

2 РОЗРОБКА МЕТОДИКИ РОЗПІЗНАВАННЯ МОВНИХ КОМАНД В СИСТЕМАХ ЛЮДИНО-МАШИННОЇ ВЗАЄМОДІЇ

За результатами аналізу першого розділу було виділено недоліки підходів та засобів до розпізнавання мовних команд, а саме: недостатня достовірність визначення команди та необхідний великий об'єм бази даних, знайти екземпляри для якої може бути важкою справою.

Виходячи з інформації про ці недоліки, можна виділити метод порівняння з шаблонами, який дозволяє зберегти достатню достовірність розпізнавання команд. Отже, методика розпізнавання мовних команд буде створюватись з використанням цього методу.

2.1 Розробка методики розпізнавання мовних команд

2.1.1 Виділення основи для методики

Прямі аналоги систем, що виконують розпізнавання команд, а не перетворення мови в текст, використовують лише повні слова для розпізнавання. При цьому більша частина команди є зайвою, адже звичайно не використовуються слова у якості команд, які відрізняються лише префіксом або закінченням. Префікс та закінчення у слові є досить великою часткою, яка для різних слів може дуже часто повторюватися. Отже це може негативно вплинути на достовірність розпізнавання команди. Приклад зображено на рис. 2.1:

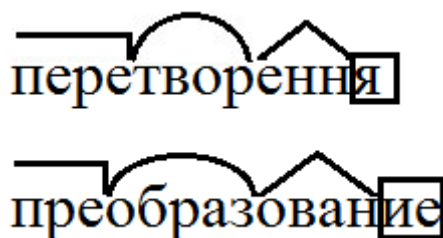


Рисунок 2.1 — Розбір команди за будовою російською та українською мовами

| | | | | | | |
|-------|------|----------|--------|------|-------------------|------|
| | | | | | ІС КРМ 122 035 ПЗ | Лист |
| Змін. | Лист | № докум. | Підпис | Дата | | 1 |

Для даної команди префікс та закінчення становить досить велику частку від слова. Осцилограму сигналів команди можна побачити на рис. 2.2:

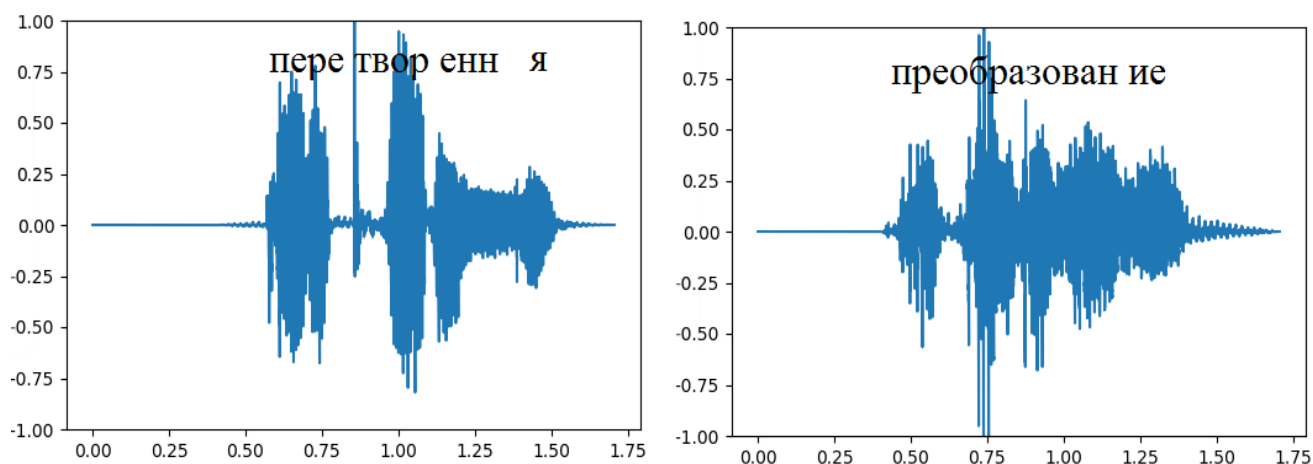


Рисунок 2.2 — Осцилограма команди російською та українською мовами

Таким чином, з представленого прикладу можна побачити, що найбільшу інформативність та унікальність містить у собі корінь слова. Суфікс не є необхідним для розпізнавання команди, але через велику варіативність та можливість мати у слові декілька суфіксів, виділення його у слові може бути складною задачею. До того ж ця задача буде потребувати додаткові ресурси або час, що негативно вплине на загальне розпізнавання.

