всієї мережі з цільовою функцією, що відбиває приналежність вхідного сигналу до відповідного трифона. Даний підхід показав явну перевагу в порівнянні з класичним підходом з гаусовими сумішами – результати розпізнавання завжди виявлялися краще, причому багатошарова мережа, навчена на мовному матеріалі в 309 годин мовлення, показала кращі результати, ніж метод з гаусовими сумішами, навчений на 2000 годиннику мови.

Слід зазначити, що пропонований алгоритм навчання створює систему, що нагадує по функціоналу слухову. У слуховій системі виявлено нейрони, що реагують на певні події в акустичному сигналі [8]. У міру «поглиблення» сигналу в центральні відділи слухової системи характер ознак, що виділяються спеціалізованими нейронами, приймає все більш складний і виборчий характер. Попереднє навчання окремих шарів нейронної мережі виконує ту ж задачу – окремі шари навчаються знаходити ознаки сигналу все більш високого рівня.

Якщо внутрішні шари нейронних мереж виділяють ознаки мовного сигналу, характерні для мови взагалі, то їх можна уніфікувати для всіх мов,

навчаючи для кожної нової мови тільки вихідний шар нейронної мережі. Це було б надзвичайно важливо, оскільки для навчання тільки одного шару нейронної мережі була б потрібна набагато менша мовна база даних, ніж для навчання всіх 5-7 шарів.

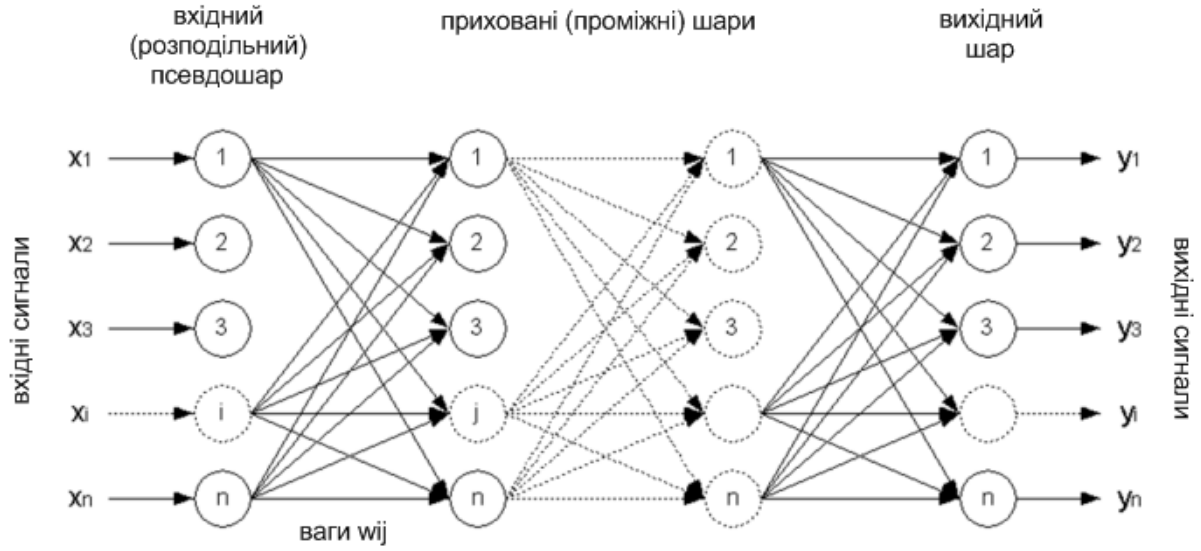


Рис. 2 Навчання нейронної мережі

Експерименти повністю підтвердили таку можливість. Використання спільно мовних баз даних для французького, німецького і італійської мов дозволило зменшити помилку розпізнавання на 3,3-5,4% в відносному вираженні в порівнянні з мономовними моделями [42].

Слід зазначити, що інформація про ознаки мовного сигналу, яка міститься у внутрішніх шарах нейромереж, може бути використана для розпізнавання мови на неспоріднених мовах. Були поставлені експерименти по використанню внутрішніх шарів нейронної мережі, навченої на базі даних європейських мов, для донавчання вихідного шару нейромережі для китайської мови. Відносний виграш склав від 21,1% до 8,3% при збільшенні бази даних китайської мови від 3 до 139 годин [40]. Розглянутий прийом відкриває можливість створювати системи розпізнавання для будь-яких мов, у тому числі малоресурсних.

Оскільки нейронні мережі не можуть ідентифікувати динамічні об'єкти, для порівняння моделей з сигналом як і раніше використовується формалізм марківських моделей, однак тепер в якості вектора ознак використовується набір апостеріорних ймовірностей трифоніва, отриманий на виході нейронної мережі. Такий метод використання нейронних мереж одним з перших запропонував для монофонів Х. Германський зі співавторами [41].

Цілком очевидно, що розробка багатошарових нейронних мереж з пошаровим навчанням – це найбільший крок в сторону біологічних механізмів обробки сигналів. По суті, залишається тільки один надзвичайно штучний елемент – це все той же алгоритм динамічного програмування в рамках марківських моделей, але під ім'ям Вітербо, оскільки апостеріорні ймовірності трифонів, отримані нейронними мережами, все ще «натягуються» на моделі з його допомогою.

Багато фахівців вважають, що ідентифікувати фонеми можливо за допомогою рекурентних нейронних мереж. Рекурентні нейронні мережі містять нейрони, об'єднані в спрямований круговий процес. Це наділяє нейронну мережу пам'яттю і, отже, здатністю розпізнавати процеси, а не тільки статичні об'єкти, як розглянуті вище глибокі нейронні мережі. Можна сподіватися, що такі мережі зможуть виносити рішення про наявність фонеми або іншого акустичного об'єкта, накопичуючи вхідну інформацію, що дозволить відмовитися від методу динамічного програмування і формалізму марківських моделей. Ідея використовувати рекурентні нейронні мережі почала досліджуватися досить давно [39, 41], але недостатня потужність комп'ютерів і тут не дозволила добитися переваги над домінантними в той час методами.

Ще однією перевагою рекурентних мереж може бути робота з векторами меншої розмірності. Контекстна залежність, тобто вплив фонем один на одного, в розглянутих багатошарових мережах моделюється побудовою вхідного вектора з декількох послідовних векторів-ознак, що

описують відрізок сигналу довжиною близько 25 мс. Вікна аналізу зміщуються на 10 мс. Для того щоб відобразити відрізок сигналу довжиною 300 мс (такі розміри контекстної залежності були виявлені в роботі [42]), потрібно близько 30 векторів-ознак, таким чином, розмірність результуючого супервектора може становити від 300 до 1000. Працювати з векторами такої розмірності незручно. Здається, ефективніше створити нейронну мережу з рекурентними зв'язками, яка буде зберігати інформацію про сигнал як рекурентний фільтр-інтегратор з витоком. Саме на основі таких інтегруючих нейронів з витоком побудовані резервуарні нейронні мережі [42]. Такі мережі містять шари, в яких нейрони пов'язані між собою, на відміну від «перцептронів», в яких зв’язки можливі тільки між нейронами різних шарів. У роботі досліджені двонаправлені нейронні мережі, що дозволяють врахувати попередній і наступний контекст щодо розглянутого фрагмента.

Штучна нейронна мережа - це математична модель біологічного нейрона, вперше змодельована Маккалоком (McCullough) і Питтсом (Pitts) в 1943 році, яка далі була застосована до завдань машинного навчання і розпізнавання образів Хеббом (Hebb) в 1949 році, Кларком в 1954 році (Clark) [Clark] і Розеблатом (Rosenblatt) в 1958 році . Інтерес до нейронних мереж на деякий час погас в результаті робіт Мінського (Minsky) і паперті (Papert) в 1969 році [35], в якій вони показали нездатність змоделювати функцію «виключає АБО» за допомогою нейронної мережі, а також неготовність обчислювальних машин того часу навчати велику нейронну мережу. Однак нова хвиля популярності нейронних мереж почалася в 2000-х роках з появою обчислювальних потужностей і концепцією глибокого навчання. Детальний огляд і теоретичну основу штучних нейронних мереж можна знайти в [15].

На сьогоднішній день найбільшу популярність в задачах розпізнавання мови отримали багатошарові нейронні мережі (Deep Neural Networks), які

покликані заповнити недоліки стандартних сумішей нормальних розподілів, описаних вище.

Багатошарові нейронні мережі були успішно застосовані для акустичного моделювання в Університеті Торонто, Microsoft Research, Google і IBM Research.

Багатошарова нейронна мережа - це односпрямована штучна нейронна мережа з одним і більше рівнями (шарами) з прихованими нейронами між вхідним і вихідним шарами. На вході нейронна мережа отримує акустичні параметри фіксованою розмірності, витягнуті з мовного сигналу. Кожен прихований нейрон j використовує логістичну функцію для відображення вхідного сигналу x_j з попереднього рівня в вихідний сигнал y_j

для наступного рівня:

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

де b_j - коефіцієнт відхилення нейрона j, i - індекс нейронів попереднього шару, і w_{ij} - вага зв'язку між i-им нейроном попереднього шару з даними j-им нейроном. При класифікації, вихідний нейрон j перетворює вхідний сигнал x_j в ймовірність класу p_j як:

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

де k - індекс, що пробігає по всіх класах.

Навчання нейронної мережі полягає в підпорі оптимального набору ваг і відбувається за допомогою алгоритму зворотного поширення помилки між істинним результатом і фактичним результатом, отриманим на виході нейронної мережі [15].

Багатошарова нейронна мережа з безліччю шарів і нейронів у кожному шарі вимагає дуже великих обчислювальних ресурсів, даних і часу для проведення повноцінного навчання. Методи градієнтного спуску, які використовуються при навчанні, можуть знайти лише локальні оптимуми, якщо не правильно задати початкові значення ваг, тим самим зводячи

мережу до перенавчання. Щоб уникнути такої ситуації і перенавчання мережі, був запропонований принципово новий підхід навчання мережі, званий генеративних попереднім навчанням, ідея якого полягає в наступному.

Кожен шар навчається окремо, причому вихідні дані попереднього шару, після його навчання, є вхідними даними для навчання наступного шару. Ваги, отримані в результаті такого навчання, є набагато кращими початковими умовами для проведення підсумкового дискримінативного навчання нейронної мережі, при якому дані ваги лише трохи зміняться.

Багатошарові нейронні мережі через здатність апроксимувати будь-яку (статичну) нелінійну функцію можуть повністю замінити суміші нормальних розподілів, але до сих пір не знайдені підходи повної заміни прихованих Марківських моделей, які показали перевагу в моделюванні часових (динамічних) залежностей.

Сьогоднішні системи автоматичного розпізнавання мови зробили крок вперед великими темпами за останні десятки років, починаючи з простих дикторо-залежних додатків до дикторо-незалежних систем автоматичної транскрипції новин, телефонних розмов, лекцій та ін. Незважаючи на повсюдне застосування таких систем, завдання розпізнавання мови далеко не вирішене в проблемах, пов'язаних з шумами, спотвореннями на лінії, іноземним акцентом, швидкістю і манерою мови та ін. Проте, дослідження по розпізнаванню мови активно ведуться в світлі останніх досягнень, і існує величезна кількість літератури на цю тему.

Рисунок 3 – Компоненти систем розпізнавання мови

На (Рис.3) 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 такі процедури як витяг і трансформація акустичних характеристик сигналу, адаптація до шумових ефектів або до варіацій між різними дикторами. Стандартним підходом до вилучення акустичних характеристик є обчислення вхідних векторів з кепстральних коефіцієнтів MFCC або коефіцієнтів перцептивних лінійних пророкувань PLP. Далі вектори трансформуються лінійною проекційною матрицею в простір меншої розмірності так, що фонетичні класи поділяються якомога більше. Зазвичай це здійснюється за допомогою лінійного дискримінаційного аналізу (LDA) або його розширення HLDA. Для забезпечення більшої стійкості від шумів застосовуються алгоритми SPLICE або QE. Обидва алгоритми були успішно протестовані на галасливих даних The Wall Street Journal (WSJ). Адаптація до різних дикторам проводиться шляхом нормалізації довжини голосового тракту до еталонного дикторові (VTLN), застосування лінійної регресії максимальної правдоподібності в просторі параметрів (fMLLR) або мінімальної фонетичної помилки в просторі параметрів (fMPE).

Акустична модель відображає акустичні характеристики звуків (фонем, діфонів, трифонів, та ін.) мови, для якої будується система розпізнавання мови. Найбільш популярними тут є підходи, засновані на прихованих Марківських моделях (ПММ) в зв'язці з сумішшю нормальних розподілів (GMM) або ж на багатошарових нейронних мережах (DNN).

Статистична мовна модель визначає ймовірність розподілу P (w_1, w_2, ..., w_n) появи послідовності з n слів w_1, w_2, ..., w_n. Дана величина приблизно оцінюється за допомогою n-грам, що обчислюються як частота їх повторенні в представницькому текстовому корпусі. Успішними методами побудови мовних моделей є мовна модель Kneyser-Ney , ієрархічна модель

Pitman-Yor , модель максимальної ентропії або, так звана, модель «М» . Дані методи в засновані на статистичному підході. Однак існують і інші методи, які використовують синтаксичну структуру мови. Мета декодера обчислити найбільш ймовірну послідовність слів W, маючи послідовність акустичних векторів X, використовуючи при цьому інформацію акустичної та мовної моделей. Класичним підходом пошуку гіпотез є алгоритм Вітербі, який ефективно обчислюється за допомогою динамічного програмування. Однак застосовуються також методи на базі зважених кінцевих трансдюсерів (WFST). Експерименти, які використовують дані моделі, досягли високих показників продуктивності. Наприклад, дикторо-незалежні системи розпізнавання телефонних розмов англійською мовою (English CTS, R04 Test Set) досягли рівня помилки слів порядку 15,2% [19].