Коріння слів є майже у кожномуязику світу [5], що є значним плюсом, адже обробку слова можна буде представити, у загальному випадку, як обробку кореня, що скоротить пошук співпадань у базі та дає можливість відійти від певної мови команди.

Такий підхід дозволить зменшити розмір даних, що зберігаються у базі. Причому, якщо для одного аудіо файлу подібне зменшення буде незначним, то для однієї команди, яка визначена щонайменше десятками файлами, різниця у розмірі буде досить помітною.

Якщо видалити малоінформативні префікс та закінчення, залишивши у слові лише корінь та суфікс, можна досягти збільшення достовірності розпізнавання у порівнянні з використанням повних слів у якості команд.

Можна зробити додаткове підтвердження на виконання і додатково забезпечити більш правильне розпізнавання для команд, які становлять найбільшу небезпеку для даних користувача, в разі, коли команда інтерфейсу була неправильно розпізнана. Для цього, припустимо, можливо використовувати більшу кількість еталонних екземплярів команди, що може спричинити за собою більш довге розпізнавання команди, проте також і більшу ймовірність її правильного визначення.

Якщо додати рівень важливості (або небезпеки) команди, буде змога корегувати суперечливі результати при розпізнаванні. Тобто, якщо команда може спричинити втрату важливих даних, їй присвоюється більший рівень небезпеки та відповідно необхідна досить велика чіткість вимовляння команди для її застосування.

2.1.1 Запропонована методика розпізнавання мовних команд

У даній роботі пропонується методика розпізнавання мовних команд в системах ЛМВ, яка полягає у наступному:

- виділення кореню у мовній команді одним з двох розроблених способів за допомогою використання алгоритму DTW;
- розпізнавання команди за отриманим коренем та еталонними значеннями команд з бази даних.

Такий підхід дозволяє збільшити достовірність розпізнавання за рахунок досить великого розміру кореню слова, а також позбавлення від найбільше повторюваних у різних словах часток – префіксу та закінчення. Також це зменшить розмір необхідної бази даних для розпізнавання за рахунок зменшення розміру величини даних та прискорить обробку даних при розпізнаванні, що вплине на загальну продуктивність системи ЛМВ з використанням цієї методики. Запропонована методика розпізнавання мовних команд в системах ЛМВ представлена на рис. 2.3:

| | | | | | | |
|-------|------|----------|--------|------|-------------------|------|
| | | | | | ІС КРМ 122 035 ПЗ | Лист |
| | | | | | | 2 |
| Змін. | Лист | № докум. | Підпис | Дата | | |

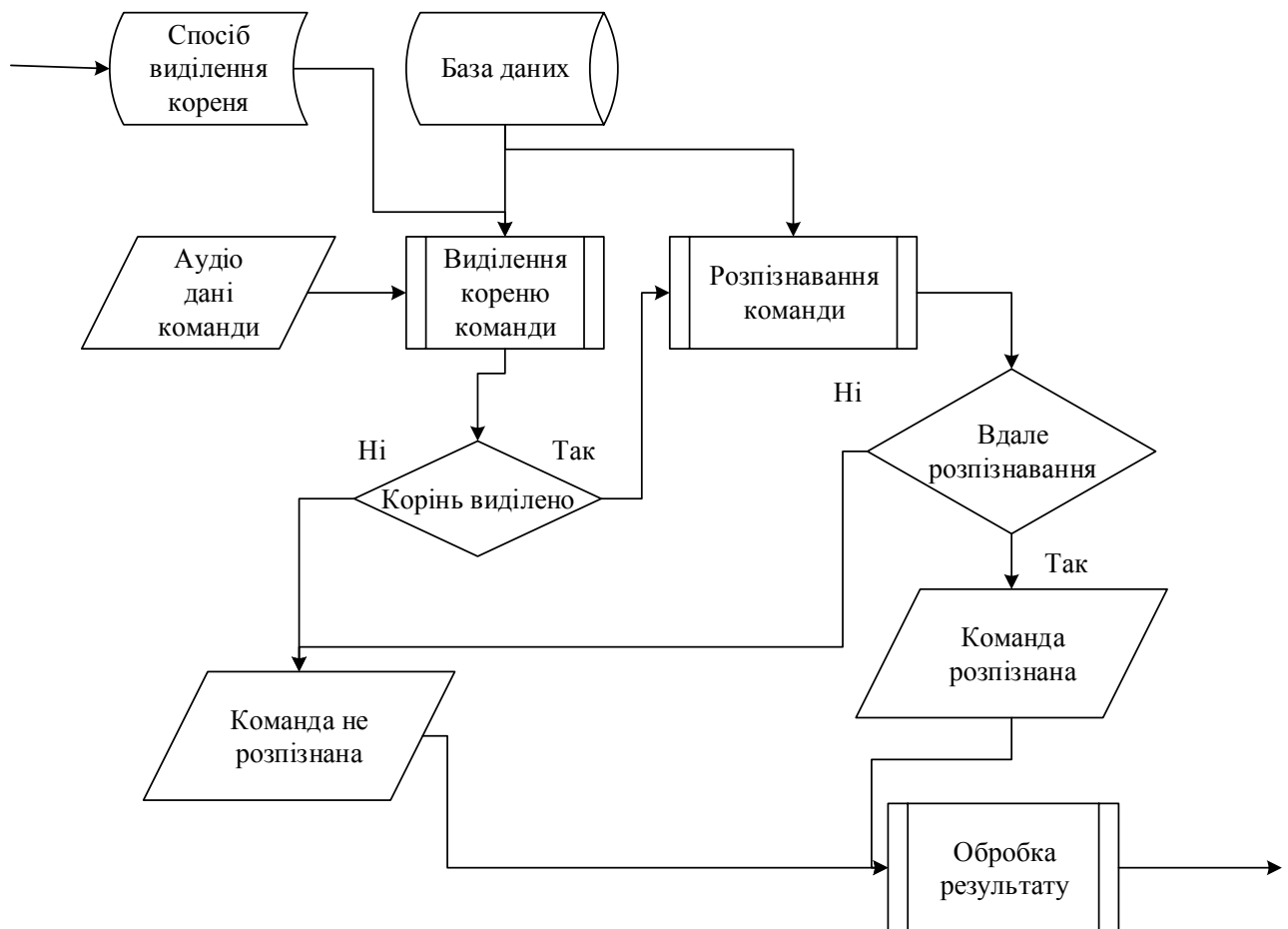


Рисунок 2.3 — Запропонована методика розпізнавання мовних команд

2.2 Розробка способів виділення частин слова з команди

При аналізі текстового представлення слів, найбільш адекватним підходом було б приведення кожного слова до його початкової форми та подальшому пошуку у слові окремих частин за допомогою бази даних цих частин. Однак при роботі зі звуковим представленням слів такий підхід неможливий.

Для того, щоб привести слово к його початковій формі, необхідно заздалегідь розпізнати його. Проте у цьому немає необхідності через те, що команди інтерфейсу не повинні змінювати форму слова. Зазвичай вони використовуються у початковій або принаймні у той формі, в якій вони були добавлені у систему.

Проблема полягає у іншому – необхідно мати велику базу даних з частинами слів для виконання розпізнавання їх таким способом. Та якщо для

текстового представлення необхідний об'єм пам'яті на жорсткому диску для збереження бази достатньо невеликий, то аудіо база подібного роду буде потребувати набагато більшого розміру пам'яті.