Як приклади сучасних систем розпізнавання мови можна виділити системи з відкритим кодом HTK, CMU Sphinx, Kaldi, Julius, а також комерційні продукти Dragon NaturallySpeaking від Nuance, ViaVoice від IBM, Windows Speech Recognition від Microsoft, SIRI від Apple, Google Voice Search від Google [22].

Незважаючи на дуже значний прогрес в автоматичному розпізнаванні мови, досягнутому в останні 3-4 роки на тлі більш ніж тридцятирічного застою, можливості систем розпізнавання ще дуже обмежені в порівнянні з людиною. В основному переваги слухової системи визначаються надзвичайно потужними можливостями по адаптації і, головне, «кінцевим пристроєм» слухової системи, здатним розуміти сказане, завдяки яким людина не відчуває труднощів при розпізнаванні віддаленої, ревербованної, акцентної мови, мови в поганих каналах зв'язку, а також може виділяти мову одного диктора з багатоголосся і розпізнавати спонтанну мову. Всі ці завдання, а особливо останні два, представляють великі труднощі для сучасних систем розпізнавання. Автоматичні системи розпізнавання мови поки що перевершують людину тільки в задачах, де розуміння і модель мови не грають ролі, наприклад, при розпізнаванні ізольованих команд або чисел.

Розвиток систем розпізнавання мови буде пов'язано з удосконаленням структури нейронних мереж, обов'язковою наявністю зворотних зв'язків на різних рівнях і розробкою нових методів навчання таких нейронних мереж з використанням алгоритму динамічного програмування. Структура нейронних мереж повинна буде мати механізми адаптації і повернення для корекцій. Буде недивним, якщо в архітектурі нейронних мереж з'являться деякі елементи слухової системи, а методи навчання запозичать від навчання дитини, наприклад, пред'явлення на першому етапі найпростіших звуків – модульованих голосних і сполучень приголосна-голосна.

Висновок:

Перший розділ був присвячений огляду задачі розпізнаванню мови, способам її рішення. В цьому розділі класифіковано системи розпізнавання мови за різними критеріями, розглянули різні підходи до поставленої проблеми та проаналізували три найбільш успішних методи, які використовуються в сучасних системах розпізнавання мови. Серед заданих методів для даної роботи був вибраний метод динамічного програмування через швидкість та ефективність на невеликому наборі словника.

2. ОПИС ЗАСОБІВ РОЗРОБКИ ТА ПРОГРАМНИХ ЗАСОБІВ.

2.1. Опис засобів розробки

Пайтон (Python) — це потужна мова програмування, якою легко оволодіти. Вона має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис Пайтона, динамічна обробка типів, а також те, що це інтерпретована мова, роблять його ідеальним для написання скриптів та швидкої розробки прикладних програм у багатьох галузях на більшості платформ.

Інтерпретатор мови Пайтон і багата стандартна бібліотека (як код-джерело, так і бінарні дистрибутиви для усіх головних операційних систем) можуть бути отримані з сайту Пайтона, і можуть вільно розповсюджуватися. Цей самий сайт має дистрибутиви та посилання на численні модулі, програми, утиліти та додаткову документацію.

Інтерпретатор мови Пайтон може бути легко розширений функціями та типами даних, розробленими на C чи C++ (або на іншій мові, яку можна викликати із C). Пайтон також зручний як мова сценаріїв що вбудовуються в прикладні програми, для додаткових налаштувань функціональності.

Цей підручник повинен у загальних рисах ознайомити читача з головними концепціями та рисами Пайтона. Працюючи з цим посібником, загалом добре мати інтерпретатор мови Пайтон під рукою, але всі приклади самодостатні, отже цей текст може просто бути прочитаний.

Щодо опису стандартних об'єктів та модулів – Python Library Reference. Python Reference Manual дає більш формальне визначення мови. Щоб писати розширення на C та C++, читайте Extending and Embedding the Python Interpreter та Python/C API Reference. Існує також кілька книжок, що детально розглядають Пайтон.

Цей огляд не є всеохопним, у ньому не розглянуто кожну окрему рису чи навіть усі найбільш вживані особливості. Натомість, він містить риси

мови, які потребують першочергової уваги, та подає читачеві загальне уявлення про смак та стиль мови. Прочитавши його, ви зможете читати і створювати власні модулі та програми, а також будете готові ознайомитися з різноманітними модулями бібліотеки Пайтона, описаними у Python Library Reference.

Python це інтерпретована об'єктно-орієнтована мова програмування високого рівня. Вона є однією з тих рідкісних мов програмування, які є одночасно і простими, і потужними для проектів різного напрямку. Високорівневі структури даних, чистий синтаксис, динамічна семантика та ефективний підхід до об’єктно- орієнтованого програмування роблять її привабливою для написання скриптів, розробки прикладних програм та веб рішень.

Програмний код у Python зазвичай організовується у функції та класи, які можуть бути об’єднаними у модулі, котрі в свою чергу можуть бути об’єднаними у пакети.

Однією із синтаксичних особливостей мови є виділення блоків програмного коду за допомогою відступів (пробілів або табуляцій), що дозволяє забезпечення відсутності операторних дужок типу «begin-end» або «{-}», завдяки чому поведінка і коректність програм можуть бути залежними від початкових пробілів у тексті.

Вирази є повноправними операторами у мові програмування Python. Зміст, синтаксис асоціативність та пріоритет операцій є достатньо звичними для мов програмування в цілому і покликані зменшувати число використовуваних дужок [7].

Python пропонує механізм документування коду pydoc. Кожний модуль, клас, функцію чи метод бажано наповнювати стрічкою документації(docstring). В цьому випадку в інтерактивному режимі можна отримати допомогу по любому з них, згенерувати гіпертекстову документацію по цілому модулю чи навіть застосувати doctest для автоматичного тестування модуля.

Багата стандартна бібліотека є одною з привабливих сторін мови програмування Python. В ній наявні засоби для роботи з багатьма мережевми протоколами і форматами інтернета, наприклад модулі для написання HTTP-серверів і клієнтів, для розкладу і створення поштових повідомлень, для роботи з XML і т.д. Набір модулів для роботи з операційною системою дозволяє писати крос-платформинні застосування. Існують модулі для роботи з регулярними виразами, текстовими кодуваннями, мультимедійними форматами, криптографічними протоколами, архівами, серіалізації даних, підтримка юніт-тестування та інше.

Дизайн мови Python побудований навколо об’єктно-орієнтованої моделі програмування.

Можливості і особливості:

1. Класи є одночасно і об’єктами.
2. Множинне спадкування.
3. Поліморфізм. Усі функції є віртуальними.
4. Інкапсуляція. Поля можуть бути як загальнодоступними, так і прихованими.
5. Управління життєвим циклом об’єктів (конструктори, деструктори, розділювачі пам’яті).
6. Перевантаження операторів (крім «is», ‘.’, ‘=’ та символьних логічних).
7. Властивості. Поля можуть імітуватися за допомогою функцій.
8. Управління доступом до полів (емуляція полів та методів, частковий доступ тощо).
9. Методи для виконання найбільш розповсюджених операцій (істинне значення, метод «len()», глибоке копіювання, серіалізація, ітерація тощо).
10. Метапрограмування.
11. Інтроспекція.

12. Класові та статичні методи, класові поля.

13. Класи, що є вкладеними у функції та інші класи.

Програмне забезпечення, написане з використанням мови програмування Python, оформлюється у вигляді модулів, які в свою чергу можуть бути зібраними у пакети. Модулі можуть бути розміщеними як у каталогах, так і в ZIP-архівах. Модулі можуть бути двох типів за своїм походженням: написані з використанням самої мови Python, а також модулі розширення (від англ. extension modules), написані з використанням інших мов програмування.

Переваги та недоліки мови Python:

• Python — інтерпретована мова програмування. З одного боку це дозволяє значно спростити налагодження програм, з іншого це зумовлює порівняно низьку швидкість виконання.

• Динамічна типізація. У Python не треба заздалегідь оголошувати тип змінної, що дуже зручно при розробці.

• Чудова підтримка модульності. Python дозволяє розбивати програми на модулі, що сприяє повторному використанню коду в інших програмах.

• Вбудована підтримка Unicode в рядках. У Python необов'язково писати все англійською мовою, в програмах цілком може використовуватися ваша рідна мова.

• Підтримка об'єктно-орієнтованого програмування. При цьому його реалізація в Python є однією з найбільш зрозумілих.

• Автоматична збірка сміття, відсутність витоків пам'яті.

• Інтеграція з C та C++, якщо можливостей python недостатньо.

• Зрозумілий та лаконічний синтаксис, що сприяє ясному відображенню коду. Зручна система функцій дозволяє при грамотному підході створювати код, в якому буде легко розібратися іншій людині у разі потреби. Також ви зможете навчитися читати програми і модулі, написані іншими людьми.

• Величезна кількість модулів. У деяких випадках для написання програми достатньо лише знайти модулі, які підходять і правильно їх скомбінувати. Таким чином, ви можете думати про написання програми на більш високому рівні, працюючи з уже готовими елементами, що виконують різні дії.

• Різноманіття бібліотек підтримки. У складі Python наявна велика кількість зібраних і функціональних можливостей, відомих як стандартна бібліотека. Ця бібліотека надає масу можливостей , які потрібні в прикладних програмах, починаючи від пошуку тексту згідно шаблону і закінчуючи мережевими функціями. Python допускає розширення як за рахунок ваших власних бібліотек, так і за рахунок бібліотек, створених іншими розробниками.

• Швидкість розробки. У порівнянні з компілюючими, або строго типізованими мовами, такими як C, C ++ або Java, Python у багато разів перевищує продуктивність праці розробника. Обсяг програмного коду на мові Python зазвичай становить третину, або навіть п'яту частину еквівалентного програмного коду на мові C ++ або Java, що означає менший обсяг введення з клавіатури, менша кількість часу на відкладання і менший обсяг трудовитрат на супровід. Крім того, програми на мові Python запускаються відразу ж, минаючи тривалі етапи компіляції і зв'язування, необхідні в деяких інших мовах програмування, що ще більше збільшує продуктивність праці програміста.

• Кросплатформеність. Програма, написана на Python, функціонуватиме зовсім однаково незалежно від того, в якій операційній системі вона запущена, чи на Windows, чи на Linux. Відмінності виникають лише в небагатьох випадках, і їх легко заздалегідь передбачити завдяки наявності докладної документації.

Відмінностей Python від інших мов доволі багато, перерахуємо основні з них:

▪ Керування пам'яттю - цілком автоматичне — не потрібно хвилюватися щодо розподілу або звільнення пам'яті. Немає загрози “небезпечного посилання”. Java - єдина мова, що пропонує таку концепцію.

▪ Типи зв'язані з об'єктами, а не зі змінними. Це означає, що змінній може бути призначене значення будь-якого типу, і що (наприклад) масив може містити об'єкти різних типів. Традиційні мови не надають такої можливості.

▪ Операції звичайно виконуються в більш високому рівні абстракції. Це частково результат того, як написана мова, і частково результат розширеної стандартної бібліотеки кодів, що поставляється разом з Python.

2.2. Організація програмних засобів

Окрім стандартної бібліотеки існує більшість інших бібліотек, які представляються інтерфейс до всіх системних викликів на різних платформах. В першу чергу потрібно виокремити numpy(numeric python) і skipy (scientific python).

Бібліотека Numpy, яка призначена для роботи з багатомірними масивами дозволяє досягнути продуктивності наукових розрахунків, які порівнюються зі спеціалізованими пакетами. Масив – це контейнер, який містить в собі елементи одного типу(і одної довжини, якшо елементи вкладені масиви), організовані в упорядковану багатомірну матрицю. Доступ до елементів здійснюється по індексу – кортежу цілих чисел. Numpy являється основним пакетом для наукових обчислень в Python. Numpy – розширення мови програмування Python, яке додає підтримку великих багатомірних масивів та матриць, разом з великою бібліотекою багаторівневих функцій для роботи з цими масивами.

Зокрема використовувались такі функції np.arrange(), яка приймає в якості аргумента ціле додатнє число n, і повертає масив з n-1 елементів від 0 до n-1.

Також масив можна ініціалізувати за допомогою функції np.linspace(), вона приймає три аргументи – числа: початковий елемент масива, кількість елементів масива та кінцевий елемент масива. Повертає масив чисел рівномірно розподілених від початкового до кінцевого значення.

За допомогою np.array() задається повністю весь масив [8].

Для роботи з файлами формату .wav була використана бібліотека Sciру. SciPy - це відкрита бібліотека високоякісних наукових інструментів для мови програмування Python. SciPy містить модулі для оптимізації, інтегрування, спеціальних функцій, обробки сигналів, обробки зображень, генетичних алгоритмів, рішення звичайних диференціальних рівнянь, і інших завдань зазвичай вирішуються в науці і при інженерної розробці. Для роботи Sciру потрібно, щоб попередньо була встановлена бібліотека Numpy. А саме використані можливості scipy.io.wavefile для відкриття та зчитування інформації з аудірфайлу, а саме функції read(), яка повертає дискретизацію та частоту семлів.

Sciру використовує Numpy і надає доступ до обширного спектру математичних алгоритмів. Numpy та Sciру прості у використанні, але в одночас являються достатньо міцними.

Matplotlib – бібліотека для побудови графічних зображень, візуалізації розрахунків для мови програмування Python.