Можливо використання наступних двох підходів до розділення слів на окремі частини: з використанням префіксів та закінчень або з використанням бази коренів.

2.2.1 Використання префіксів та закінчень

Перший підхід є більш емпіричним, він потребує невеликої за розміром бази, яка включає в себе лише префікси та закінчення (рис. 2.3).



Рисунок 2.4 — Алгоритм розділення слова на частини з використанням префіксів та закінчень, аркуш 1

Згідно цього підходу, якщо знайдена достатня схожість з одним з еталонних префіксів на початку досліджуваного слова, то, приймаючи до уваги

розміри цього префіксу у порівнянні з загальним розміром аудіо представлення команди, можна зробити висновок про його наявність або відсутність

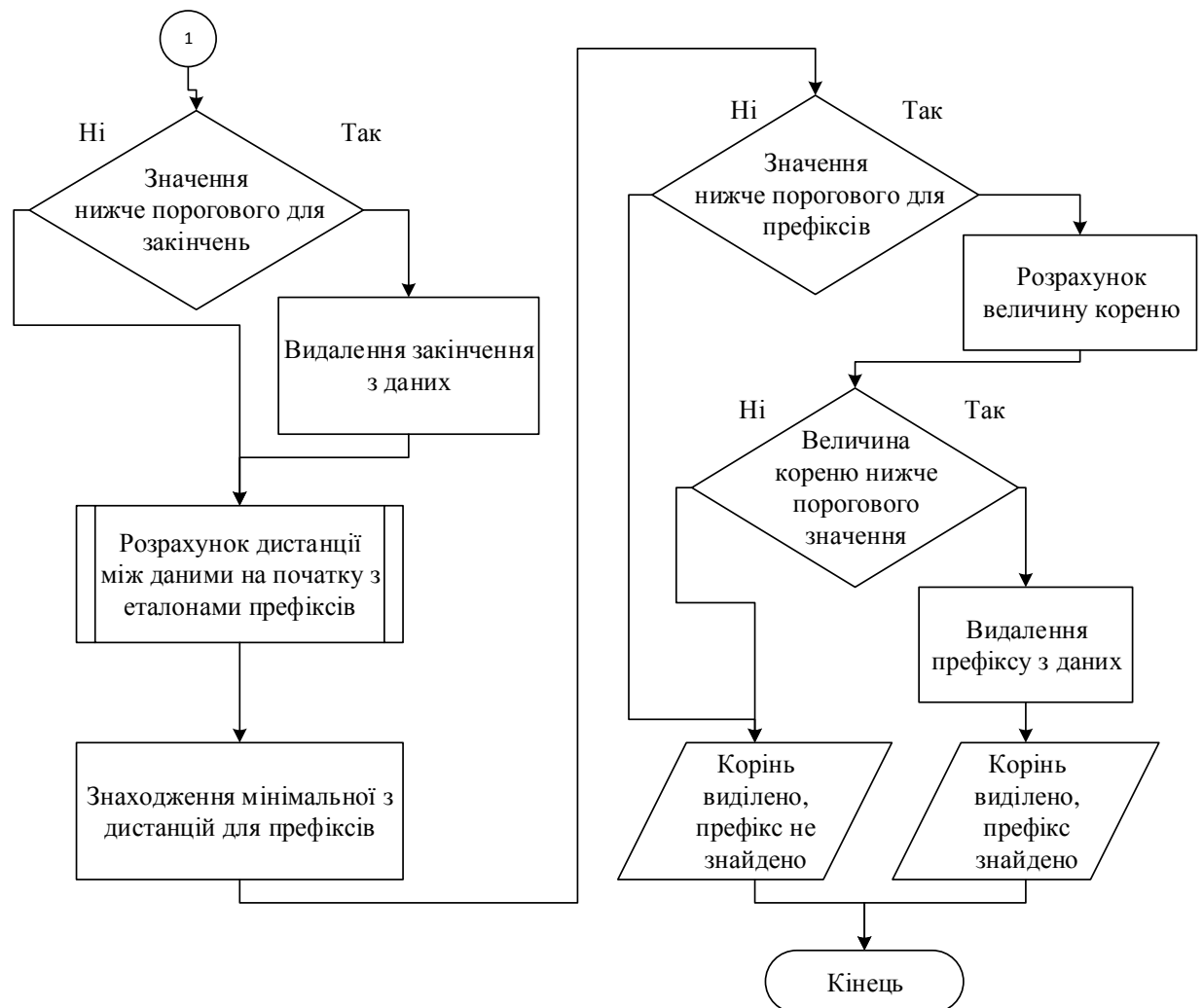


Рисунок 2.4 — Алгоритм розділення слова на частини з використанням префіксів та закінчень, аркуш 2

. Також приймається до уваги й наявність або відсутність закінчення, та, у відповідності з цим, визначається розмір передбачуваного кореня. Якщо він менше деякого порогового значення від загального розміру даних, то наявність префіксу відкидається. Такий підхід було розроблено у відповідності зі статистичними даними про розміри окремих частин у слові. Він забезпечує більшу широту використання у тому розумінні, що додавання нових команд у

систему буде досить простим процесом та буде залежати лише від того, чи є в базі даних префікси та закінчення для цієї мови.

2.2.2 Використання бази коренів

Другий підхід більш простий у представленні, проте потребує більшого об'єму бази даних та є значно менш гнучкішим у використанні в порівнянні з першим підходом (рис. 2.5).