Бібліотека Matplotlib побудована на принципах ООП, але має процедурний інтерфейс pylab. Sciру використовує Matplotlib.

Matplotlib - бібліотека на мові програмування Python для візуалізації даних двовимірної (2D) графікою (3D графіка також підтримується). Matplotlib є гнучким, легко конфігурованим пакетом, який разом з NumPy, SciPy і IPython надає можливості, подібні MATLAB. В даний час пакет працює з декількома графічними бібліотеками.

Пакет (бібліотека) мови Python NumPy надає програмісту кошти для високоефективної роботи з величезними багатовимірними масивами даних. Як складова частина і основа, пакет NumPy входить в більшість проектів,

які використовують мову Python і вимагають більш-менш громіздких обчислень. Зокрема, з його використанням написані популярні пакети обчислювальної математики і наукової графіки SciPy і mathplotLib.

Модуль os надає безліч функцій для роботи з операційною системою, причому їх поведінку, як правило, не залежить від ОС, тому програми залишаються переносяться.

Стандартні засоби мови програмування Python дозволяють отримувати з програми доступ до об'єктів WWW як в простих випадках, так і в складних обставинах, зокрема при необхідності передавати дані форми, ідентифікації доступа через проксі і т.д. Варто відмітити, що при роботі з WWW використовується в основному протокол HTTP, але WWW охоплює не тільки HTTP, але і інші схеми. Використана схема зазвичай вказується у самому початку URL.

Urllib2 – модуль, який дозволяє працювати з URL – адресами. Модуль має свої функції та класи, які допомагають в роботі з URL. Urllib2 пропонує дуже простий інтерфейс, у вигляді функції urlopen(). Ця функція здатна витягнути URL-адресу за допомогою різних протоколів (HTTP, FTP,…). Вона приймає як аргумент адресу, щоб отримати доступ до видалених даних. Крім того urllib2 пропонує інтерфейс обробки розповсюджених ситуацій, таких як basic-аутентифікація, cookies, проксі-сервери і т.д. Модуль urllib2 має

І спеціальний клас для втілення запиту на відкриття URL. Називається цей клас urllib2.Request. Цей екземпляр містить в собі стан запиту. Можна задати початкові дані в Request, які потрібно відправити на сервер. Крім цього, можна передавати на сервер і додаткову інформацію(метадані) про дані, відправлених на сервер або в самому запиті, ця інформація передається у вигляді HTTP-заголовків.

Модуль Beautiful Soup - це парсер для синтаксичного розбіру HTML/XML, який може перетворити навіть не правильну розмітку в дерево синтаксичного розбору. Він підтримує і природні способи навігації пошуку

та модифікації дерева синтаксичного розбору. У більшості випадків він допомагає зекономити час і дні роботи.

Також для роботи з аудіо файлами формата .wav була використана бібліотекаа PyWavelets (pywt), яка призначеня для вейвлет-перетворень. Це модуль з математичними функціями, які дозволяють аналізувати різні частотні компоненти даних. Вейвлет як засіб багато масштабного аналіза дозволяє виділяти одночасно як основні характеристики сигналу, так і короткоживучі високочастотні складові в акустичному сигналі. Вейвлет-перетворення замінює перетворення Фур'є, так як перетворення Фур'є порівняно з вейвлет-перетвореннями має багато недоліків, у результаті яких відбувається втрата інформації про часові характеристики оброблюваних сигналів. Цей аналіз має на увазі використання частотно-часової локалізації, наприклад, вікон даних.

Висновки:

З вищесказаного можна зробити висновок, що мова програмування Python являється і простою і в той же час дуже потужно та універсальною. Вона підтримує декілька парадигм програмування, що добре для програм, які вимагають гнучкості. А наявність безлічі пакетів і модулів забезпечує універсальність та економить час. Вона має ефективні високо рівневі структури даний і простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис мови та динамічна типізація поєднані з командно-інтерпритованою природою та кросплатформеністю роблять Python ідеальною мовою для написання скриптів та швидкої розробки програм в багатьох сферах та на різних платформах.

3. ПОПЕРЕДНЯ ОБРОБКА МОВНИХ СИГНАЛІВ

3.1. Виділення признаків голосового сигналу та розпізнавання

Звуковий сигнал є одним із засобів взаємодії людини з навколишнім середовищем і людей між собою. Голос залежить від багатьох фізіологічних параметрів мовця і є по своїй суті індивідуальною характеристикою кожної людини. Проте, голос не є постійною характеристикою, він змінюється протягом життя людини, на нього також впливають стан здоров'я та емоції.

Розглянемо архітектуру сучасних систем розпізнавання мови. Будемо вважати, що вхідними даними є сам сигнал у форматі .wav або голос з мікрофона. Процес розпізнавання мови включає наступні етапи (рис. 4) :

- попередня обробка сигналу і параметризації;
- перетворення сигналу в вектори особливостей;
- розпізнавання мовної частини (класифікація).

Більшість сучасних систем автоматизованого розпізнавання використовують модульну архітектуру. Модуль попередньої обробки включає в себе(рис.5):

- цифрова фільтрація;
- сегментація мовного сигналу кадрами, що перекриваються;
- спектральне перетворення;
- обробка сигналу з використанням віконних функцій;
- спектральне перетворення.

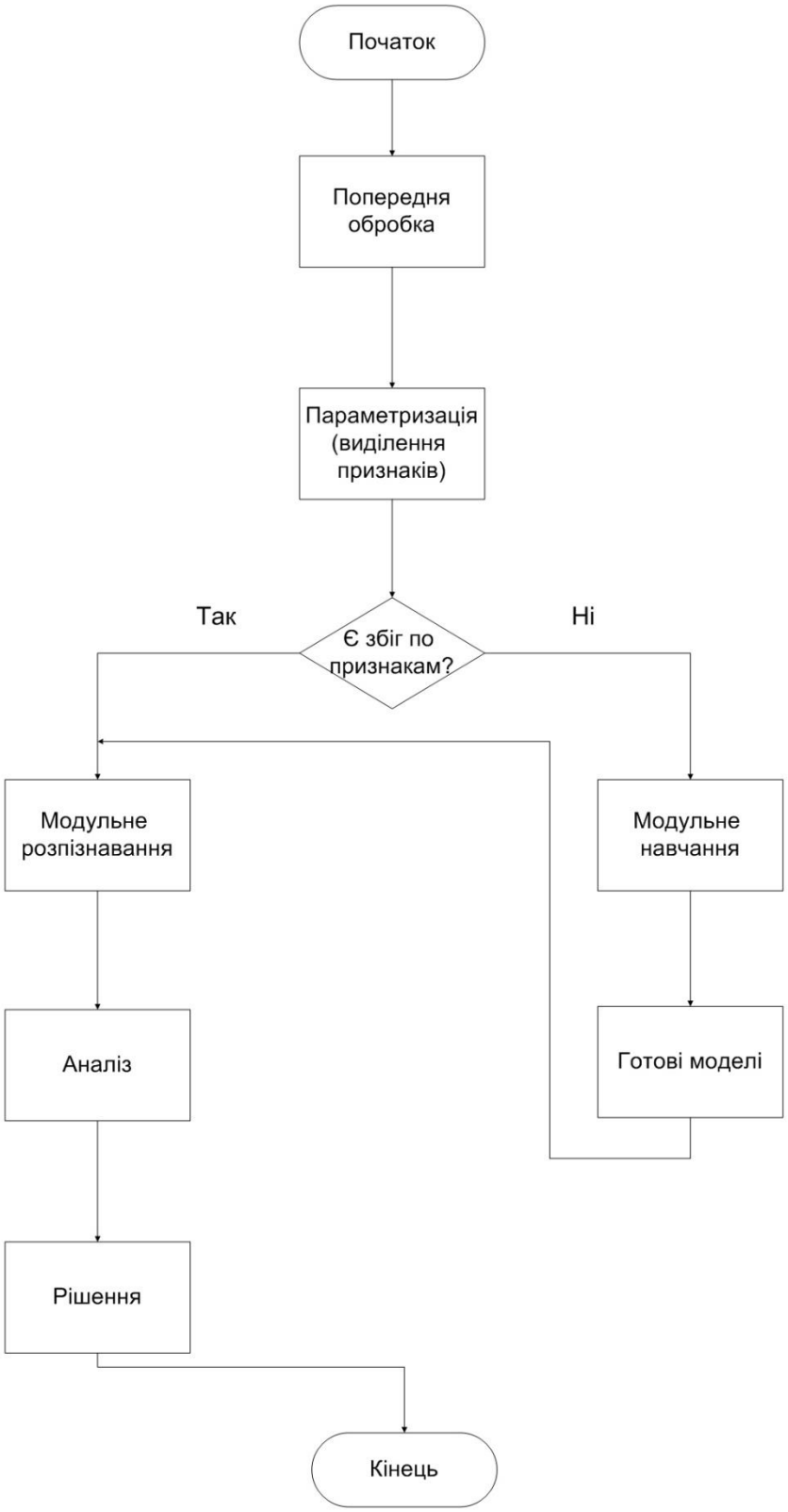


Рисунок 4 – Основні компоненти системи АРМ

Рисунок 5 – Етапи попередньої обробки мовного сигналу

Мовний сигнал після попередньої обробки перетворюється в набір векторів ознак, які містять спектральні характеристики: мел-частотні кепстральні коефіцієнти MFCC, і вони в свою чергу містять інформацію про сигнал, де ділянки, що містять мову, перетворюються в набори коефіцієнтів, які в подальшому надходять в блок розпізнавання (класифікації):

1) отримання спектра частот мовного сигналу за допомогою набору програмних смугових фільтрів (ДПФ);

2) перетворення отриманого спектра мовного сигналу:

a) логарифмічна зміна масштабу в просторі амплітуд і частот;

b) згладжування спектра з метою виділення його огинаючої;

c) кепстральних аналіз, т. е. зворотне перетворення Фур'є від логарифма прямого перетворення.

Сучасні системи АРМ грунтуються на статистичному моделюванні. Ці системи спочатку навчаються на багатогодинних колекціях мовних даних (процес навчання полягає в налаштуванні параметрів статистичних моделей). Потім, на етапі розпізнавання, системи виробляють зіставлення вхідних образів з образами, які вводились по навченим моделям. Іншими словами, вектори ознак, що відповідають невідомій мові, що розпізнається, порівнюються з моделями, в результаті чого обчислюються характеристики правдоподібності для кожної моделі. Такий підхід має істотний недолік - він ефективний, тільки якщо мовні характеристики навчальних образів і тестових близькі, чого на практиці домогтися складно. Проте, на сьогоднішній день статистичний метод прихованих марковських моделей (ПММ) є стандартним підходом до процесу модельного розпізнавання.

Останній етап є аналізом отриманої в попередніх етапах інформації з урахуванням граматики і словника (мовної моделі): видається рішення про те, яка модель підходить до невідомих векторів ознак. Мовна модель дозволяє отримати значення ймовірності появи якої-небудь послідовності слів незалежно від послідовності, що спостережується.

При вирішенні завдань обробки мовних сигналів важливу роль відіграє етап параметризації - процес виділення векторів ознак мовного сигналу. На даному етапі мовний сигнал після попередньої обробки (фільтрація, дискретизація, квантування, сегментація і ін.) перетворюється в вектори ознак, які містять необхідну інформацію про сигнал. Вектори ознак характеризують, властивості мовного сигналу, що змінюються в часі.

Більшість використовуваних векторів ознак тим чи іншим чином пов'язане зі спектральними або кореляційними характеристиками мовних сигналів. У дипломному проекті в якості векторів ознак використано коефіцієнти MFCC. Результатом попередньої обробки мовних сигналів є перетворення вихідної аналогової мови в цифровий вигляд, зручний для подальшої обробки. Принциповим припущенням, яке робиться в сучасних системах АРМ, є те, що мовний сигнал розглядається як стаціонарний (тобто його імовірнісні характеристики щодо постійні) на інтервалі в декілька десятків мілісекунд. Тому основною функцією попередньої обробки є сегментація - розбивка вхідного мовного сигналу на інтервали стаціонарності і отримання для кожного інтервалу спектральних оцінок.

Фільтрація представляється для зниження впливу локальних спотворень на характерні ознаки, які в подальшому будуть використовуватися для розпізнавання. Крім цього, попередній високочастотний фільтр використовується для видалення постійної складової, яка виникає в роботі пристроїв запису. Для цих цілей може бути застосований простий фільтр першого порядку зі значенням коефіцієнта 0,97:

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

де x[n] - вхідний сигнал до фільтрації, y[n] - вихідний сигнал після фільтрації. Сегментація мовного сигналу проводиться для того, щоб отримати вектори ознак однакової довжини: спочатку мовної сигнал сегментируется на рівні частини, після чого виконуються перетворення

всередині кожного кадру. Перекриття використовується для запобігання втрати інформації на кордонах сегментів.

Обробка сигналу з використанням віконних функцій призначена для зниження негативного впливу граничних ефектів (спотворення спектра), що виникають в результаті сегментації. Щоб мінімізувати спотворення спектра використовуються вікна w[n], що спадають до нуля на початку і кінці кожного сегмента.

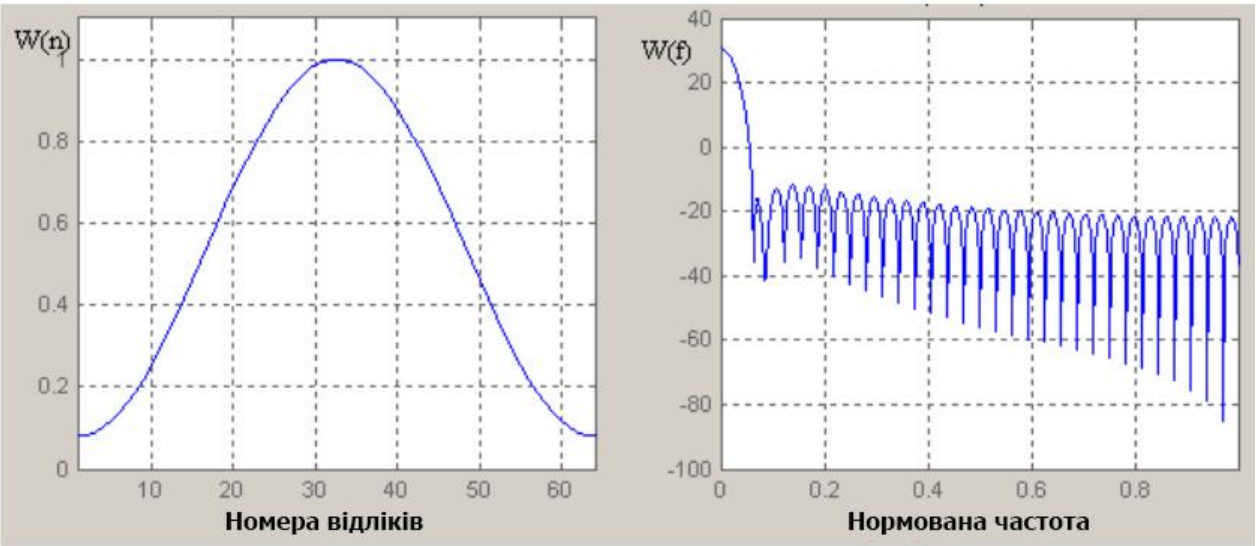


Рисунок 6 – Вікно Хеммінга та його спектр

В якості такого вікна зазвичай використовується вікно Хеммінга:

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

На рис. 6 показаний вид вікна Хеммінга і його спектр. Після застосування віконної функції використовується дискретне перетворення Фур'є (ДПФ) для отримання спектральних відліків сегментів.

$$X[k]=\sum_{n=0}^{N-1}x[n]w[n]\exp\left(-j\frac{2\pi kn}{N}\right),$$

де x[n] - відлік сигналу у часовій області, N - кількість відліків в одному сегменті, w[n] - віконна функція, X[k] - відлік сигналу в спектральної області. Отримані спектральні відліки піддаються наступного етапу

обробки, на якому відбувається обчислення різних типів векторів ознак: MFCC.

Сучасні засоби запису дозволяють уявити звуковий сигнал у вигляді тимчасового ряду, що показує зміну частоти в часі. Спектр сигналу, його уявлення в частотному просторі є більш інформативним для аналізу, ніж сигнал сам по собі. Для обчислення спектра часто використовується швидке перетворення Фур'є, алгоритм якого є досить простим для реалізації і має складність O(Nlog2N), меншу, ніж складність класичного алгоритму дискретного перетворення Фур'є O (N2).

Люди реагують на частотні зміни, тому при вирішенні задач, пов'язаних з аналізом людського голосу, часто використовують «кепстр» (cepstrum) - результат. Застосування перетворення Фур'є до спектру сигналу.

Також в процесі еволюції звуки в більш низькому частотному діапазоні містять більше корисної інформації, ніж знаходяться в більш високому частотному діапазоні. З урахуванням цих особливостей людського слуху були розроблені мел-частотні кепстральні коефіцієнти («мел» - скорочення англійського слова «melody» (мелодія)). За допомогою даних коефіцієнтів ретельніше аналізується інформація, що отримується з низькочастотного діапазону, а вплив високочастотних складових, зазвичай містять сторонній шум, на результат розпізнавання зменшується. Весь голосовий запис розділяється на невеликі інтервали тривалістю ~ 10-30 мс (час квазістаціонарності сигналу), звані кадрами. Для кожного фрейму окремо розраховується набір мел-частотних кепстральних коефіцієнтів, якій в подальшому будуть використовуватися для кластеризації. Алгоритм обчислення мел-частотних кепстральних коефіцієнтів можна розбити на наступні етапи:

- розбиття сигналу на фрейми;
- застосування вагової функції (вікна) до кожного кадру;
- застосування перетворення Фур'є;
- зикористання мел-частотного фільтра;

- обчислення кепстра.

Звуковий сигнал в загальному випадку не є стаціонарним, тобто його амплітуда і спектр змінюються в часі, що призводить до неможливості застосування багатьох технік аналізу. Але окремо взятий короткий інтервал порядку 10-30 мс можна вважати стаціонарним. Часто застосовують таку методику розподілу сигналу на фрейми: сигнал розділяється на інтервали довжиною N мс наступним чином: початок першого фрейму збігається з початком запису, другий фрейм починається через M мс інтервалів (M <N), відповідно, він на N-M мс перекриває перший фрейм.

Незважаючи на стаціонарність, таке уявлення сигналу не дозволяє використовувати перетворення Фур'є. Якщо частоти гармонік (частотних складових) сигналу не збігаються з базисними частотами перетворення Фур'є, в спектрі можуть виникати «зайві» гармоніки, які будуть лише «зашумлювати» отримане уявлення. Даний ефект носить назву «розмиття спектру» або «спектральна витік».

На Рис.5 показаний випадок для N = 20 мс і M = 16 мс.

Застосування вагової функції (вікна). Одним з можливих варіантів вирішення проблеми є застосування до сигналу вагової функції спеціального виду:

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

Результат застосування вагової функції до кожного кадру виглядає наступним чином:

$$y(n) = x(n) \cdot \omega(n), 0 \leq n \leq N-1$$

де x (n) - значення часового ряду в точці n; y (n) - зважене значення часового ряду в точці n (рис. 7а).

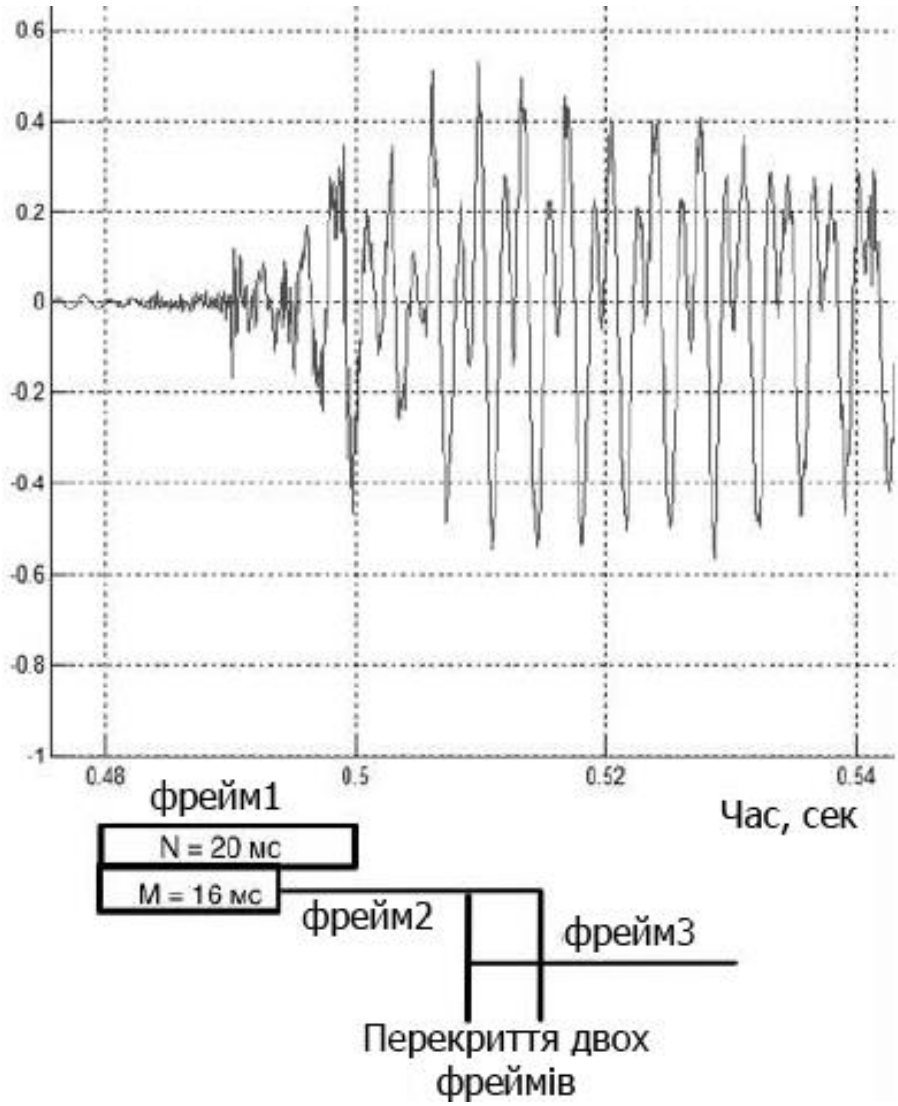


Рисунок 7 – Розбиття на фрейми мовного сигналу

Віддається перевага застосуванню «м'яких» вагових функцій, які зводять значення на кордонах фрейму до нуля. Ця операція називається «згладжуванням». Найбільш використованою є вагова функція Хеммінга, яку можна представити наступною формулою:

$$\omega(n) = 0.53836 - 0.46165 \cdot cos\left(\frac{2\pi n}{N-1}\right)$$

На наступному етапі необхідно застосувати перетворення Фур'є, яке переведе сигнал з тимчасового простору в частотний. На практиці найчастіше застосовується швидке перетворення Фур'є, що має такий вигляд:

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

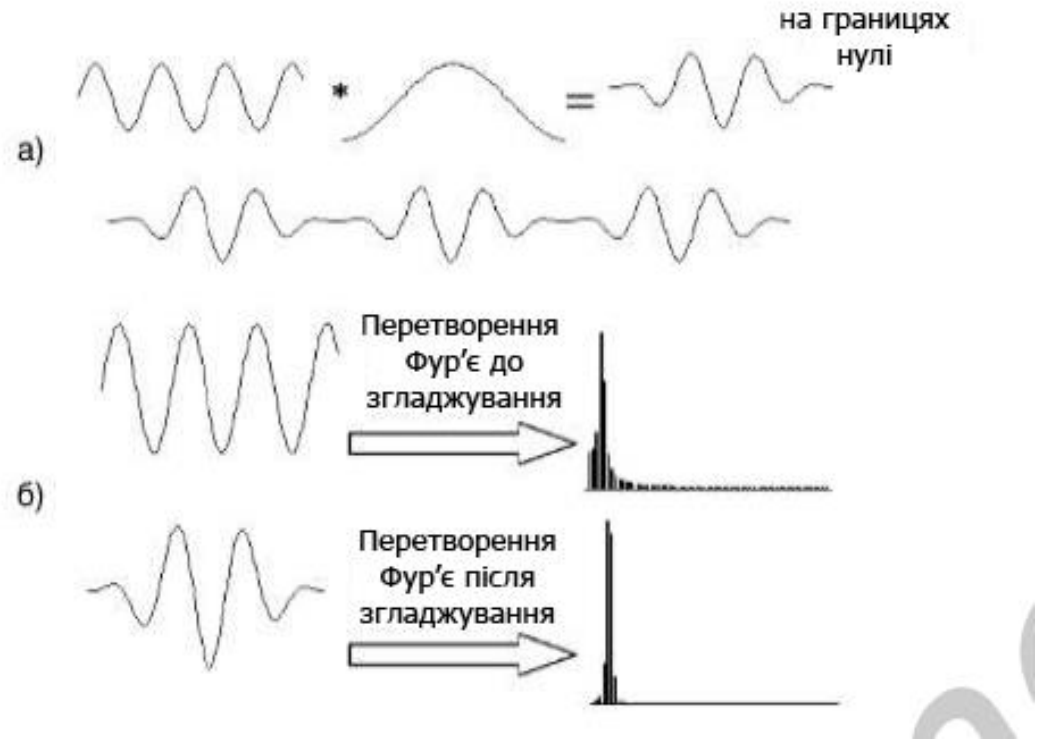


Рисунок 8 – Згладжування сигналу:

а) Застосування вагової функції до фрейму б) Застосування перетворення Фур'є

де yk - зважене значення часового ряду в точці k; Yn - комплексна амплітуда n-ої гармоніки сигналу, яку представляють тимчасовим рядом. Результатом даного етапу є спектр сигналу.

Щоб позбутися від негативних впливів шуму, сигнал обробляють спеціальними частотними фільтрами, про які йтиметься нижче. Частотний фільтр (band-pass filter) працює наступним чином: з усього набору гармонік, складових звуковий сигнал, фільтр залишає лише ті, частоти яких потрапляють в зазначену смугу пропускання.

Рисунок 9 – Два різних диктора говорять одну і ту ж фразу.

Розпізнавальна система є незалежною від диктора, якщо вона розпізнає слово незалежно від того, хто його вимовляє. На практиці реалізувати таку систему дуже складно з тієї причини, що звукові сигнали значно залежать від гучності, тембру голосу, стану і настрою диктора. На (рис. 9) зображено фонограми однієї і тієї ж фрази, яку він виголосив різними дикторами. Для вилучення інформації з таких сигналів нерідко використовують фільтри тонових частот, які усереднюють спектральні складові в певних діапазонах частот, тим самим роблячи сигнал менш залежним від диктора. Такі фільтри є основою технології MFCC (Mel-Frequency Cepstral Coefficients), яка використовується в розпізнає системі, що розглядається в цій роботі.