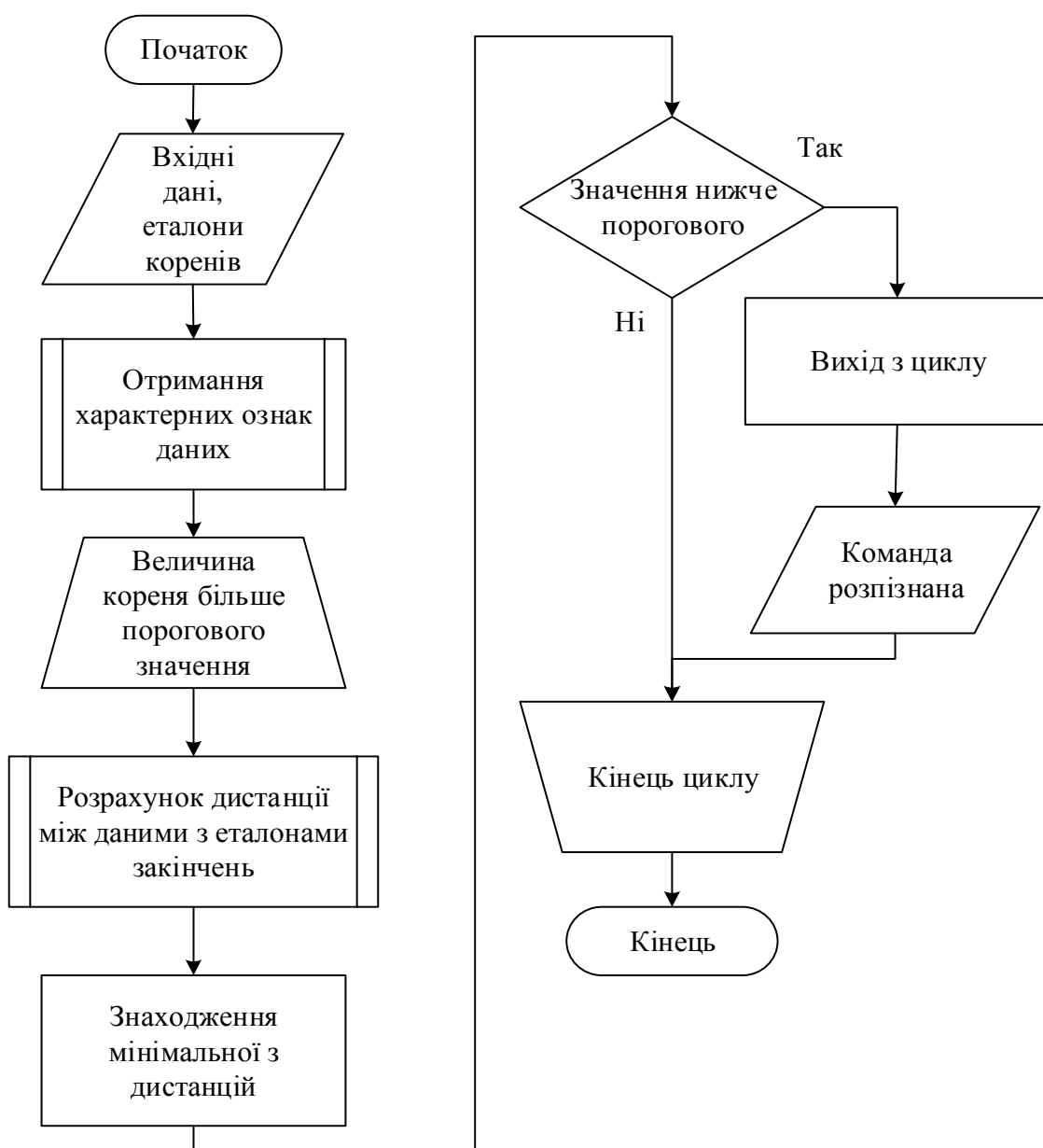


Рисунок 2.5 — Алгоритм розділення слова на частини з використанням коренів

Він полягає в отриманні бази коренів слів для знаходження кореню у вхідному слові. Такий спосіб забезпечить більшу у порівнянні з попереднім достовірність виділення та одночасне розпізнавання кореню, проте це буде можливе лише для тих коренів, які є у базі даних.

Такий підхід також дозволяє позбавитись від суфікса у слові, що значно скоротить розмір кожного з аудіо файлів. Для додавання у систему команд іншою мовою, необхідно забезпечити присутність у базі системи велику кількість коренів слів для цієї мови.

2.3 Використання алгоритму динамічної трансформації часу для оцінки збіжності даних

2.3.1 Загальне представлення алгоритму

Dynamic Time Warping (DTW) – це алгоритм для вимірювання подібності між двома часовими послідовностями, які можуть варіюватися в залежності від швидкості. Його можна використовувати для зіставлення з еталонами за допомогою мовних команд з іншими командами, навіть якщо людина говорить швидше або повільніше, ніж попередньо записаний зразок голосу. DTW може застосовуватися до тимчасових послідовностей відео-, аудіо- та графічних даних, будь-яких даних, які можуть бути перетворені в лінійну послідовність, можуть бути проаналізовані за допомогою DTW.

У загальному вигляді алгоритм можна представити таким, як зображено на рис. 2.6. Існує три умови, що накладаються на DTW алгоритм для забезпечення швидкої конвергенції. Монотонність – шлях ніколи не повертається, тобто: обидва індекси, i та j , які використовуються в послідовності, ніколи не зменшуються. Безперервність – послідовність просувається поступово: за один крок індекси, i та j , збільшуються не більше ніж на 1.

| | | | | | | |
|-------|------|----------|--------|------|-------------------|------|
| | | | | | ІС КРМ 122 035 ПЗ | Лист |
| | | | | | | 2 |
| Змін. | Лист | № докум. | Підпис | Дата | | |

Граничність – послідовність починається в лівому нижньому кутку і закінчується в правому верхньому.