На даному етапі використання мел-частотного фільтра до спектру сигналу застосовується фільтр спеціального виду. Кожному значенню частоти, отриманого на попередньому кроці, ставиться у відповідність значення на мел-частотної шкалі. Значення цієї шкали для частот нижче 1 000 Гц точно відповідають спектру сигналу, отриманого при перетворенні Фур'є, частоти понад 1 000 Гц логарифмується. В результаті виходить модифікований енергетичний спектр сигналу mel (f) для кожної гармоніки частоти f, для обчислення якого використовується наступна наближена формула: $$mel(f) = 2595 \cdot lg\left(1 + \frac{f}{700}\right)$$

До даного спектру застосовується фільтр спеціального виду, що ставить у відповідність кожній частоті певний набір мел-коефіцієнтів $\tilde{S}_k$, 1, ..., K, де K – кількість мел-коефіцієнтів (на практиці часто вибирають значення от 12 до 24).

На попередньому кроці алгоритму отримані коефіцієнти $\tilde{S}_k$ необхідно перевести в мел-кепстальний простір.

Для цього зручно використовувати дискретне косинусне перетворення, яке описується наступною формулою [10, 12]:

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

де $\tilde{c}_n$ – отримані крейда-частотні кепстральних коефіцієнти.

Алгоритм акустичних векторів застосовується до кожного кадру, в результаті чого останньому відповідає набір мел-коеффіцієнтів, який використовується в більшості робіт як модель користувача для кластеризації і називається акустичним вектором.

Але зміна мел-коефіцієнтів також містить певну інформацію про користувача. Основна відмінність даної роботи від попередніх - розширення акустичного вектора шляхом обліку динаміки зміни крейда-коефіцієнтів δi, яка виражається різницею мел-частотних кепстральних коефіцієнтів даного фрейма і попереднього:

$$\delta_i(\tilde{C}_k[i]) = \tilde{C}_k[i-1] - \tilde{C}_k[i]$$

При цьому підході перший фрейм не може використовуватися для кластеризації, так як зміна мел-частотних кепстральних коефіцієнтів буде нульовою. А L - кількість елементів акустичного вектора x - збільшується вдвічі:

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

Часовий динамічний алгоритм (DTW) обчислює оптимальну послідовність трансформації (деформації) часу між двома часовими рядами.

Алгоритм обчислює обидва значення деформації між двома рядами й відстанню між ними [3].

Припустимо, що є дві числові послідовності $(a_1, a_2, \dots, a_n)$ та $(b_1, b_2, \dots, b_n)$. Як видно, довжина двох послідовностей може бути різною. Алгоритм починається з розрахунків локальних відхилень між елементами двох послідовностей, що використовують різні типи відхилень. Найпоширеніший спосіб для обчислення відхилень – метод, що розраховує абсолютне відхилення між значеннями двох елементів (евклідова відстань). У результаті отримаємо матрицю відхилень, що має n рядків і m стовпців загальних членів: $d_{ij} = |a_i - b_j|, i = \overline{1,n}, j = \overline{1,m}$ [4].

Мінімальна відстань у матриці між послідовностями визначається за допомогою алгоритму динамічного програмування та наступного критерію оптимізації:$a_{ij} = d_{ij} + min(a_{i-1,j-1}, a_{i-1,j}, a_{i,j-1})$, де: a_{ij} – мінімальна відстань між послідовностями $(a_1, a_2, \dots, a_n)$ та $(b_1, b_2, \dots, b_n)$. Шлях деформації – це мінімальна відстань у матриці між елементами a_{11} та a_{nm}, що складаються із тих a_{ij} елементів, які виражають відстань до a_{nm}.

Розрахунки проводилися для двох коротких послідовностей. Результати приведено в таблиці, у якій виділена послідовність 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 10 – Визначення відстані між двома послідовностями

Глобальні деформації складаються із двох послідовностей і визначаються по формулі: $GC = \frac{1}{p}\sum_{i=1}^{p} w_i$, де: w_i – елементи, які відносяться шляху деформації; p – їх кількість.

Існує три умови, що забезпечують роботу DTW відповідно до алгоритму забезпечення швидкої конвергенції [5]:

1. Монотонність – шлях ніколи не повертається назад і не повторюється, тобто індекс i та j, які використовуються, ніколи не зменшуються.

2. Безперервність – послідовність просувається поступово: за один крок індекси i та j збільшуються не більше ніж на 1.

3. Граничність – послідовність починається в лівому нижньому куті й закінчується в правому верхньому.

Оскільки для визначення основи послідовності в динамічному програмуванні оптимальним є використання методу зворотного програмування, необхідно використовувати певний динамічний тип структури, який називається «стек». Як і будь-який динамічний алгоритм програмування, DWT має поліноміальну складність. Коли ми маємо справу з більшими послідовностями, виникають дві незручності: запам’ятовування більших числових матриць; виконання великої кількості розрахунків відхилень.

У даному випадку використовується поліпшена версія алгоритму, Fastdwt, яка вирішує дві вищевказані проблеми. Розв’язок полягає в розбивці матриці станів на 2, 4, 8, 16 і т.д. менших по розміру матриць, за допомогою повторюваного процесу розбивки послідовності введення на дві частини. Т.ч., розрахунки відхилення проводяться тільки на цих невеликих матрицях, і шляхах деформації розраховується для невеликих матриць.

3.2. Особливості функціонування

У розробленому програмному забезпеченні для аналізу акустичних даних та розпізнаванню акустичної інформації використовується наступна модель роботи:

Рисунок 11 – Основна модель розпізнавання голосових команд

Функціонал програми:

- збереження команди в звуковий файл;
- додання команди до словнику для подальшого використання;
- виконання всіх системних команд, які вказані в словнику;
- порівняння з шаблоном;

- виконання команди зі звукового файлу.

Параметри, які необхідні для запуска програми:

- hmm <шлях до акустичної моделі>. Якщо ви скачували модель по наведеної вище посиланням, то акустична модель буде знаходиться в папці zero_ru_cont_8k_v3 / zero_ru.cd_cont_4000.
- dict <шлях до словника>
- jsgf <шлях до граматики>
- lm <шлях до мовної моделі>
- logfn <шлях до файлу для логгірованія>. За замовчуванням лог пишеться в stdout.
- infile <шлях до файлу з голосом>. Треба упевнитися що частота дискретизації файлу збігається з частотою дискретизації моделі.
- inmic <yes | no>. Звук в реальному часі з мікрофона.
- remove_noise <yes | no>. Фільтрація шумів. За замовчуванням yes.

Спочатку аудіо сигнал піддається первинній обробці та параметризації, які включають в себе цифрову фільтрацію, сегментацію мовного сигналу кадрами, що перекриваються, спектральне перетворення, обробку сигналу з використанням віконних функцій, а саме вікна Хеммінга, та зі спектрального перетворення. Після первинної обробки сигналу формуються набір векторів ознак, які містять в собі спектральні характеристики: мел-частотні кепстральні коефіцієнти MFCC, а вони в свою чергу містять інформацію на який ділянках сигналу присутня мова, та перетворюється в коефіцієнти, які в подальшому надходять в блок розпізнавання. Далі відбувається перетворення сигналу у вектори особливостей. На етапі розпізнавання відбувається зіставлення вхідних з образів з уже наявними шаблонами. Інши кажучи вектори ознак, порівнюються з моделями, в результаті чого обчислюється правдоподібність для кожної моделі. Важливим етапом є параметризація, так як в результаті чого і виконується

розпізнавання. Далі виконується розпізнавання за алгоритмом динамічного програмування.

Висновки:

У даному розділі було повністю описано алгоритми та методи аналізу акустичного сигналу(попередньої обробки, параметризації, виділення векторів ознак та розпізнавання). Описана теорія харківських ланцюгів і питання її застосування в задачах розпізнавання мови: приведені підходи до складанню прихованих харківських моделей для язикових об’єктів (слів), розглянуті процедури, за допомогою яких відбувається розпізнавання. Також було описані принципи та особливості розробленого програмного забезпечення.

4. ОЦІНКА ЯКОСТІ АНАЛІЗУ ТА РОЗПІЗНАВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

Для тестування розробленого програмного забезпечення було обрано такі критерії: відстань, кут, шум.

1). За нормальної відстані від комп’ютера (0.3-1м). Кут = 0 градусів. Без шуму.

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

2). За нормальної відстані від комп’ютера (0.3-1м). Кут = 45 градусів. Без шуму.

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

3). За нормальної відстані від комп’ютера (0.3-1м). Кут = 45 градусів. Без шуму.

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

4). За відстані 5 метрів від комп’ютера. Кут = 0 градусів. Без шуму.

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

5). За нормальної відстані від комп’ютера (0.3-1м). Кут = 0 градусів. Шум = 100% (на фоні відбувається шум з такою ж гучністю, з якою дає команду користувач).

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

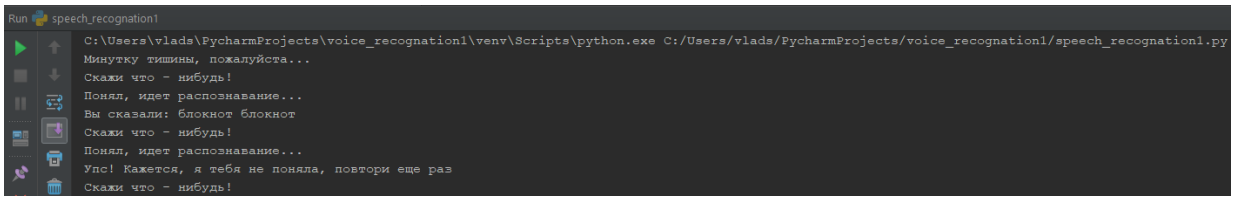
6) За відстані 5 метрів від комп’ютера. Кут = 90 градусів. Шуму = 100%.

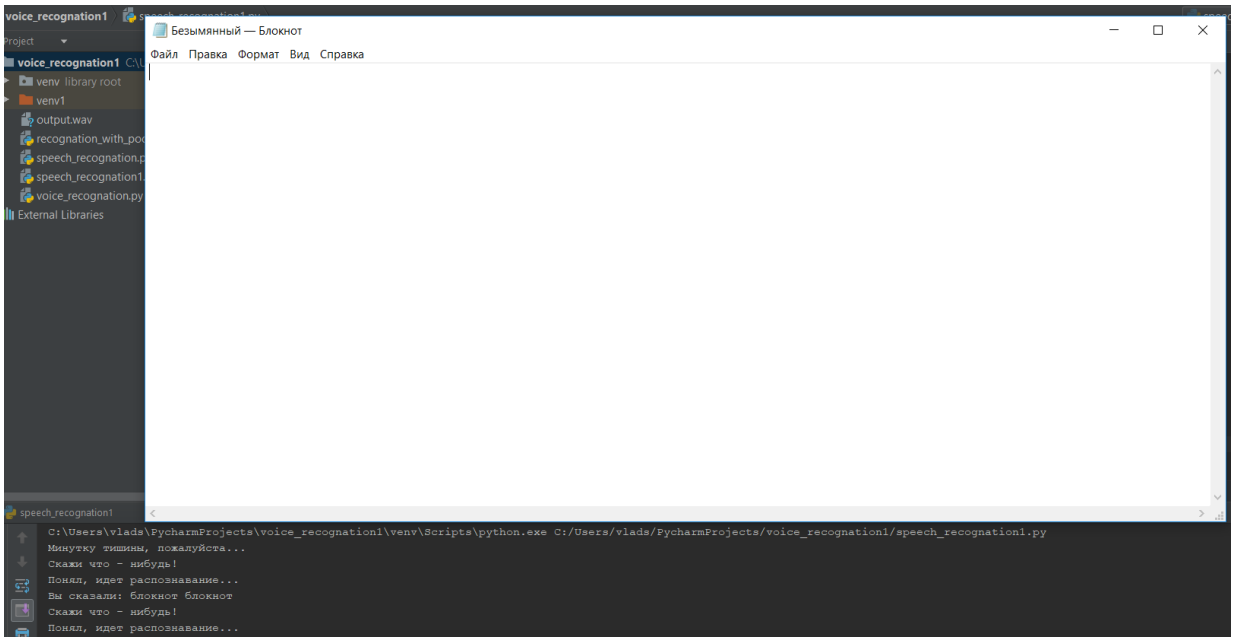
Команда	Відстань	Кут	Шум	Коефіцієнт розпізнавання
«Блокнот»	5м	90°	100%	49%
«Ворд»	5м	90°	100%	32%
«Ком’ютер»	5м	90°	100%	45%
«Пейнт»	5м	90°	100%	43%

Можемо зробити висновок, що за нормальної відстані від мікрофона, кут майже не впливає на якість та точність розпізнавання команд; відстань – дещо впливає на якість, проте напряму залежить від якості мікрофона та

його фізичних характеристик. Шум – це найбільш впливовий фактор для розпізнавання мовлення.

Демонстрація роботи програми:





ВИСНОВОК

У результаті виконання дипломного проекту було проаналізовано процеси аналізу акустичних даних та розпізнавання акустичної інформації та розроблено програмне рішення поставленого питання.

Проведено аналіз задачі представлення і розпізнавання акустичної інформації, виділені основні компоненти систем автоматичного розпізнавання акустичної інформації (команди):

- попередня обробка акустичного сигнала;
- перетворення сигнала в вектори ознак;
- розпізнавання акустичної інформації(класифікація).

Були розроблені та розглянуті методи попередньої обробки і виділення ознак мовного сигнала, серед яких був вибраний один з найпопулярніший та найкорисніших підходів, який заснований на знаходженні мел-кепстральних коефіцієнтів(MFCC).

Розглянуті методи розпізнавання акустичної інформації та вибраний метод динамічного програмування. В описі даних методів приведена їх коротка характеристика, класифікація, яку задачу вони рішають(попередню обробку та розпізнавання), алгоритми побудови систем розпізнавання на основі цих методів, а також їх застосування.

Таким чином, ефективна система розпізнавання має містити в собі такі етапи обробки вхідного сигналу, параметризацію та розпізнавання.

Був розроблений програмний комплекс, який дозволяє створювати бази голосових команд та який включає в себе реалізовані всі вище вказані методи та алгоритми, які беруть участь в аналізі та розпізнаванні акустичної інформації.

Розглянута модель розпізнавання голосових команд призначена для створення мовного інтерфейсу, який дозволить істотно підвищити ефективність роботи людино–машинних систем. Ця модель базується на використанні частотного аналізу мовного сигналу, зокрема, перетворення

Фур’є, що забезпечить високу швидкодію програмних засобів. Класифікація мовних команд виконується на основі часового динамічного алгоритму, який опрацьовує кожний із наборів однієї команди та надає середнє значення схожості, що дозволяє отримати більший коефіцієнт точності, на відміну від, інших систем розпізнавання команд.

Були проведені експериментальні дослідження з розпізнавання мови та різних технічних звуків.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1.Rabiner L.R. Juang B., Fundamentals of Speech Recognition // Pearson Education. -1993. -507p.

2. Soong, F.,A. Rosenberg, L. Rabiner, B.Juang. A vector Quantization approach to Speaker Recognition. IEEE Proceedings International Conference on Acoustics, Speech and Signal Processing ICASSP. – 1985. –Vol. 1. –P 387-390.

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

4. Bhatnagar A.C. Analysis of Hamming window using advance peak window method // Interatinal Journal of Scientific Research ngineering&Technolongy Vol.1 Issue 4, 2012,C.15-20.

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

6. А. Оппенгейм, Р. Шафер, Цифровая обработка сигналов, М.: Техносфера, 2006. – 856 с.

7. Trentin E., Gori M., A survey of hybrid ANN/HMM models for automatic speech recognition // Neurocomputing, 2001. Vol. 37, No. 1-4. – Pp. 91-126.

8. Patel I., Rao Y.S., Speech recognition using hidden markov model with MFCC-subband technique // International Conference on Recent Trends in Information, Telecommunication and Computing, 2010. – Pp. 168-172.

9. S. Kumar, M. Rao, Design Of An Automatic Speaker Recognition System Using MFCC, Vector Quantization And LBG Algorithm // International Journal on Computer Science and Engineering, 2011. Vol. 3, No. 8. – Pp. 2942-2954.

10. Molau S., Pitz M., Schlüter R., Ney H. Computing mel-frequency cepstral coefficients on the power spectrum. IEEE Intern. Conf. on Accoustics, Speech, and Signal Processing, 2001, vol. 1, pp. 73–76.

10. Рабинер Л. Скрытые марковские модели и их применение в избранных приложениях при распознавании речи: Обзор // Труды

института инженеров по электротехнике и радиоэлектронике, т. 77, № 2. – М.: Мир, 1989. – с. 86-120.

11. Л. Рабинер, Р. Шафер, Цифровая обработка речевых сигналов. – М.: Радио и связь, 1981. – 496 с.

12. Фролов, А. Синтез и распознавание речи. Современные решения [Электронный ресурс] / А. Фролов, Гр. Фролов. – Электрон. журн. – 2003. – Режим доступа : http://www.frolov-lib.ru.

13. M. Bahoura, H. Ezzaidi, Hardware implementation of MFCC feature extraction for respiratory sounds analysis // 8th Workshop on Systems, Signal Processing and their Applications, 2013. – Pp. 226-229.

14. M. Bahoura, H. Ezzaidi, Hardware implementation of MFCC feature extraction for respiratory sounds analysis // 8th Workshop on Systems, Signal Processing and their Applications, 2013. – Pp. 226-229.

15. Хайкин. Нейронные сети: полный курс, 2-е изд., испр.: Пер. с англ./Саймон хайкин. - М.: ООО "І.Д.Вільямс", 2006. - 1104 с.: ил. - Парал. тит . англ

16. Levin K., Ponomareva I., Bulusheva A., Chernykh G., Medennikov I., Merkin N., Prudnikov A., Tomashenko N. Automated closed captioning for Russian live broadcasting. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. Singapore, 2014, pp. 1438–1442.

17. Terry K. Instant patient records and all you have to do is talk. Medical Economics, 1999, vol. 76, no. 19, pp. 101–102, 107–108, 111–112.

18. Zafar A., Overhage J.M., McDonald C.J. Continuous speech recognition for clinicians. Journal of the American Medical Informatics Association, 1999, vol. 6, no. 3, pp. 195–204.

19. Goedart J. Speech recognition technology gives voice to clinical data. Health Data Management, 2002, vol. 10, no. 12, pp. 30–32, 34, 36.

20. Zick R.G., Olsen J. Voice recognition software versus a traditional transcription service for physician charting in the ED. American Journal of Emergency Medicine, 2001, vol. 19, no. 4, pp. 295–298.

21. Apple - iOS 8 - Siri. Available at: http://www.apple.com/ru/ios/siri (accessed 10.10.2015).

22. Voco: Windows application for translation speech to text. Available at: http://www.speechpro.ru/product/transcription/voco (accessed 10.10.2015).

23. Chistovich L.A., Ventsov A.V., Granstrem M.P. et. al. Rukovodstvo po Fiziologii. Fiziologiya Rechi. Vospriyatie Rechi Chelovekom [Guidance on Physiology. Physiology of Speech. The Perception of Human Speech]. Leningrad, Nauka Publ., 1976, 388 p.

24. Huang X., Acero A., Hon H.-W. Spoken Language Processing. Prentice Hall, 2001, 1008 p.

25. The HTK book. Cambridge University Engineering Department. Available at: http://speech.ee.ntu.edu.tw/homework/DSP_HW2-1/htkbook.pdf (accessed 22.10.2015).

26. Tou J.T., Gonzalez R.C. Pattern Recognition Principles. 2nd ed. Addison-Wesley, 1977, 377 p.

27. Hermansky H. Should recognizers have ears? Speech Communication, 1998, vol. 25, no. 1–3, pp. 3–27.

28. Vintsyuk T.K. Raspoznavanie slov ustnoi rechi metodami dinamicheskogo programmirovaniya [Oral speech recognition using dynamic programming]. Kibernetika, 1968, no. 1, pp. 81–88.

29. Velichko V.M., Zagoruiko N.G. Avtomaticheskoe raspoznavanie ogranichennogo nabora ustnykh komand [Automatic recognition of a limited set of verbal commands]. Vychislitel'nye Sistemy, 1969, no. 36, pp. 101– 110.

30. Sakoe H., Chiba S. Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1978, vol. 64, pp. 43–49. doi: 10.1109/TASSP.1978.1163055

31. Kullback S. Letter to the Editor: The Kullback-Leibler distance. The American Statistician, 1987, vol. 41, no. 4, pp. 340–341.

32. Mansour D., Juang B.H. A family of distortion measures based upon projection operation for robust speech recognition. IEEE Transactions on Acoustics, Speech and Signal Processing, 1989, vol. 37, no. 11, pp. 1659–1671. doi: 10.1109/29.46548

33. Itakura F., Saito S. Analysis synthesis telephony based on the maximum likelihood method. Proc. 6th Int. Congress on Acoustics. Los Alamitos, 1968, pp. 17–20.

34. Flanagan J.L. Speech Analysis, Synthesis and Perception. Springer, 1965. doi: 10.1007/978-3-662-00849-2 20. Baker J.K. The dragon system – an overview. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1975, vol. ASSP 23, no. 1, pp. 24–29.

35. Jelinek F. Continuous speech recognition by statistical methods. Proc. of IEEE, 1976, vol. 64, no. 4, pp. 532–556. doi: 10.1109/PROC.1976.10159

36. Rabiner L.R. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 1989, vol. 77, no. 2, pp. 257–286. doi: 10.1109/5.18626

37. Ramesh P., Wilpon J.G. Modeling state durations in hidden Markov models for automatic speech recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, ICASSP-92. San Francisco, USA, 1992, vol. 1, pp. 381–384.

38. Bonafonte A., Ros X., Marifio J.B. An efficient algorithm to find the best state sequence in HSMM. Proc. 3rd European Conf. on Speech, Communication and Technology, EUROSPEECH’93. Berlin, Germany, 1993, pp. 1547–1550.

39. Pylkkönen J. Phone Duration Modeling Techniques in Continuous Speech Recognition. Master’s Thesis. Helsinki University of Technology, 2004. Available at: http://users.ics.aalto.fi/jpylkkon/mt.pdf (accessed 18.10.2015).

40. Introduction to Automatic Speech Recognition. MIT, 2003. Available at: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-345-

automatic-speechrecognition-spring-2003/lecture-notes/lecture1.pdf (accessed 23.10.2015).

41. Sakti S., Markov K., Nakamura S. Incorporation of pentaphone-context dependency based on hybrid HMM/BN acoustic modeling framework. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP. Toulouse, France, 1996, vol. 1, pp. I1177–I1180.

42. Shafran I., Ostendorf M. Use of higher level linguistic structure in acoustic modeling for speech recognition. Proc. IEEE Int. Conf. on Acoustic Signal and Speech Processing. Istanbul, Turkey, 2000, vol. 2, pp. 1021– 1024.

Додаток 1. Копії графічного матеріалу

Рисунок 1 – Етапи попередньої обробки мовного сигналу

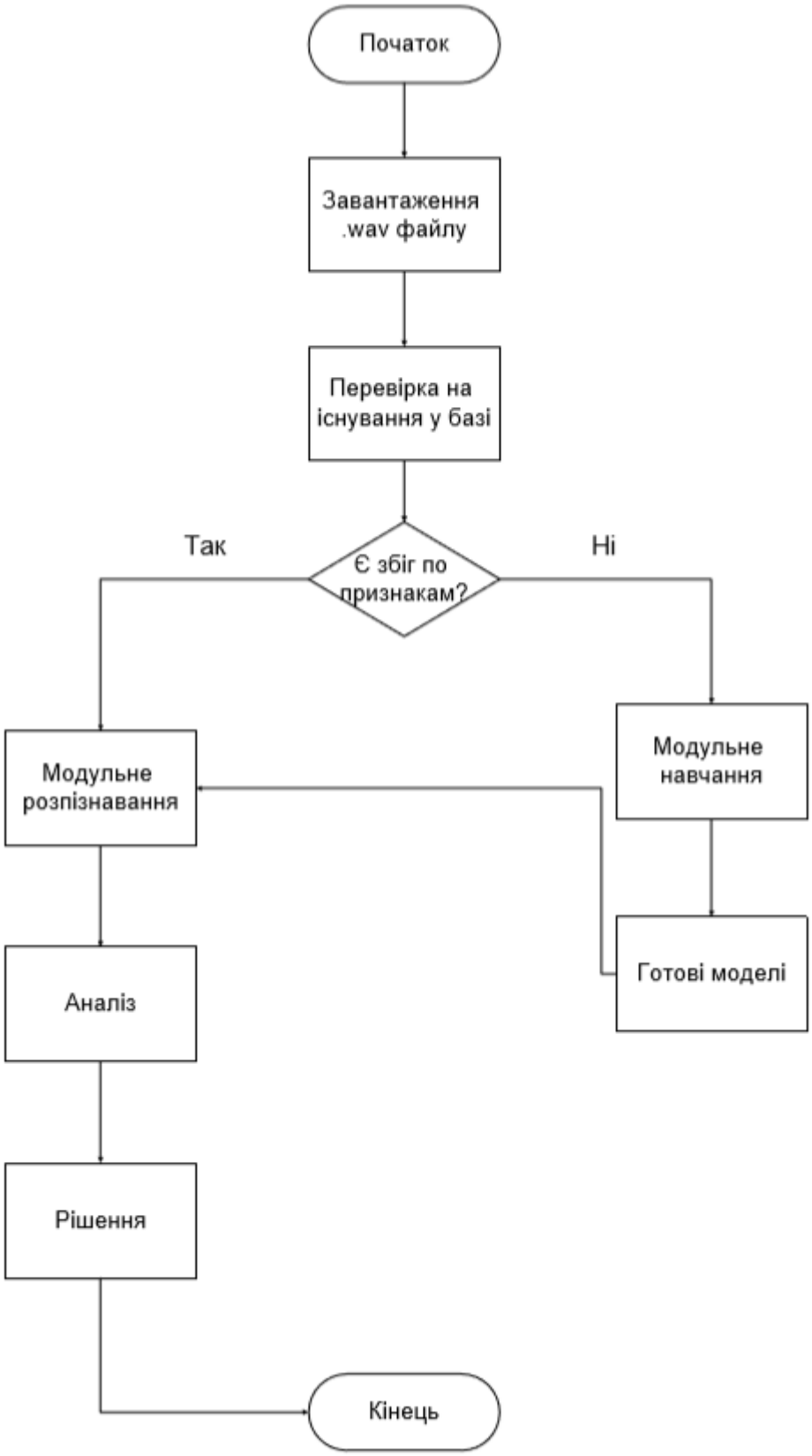


Рисунок 2 – Загальна схема роботи з .wav файлами

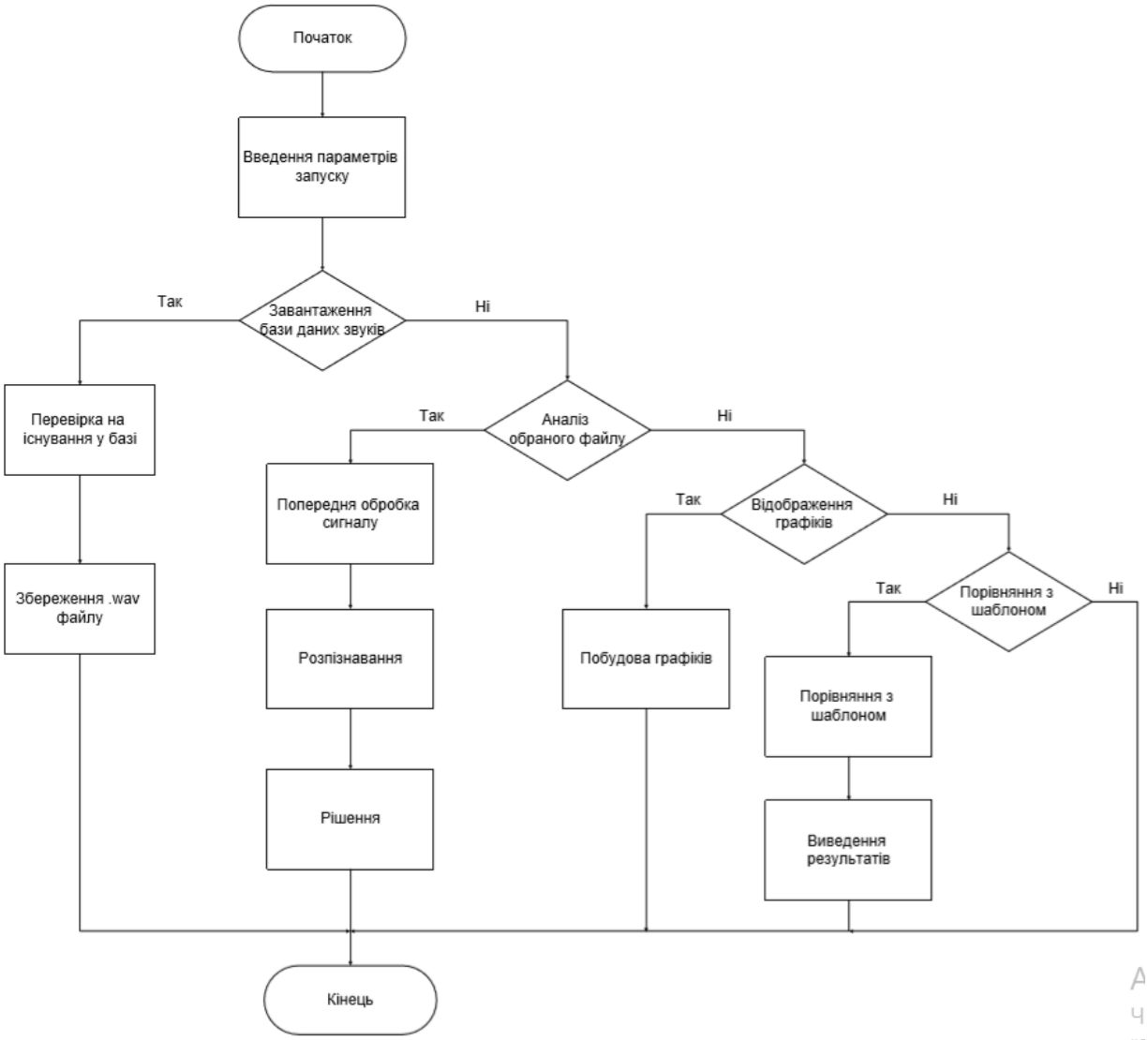


Рисунок 3 – Деталізована схема роботи програми

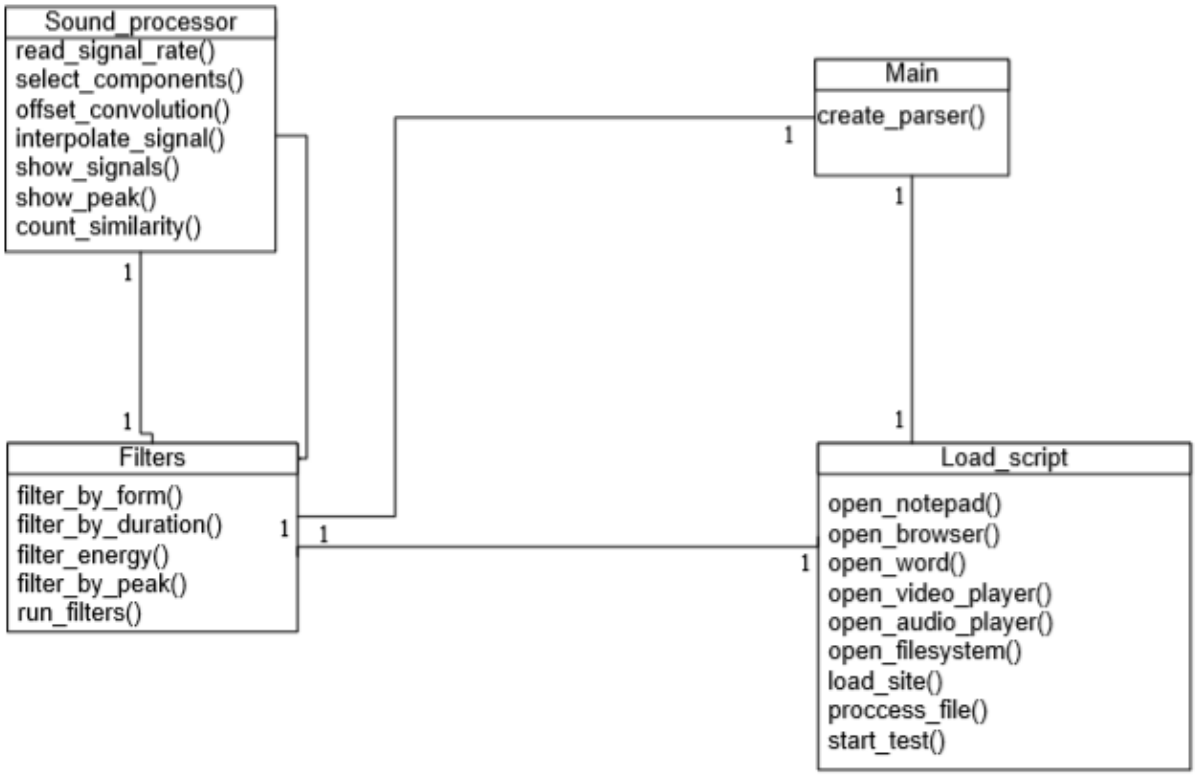


Рисунок 4 – Взаємодія класів

Рисунок 5 – Загальна схема роботи програми з голосовою командою

Рисунок 6 – Компоненти систем розпізнавання мови

Додаток 2. Копії публікацій за темою магістерської дисертації

X наукова конференція магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2018 (Київ, 21-23 березня 2018 р.).

УДК 004.934

К.т.н., доцент Терейковський І.А., студент Шуліка В.П.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

РОЗПІЗНАВАННЯ КОМАНД ГОЛОСОВОГО УПРАВЛІННЯ КОМП'ЮТЕРНИХ СИСТЕМ НА ОСНОВІ МЕТОДУ ДИНАМІЧНОГО ПРОГРАМУВАННЯ

Abstract

Ihor A. Terejkowski, assoc. prof.; Vladyslav Shulika, student

Recognition of voice command for control computer system based on dynamic programming method

This paper concerns the description of the proposed algorithm for voice command recognition by using the dynamic programming method for computer system remote control.

Вступ

Мова є найбільш природною формою людського спілкування і тому реалізація інтерфейсу на основі аналізу мовної інформації є перспективним напрямком розвитку інтелектуальних систем управління.

Задача розпізнавання мовної інформації є складною задачею, яка використовує такі області науки як: цифрова обробка сигналів, розпізнавання образів та лінгвістика [2].

Діалог з комп'ютерами, роботами, автоматизованими системами управління за допомогою голосових повідомлень відкриває великі перспективи:

- простота спілкування з системою;
- доступність мовного інтерфейсу людям з порушеннями опорно-рухового та зорового апарату;
- можливість роботи користувачів в умовах перевантаженості тактильно-зорового каналу.

Постановка задачі

Метою дослідження є розробка ефективного способу розпізнавання мовних комп'ютерних команд людини на основі методу динамічного програмування.

Опис

Процес розпізнавання голосових команд можна розділити на два етапи - аналіз звукового сигналу та класифікація цієї команди.

Аналіз звукового сигналу призначений для отримання опису мовного сигналу – представлення мовного сигналу у вигляді набору ознак, які зберігають інформацію про зміст мовного повідомлення.

На етапі класифікації по отриманим даним голосова команда відноситься до того або іншого класу команд. Класом виступає одна команда.

Загальна модель розпізнавання зображена на рис. 1.

Рисунок 1 – Модель розпізнавання голосових команд

Аналіз звукового сигналу включає в себе аналого-цифрове перетворення, нормування отриманих даних та отримання з цих даних частотної характеристики звукового сигналу. Сигнал подається на аналого-цифровий перетворювач, який з деякою частотою (частотною дискретизації), записує поточний рівень сигналу в цифровій формі, тобто квантує сигнал за часом і по амплітуді [1].

Ряди Фур'є дозволяють виразити складну функцію сумою простих. Так як вимовлені одні і ті ж команди будуть мати майже однакові частотні

характеристики, то для нашої задачі найвигідніше взяти саме частотну характеристику звукового сигналу. Частотний аналіз дозволяє отримати розподіл амплітуди по частоті (амплітудо-частотні спектри) та розподіл фаз складових по частотам (фазо - частотні спектри).

Часовий динамічний алгоритм (DTW) обчислює оптимальну послідовність трансформації (деформації) часу між двома часовими рядами. Алгоритм обчислює обидва значення деформації між двома рядами й відстанню між ними [3].

Припустимо, що є дві числові послідовності $(a_1, a_2, \dots, a_n)$ та $(b_1, b_2, \dots, b_n)$. Як видно, довжина двох послідовностей може бути різною. Алгоритм починається з розрахунків локальних відхилень між елементами двох послідовностей, що використовують різні типи відхилень. Найпоширеніший спосіб для обчислення відхилень – метод, що розраховує абсолютне відхилення між значеннями двох елементів (евклідова відстань). У результаті отримаємо матрицю відхилень, що має n рядків і m стовпців загальних членів: $d_{ij} = |a_i - b_j|, i = \overline{1,n}, j = \overline{1,m}$ [4].

Мінімальна відстань у матриці між послідовностями визначається за допомогою алгоритму динамічного програмування та наступного критерію оптимізації:$a_{ij} = d_{ij} + min(a_{i-1,j-1}, a_{i-1,j}, a_{i,j-1})$, де: a_{ij} – мінімальна відстань між послідовностями $(a_1, a_2, \dots, a_n)$ та $(b_1, b_2, \dots, b_n)$. Шлях деформації – це мінімальна відстань у матриці між елементами a_{11} та a_{nm}, що складаються із тих a_{ij} елементів, які виражають відстань до a_{nm}.

Глобальні деформації складаються із двох послідовностей і визначаються по формулі: $GC = \frac{1}{p}\sum_{i=1}^{p} w_i$, де: w_i – елементи, які відносяться шляху деформації; p – їх кількість.

Існує три умови, що забезпечують роботу DTW відповідно до алгоритму забезпечення швидкої конвергенції [5]:

1. Монотонність – шлях ніколи не повертається назад і не повторюється, тобто індекс i та j, які використовуються, ніколи не зменшуються.

2. Безперервність – послідовність просувається поступово: за один крок індекси i та j збільшуються не більше ніж на 1.

3. Граничність – послідовність починається в лівому нижньому куті й закінчується в правому верхньому.

Висновки

Розглянута модель розпізнавання голосових команд призначена для створення мовного інтерфейсу, який дозволить істотно підвищити ефективність роботи людино–машинних систем. Ця модель базується на використанні частотного аналізу мовного сигналу, зокрема, перетворення Фур'є, що забезпечить високу швидкодію програмних засобів. Класифікація мовних команд виконується на основі часового динамічного алгоритму, який опрацьовує кожний із наборів однієї команди та надає середнє значення схожості, що дозволяє отримати більший коефіцієнт точності, на відміну від, інших систем розпізнавання команд.

Література

1. *Казакова Н. Ф.* Аналіз напрямів розвитку інформаційної безпеки у комп'ютерних системах та мережах на основі застосування програмних засобів захисту інформації [Текст] / Н. Ф. Казакова // Вісник Львівського національного аграрного університету: Агроінженерні дослідження. – 2010. – № 14. – С. 47-57.

2. *Фразе-Фразенко О. О.* Спосіб регуляризації некоректно поставленої задачі розпізнавання у системах телебачення замкнутого контуру [Текст] / О. О. Фразе-Фразенко // Східно-Європейський журнал передових технологій. – 2012. – № 6/4(8). – С. 19-20.
3. *Рыбальский О. В.* Анализ возможных цифровых и аналоговых способов подделки фонограмм и требований к анализаторам для выявления их следов [Текст] / О. В. Рыбальский // Захист інформації. – 2004. – Спеціальний випуск. – С. 44-48
4. *Местецький Л.М.*. Математические методы распознавание образов, М.: МГУ, ВМиК, 2002-2004. -85 с.
5. *F. Jelinek* «Continuous Speech Recognition by Statisical Methods.» IEEE Proceedings 64:4(1976): 532-556 с.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Київський національний університет
імені Тараса Шевченка

І МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА

КОНФЕРЕНЦІЯ

“ПРОБЛЕМИ КІБЕРБЕЗПЕКИ ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ” (PCSITS)

05-06 квітня 2018 року

УДК 004.934

К.т.н. Терейковська Л.О., студент Шуліка В.П.

Національний університет будівництва і архітектури,
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ГОЛОСОВЕ УПРАВЛІННЯ КОМП’ЮТЕРНИМИ СИСТЕМАМИ

Мова є найбільш природною формою людського спілкування і тому реалізація інтерфейсу на основі аналізу голосового сигналу є одним із найбільш перспективних напрямків розвитку засобів управління комп’ютерними системами. Діалог з комп’ютерними системами за

допомогою голосових повідомлень відкриває великі перспективи: доступність мовного інтерфейсу людям з порушеннями опорно-рухового та зорового апарату; можливість роботи користувачів в умовах перевантаженості тактильно-зорового каналу. Слід зазначити, що вирішення задачі голосового управління деталізується на ряд достатньо ізольованих між собою підзадач. Однією із таких підзадач є розробка ефективного способу розпізнавання голосових команд. Оскільки кількість голосових команд комп'ютерної системи досить обмежена, а потужність сучасних обчислювальних засобів достатньо висока, то для розпізнавання можливо застосувати метод динамічного програмування. Цим визначається завдання представленої наукової роботи – розробка підходів до розпізнавання голосових команд комп'ютерної системи на основі методу динамічного програмування.

Процес розпізнавання голосових команд можна розділити на два етапи - аналіз голосового сигналу та класифікація поданої команди.

Аналіз голосового сигналу реалізується з метою його представлення у вигляді набору параметрів, які характеризують інформацію про зміст мовного повідомлення.

На етапі класифікації по отриманим даним голосова команда відноситься до того або іншого класу команд. Класом виступає одна команда.

Загальна модель розпізнавання представлена на рис. 1.

Рисунок 1 – Модель розпізнавання голосових команд

Аналіз голосового сигналу включає в себе аналого-цифрове перетворення, нормування отриманих даних та отримання з цих даних частотної характеристики звукового сигналу.

Спочатку сигнал подається на аналого-цифровий перетворювач, який з деякою частотою (частотою дискретизації), записує поточний рівень сигналу в цифровій формі, тобто квантує сигнал за часом і по амплітуді [1]. Після цього за допомогою методу Фур'є розраховується частотно-амплітудна характеристика сигналу. Частотний аналіз дозволяє отримати розподіл амплітуди по частоті (амплітудо-частотні спектри) та розподіл фаз складових по частотам (фазо - частотні спектри).

Власне розпізнавання голосових команд на основі методів динамічного програмування може бути здійснене за допомогою так званого DTW-алгоритм.

Даний алгоритм спрямований на обчислення оптимальної послідовності трансформації (деформації) часу між двома часовими рядами. Алгоритм обчислює обидва значення деформації між двома рядами й відстанню між ними [2]. Припустимо, що є дві числові послідовності $(a_1, a_2, \dots, a_n)$ та $(b_1, b_2, \dots, b_n)$. Алгоритм починається з розрахунків локальних відхилень між елементами двох послідовностей, що використовують різні типи відхилень. Найпоширеніший спосіб для обчислення відхилень – метод, що розраховує абсолютне відхилення між значеннями двох елементів (евклідова відстань). У результаті отримаємо матрицю відхилень, що має n рядків і m стовпців загальних членів: $d_{ij} = \left|a_i - b_j\right|, i = \overline{1,n}, j = \overline{1,m}$ [3].

Мінімальна відстань у матриці між послідовностями визначається за допомогою алгоритму динамічного програмування та наступного критерію оптимізації:$a_{ij} = d_{ij} + min\left(a_{i-1,j-1}, a_{i-1,j}, a_{i,j-1}\right)$, де: a_{ij} – мінімальна відстань між послідовностями $(a_1, a_2, \dots, a_n)$ та $(b_1, b_2, \dots, b_n)$. Шлях деформації – це мінімальна відстань у матриці між елементами a_{11} та a_{nm}, що складаються із тих a_{ij} елементів, які виражають відстань до a_{nm}.

Глобальні деформації складаються із двох послідовностей і визначаються по формулі: $GC = \frac{1}{p}\sum_{i=1}^{p} w_i$, де: w_i – елементи, які відносяться шляху деформації; p – їх кількість.

Існує три умови, що забезпечують роботу DTW-алгоритму відповідно до вимоги забезпечення швидкої конвергенції [3]:

1. Монотонність – шлях ніколи не повертається назад і не повторюється, тобто індекс i та j, які використовуються, ніколи не зменшуються.

2. Безперервність – послідовність просувається поступово: за один крок індекси i та j збільшуються не більше ніж на 1.

3. Граничність – послідовність починається в лівому нижньому куті й закінчується в правому верхньому.

Розглянута модель розпізнавання голосових команд призначена для створення голосового інтерфейсу, який дозволить істотно підвищити ефективність роботи людино–машинних систем. Ця модель базується на використанні частотного аналізу голосового сигналу, зокрема, перетворення Фур'є, що забезпечить високу швидкодію програмних засобів. Класифікація голосових команд виконується на основі часового динамічного алгоритму, який опрацьовує кожний із наборів однієї команди та надає середнє значення схожості, що дозволяє отримати більший коефіцієнт точності, на відміну від інших систем розпізнавання команд.

Література

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

7. Rabiner L.R. Juang B., Fundamentals of Speech Recognition // Pearson Education. -1993. -507p.

8. *Местецький Л.М.*. Математические методы распознавание образов, М.: МГУ, ВМиК, 2002-2004. -85 с.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МІНІСТЕРСТВО КУЛЬТУРИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ КУЛЬТУРИ І МИСТЕЦТВ
Факультет економіки, права та інформаційних
технологій
Кафедра комп'ютерних наук
МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА

КОНФЕРЕНЦІЯ
студентів і молодих учених
"ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В СОЦІОКУЛЬТУРНІЙ СФЕРІ, ОСВІТІ, ЕКОНОМІЦІ ТА ПРАВІ"
18-19 квітня 2018 р.

УДК 004.934

К.т.н., доцент Терейковський І.А., студент Шуліка В.П

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Київ, Україна

СИСТЕМА ВІДДАЛЕНОЇ ВЗАЄМОДІЇ КОМП'ЮТЕРА ТА КОРИСТУВАЧА ЗА ДОПОМОГОЮ ГОЛОСУ

Вступ

Мова є найбільш природною формою людського спілкування і тому реалізація інтерфейсу на основі аналізу мовної інформації є перспективним напрямком розвитку інтелектуальних систем управління.

Задача розпізнавання мовної інформації є складною задачею, яка використовує такі області науки як: цифрова обробка сигналів, розпізнавання образів та лінгвістика [2].

Діалог з комп'ютерами, роботами, автоматизованими системами управління за допомогою голосових повідомлень відкриває великі перспективи:

- простота спілкування з системою;
- доступність мовного інтерфейсу людям з порушеннями опорно-рухового та зорового апарату;
- можливість роботи користувачів в умовах перевантаженості тактильно-зорового каналу.

Постановка задачі

Метою дослідження є розробка ефективного способу розпізнавання мовленнєвих комп'ютерних команд людини на основі методу динамічного програмування.

Опис

Процес розпізнавання голосових команд можна розділити на два етапи - аналіз звукового сигналу та класифікація цієї команди.

Аналіз звукового сигналу призначений для отримання опису мовного сигналу – представлення мовного сигналу у вигляді набору значень ознак, які зберігають інформацію про зміст мовного повідомлення.

На етапі класифікації по отриманим даним голосова команда відноситься до того або іншого класу команд. Класом виступає одна команда.

Загальна модель розпізнавання представлена на рис. 1.

Рисунок1 – Модель розпізнавання голосових команд

Аналіз звукового сигналу включає в себе аналого-цифрове перетворення, нормування отриманих даних та отримання з цих даних частотної характеристики звукового сигналу. Сигнал подається на аналого-цифровий перетворювач, який з деякою частотою (частотною дискретизації), записує поточний рівень сигналу в цифровій формі, тобто квантує сигнал за часом і по амплітуді [1].

Ряди Фур'є дозволяють виразити складну функцію сумою простих. Так як вимовлені одні і ті ж команди будуть мати майже однакові частотні характеристики, то для нашої задачі найвигідніше взяти саме частотну характеристику звукового сигналу. Частотний аналіз дозволяє отримати розподіл амплітуди по частоті (амплітудо-частотні спектри) та розподіл фаз складових по частотам (фазо - частотні спектри).

Часовий динамічний алгоритм (DTW) обчислює оптимальну послідовність трансформації (деформації) часу між двома часовими рядами. Алгоритм обчислює обидва значення деформації між двома рядами й відстанню між ними [3].

Припустимо, що є дві числові послідовності $(a_1, a_2, \dots, a_n)$ та $(b_1, b_2, \dots, b_n)$. Як видно, довжина двох послідовностей може бути різною. Алгоритм починається з розрахунків локальних відхилень між елементами двох послідовностей, що використовують різні типи відхилень. Найпоширеніший спосіб для обчислення відхилень – метод, що розраховує абсолютне відхилення між значеннями двох елементів (евклідова відстань).

У результаті отримаємо матрицю відхилень, що має n рядків і m стовпців загальних членів: $d_{ij} = |a_i - b_j|, i = \overline{1,n}, j = \overline{1,m}$ [4].

Мінімальна відстань у матриці між послідовностями визначається за допомогою алгоритму динамічного програмування та наступного критерію оптимізації:$a_{ij} = d_{ij} + min(a_{i-1,j-1}, a_{i-1,j}, a_{i,j-1})$, де: a_{ij} – мінімальна відстань між послідовностями $(a_1, a_2, \dots, a_n)$ та $(b_1, b_2, \dots, b_n)$. Шлях деформації – це мінімальна відстань у матриці між елементами a_{11} та a_{nm}, що складаються із тих a_{ij} елементів, які виражають відстань до a_{nm}.

Глобальні деформації складаються із двох послідовностей і визначаються по формулі: $GC = \frac{1}{p}\sum_{i=1}^{p} w_i$, де: w_i – елементи, які відносяться шляху деформації; p – їх кількість.

Існує три умови, що забезпечують роботу DTW відповідно до алгоритму забезпечення швидкої конвергенції [5]:

1. Монотонність – шлях ніколи не повертається назад і не повторюється, тобто індекс i та j, які використовуються, ніколи не зменшуються.

2. Безперервність – послідовність просувається поступово: за один крок індекси i та j збільшуються не більше ніж на 1.

3. Граничність – послідовність починається в лівому нижньому куті й закінчується в правому верхньому.

Висновки

Розглянута модель розпізнавання голосових команд призначена для створення мовного інтерфейсу, який дозволить істотно підвищити ефективність роботи людино–машинних систем. Ця модель базується на використанні частотного аналізу мовного сигналу, зокрема, перетворення Фур'є, що забезпечить високу швидкодію програмних засобів. Класифікація мовних команд виконується на основі часового

динамічного алгоритму, який опрацьовує кожний із наборів однієї команди та надає середнє значення схожості, що дозволяє отримати більший коефіцієнт точності, на відміну від, інших систем розпізнавання команд.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

9. Rabiner L.R. Juang B., Fundamentals of Speech Recognition // Pearson Education. -1993. -507p.
10. Аграновский А. В. Теоретические аспекты алгоритмов обработки и классификации сигналов / А. В. Аграновский, Д. А. Леднов. — М. : Радио и связь, 2004. — 164 с.
11. *Рыбальский О. В.* Анализ возможных цифровых и аналоговых способов подделки фонограмм и требований к анализаторам для выявления их следов [Текст] / О. В. Рыбальский // Захист інформації. – 2004. – Спеціальний випуск. – С. 44-48
12. *Местецький Л.М.*. Математические методы распознавание образов, М.: МГУ, ВМиК, 2002-2004. -85 с.
13. *F. Jelinek* «Continuous Speech Recognition by Statisical Methods.» IEEE Proceedings 64:4(1976): 532-556 с.

Додаток 3. Фрагмент 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')

            if ((statement.find("блокнот") != -1) or (statement.find("notepad")
!= -1)):
                self.osrun('notepad')

            if ((statement.find("paint") != -1) or (statement.find("паинт") != -
1)):
                self.osrun('mspaint')

            if ((statement.find("browser") != -1) or (statement.find("браузер")
!= -1)):
                self.openurl('http://google.ru', 'Открываю браузер')

            # Команды для открытия URL в браузере

            if (((statement.find("youtube") != -1) or (statement.find("youtub")
!= -1) or (
                    statement.find("ютуб") != -1) or (statement.find("you tube") !=
-1)) and (
                    statement.find("смотреть") == -1)):
                self.openurl('http://youtube.com', 'Открываю ютуб')

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

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

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

```
            # Команды для поиска в сети интернет

            if ((statement.find("найти") != -1) or (statement.find("поиск") != -
1) or (
                    statement.find("найди") != -1) or (statement.find("дайте") != -
1) or (
                    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")

            if ((statement.find("смотреть") != -1) and (
                    (statement.find("фильм") != -1) or (statement.find("film") != -
1))):
                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+онлайн+фильм+' +
statement,
                             "Выберите сайт где смотреть фильм")

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

                if ((statement.find("слушать") != -1) and (statement.find("песн") !=
-1)):
                    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"], language="zh-CN"),
u"砸 自己 的 脚 ")



if __name__ == "__main__":
    unittest.main()
```