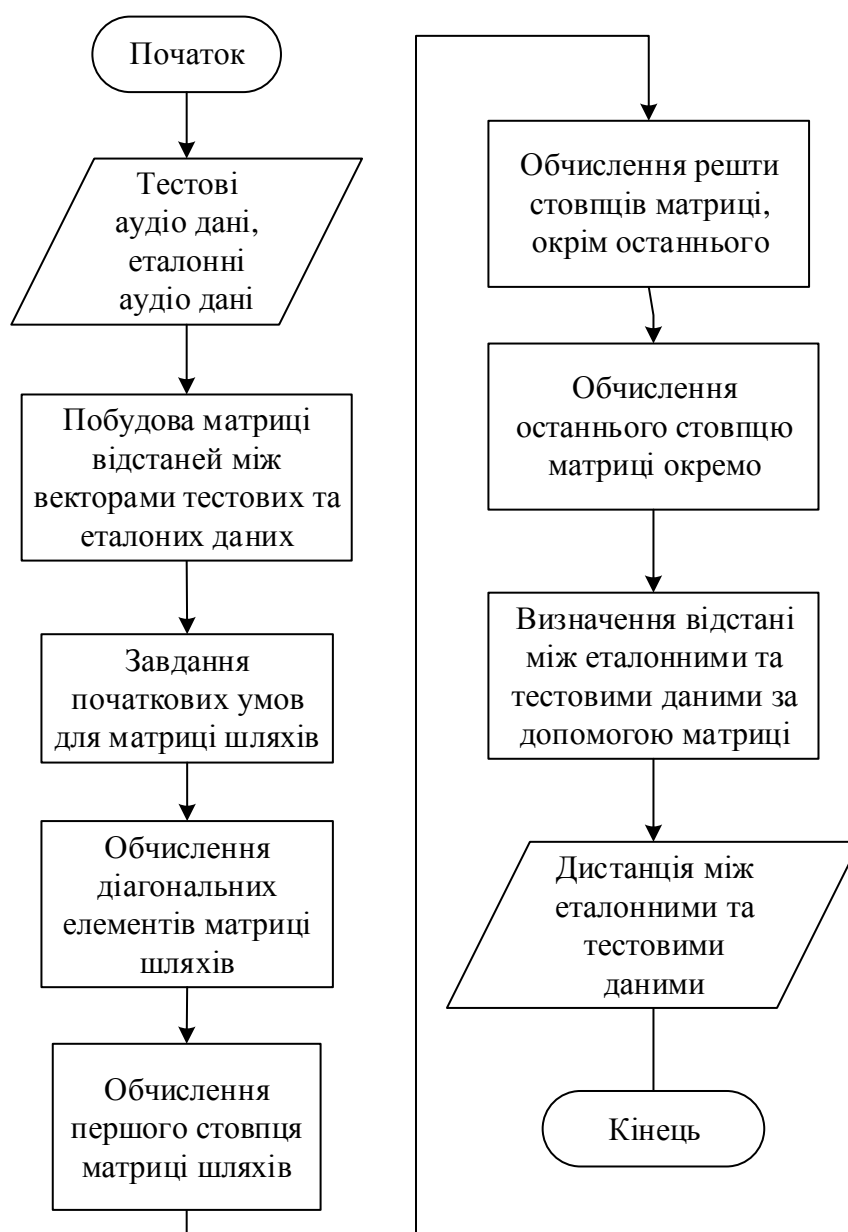


Рисунок 2.5 — Алгоритм роботи ДТЧ

Загалом, DTW – це метод, який обчислює оптимальний збіг між двома заданими послідовностями з певними обмеженнями. Наведемо приклад для двох послідовностей S і T , та розрахуємо їх збіжність. Розглянемо наступні послідовності: $S = \{1, 2, 3, 5, 5, 5, 6\}$ та $T = \{1, 1, 2, 2, 3, 5\}$

Знайдемо оптимальний збіг між цими двома послідовностями. Спочатку визначаємо відстань між двома точками, $d(x, y)$, де x і y представляють дві точки.

Створимо таблицю двомірної матриці, використовуючи ці дві послідовності. Розрахуємо відстані між кожною точкою зразка з кожною точкою тесту і знайдемо оптимальне збіг між ними.

Тут таблиця $[i][j]$ представляє оптимальну відстань між двома послідовностями, якщо розглядати послідовності $S[i]$ та $T[j]$, враховуючи всі оптимальні відстані, які спостерігалися раніше.

Для першого рядка, якщо не приймаємо значення з S , відстань між ним та T буде нескінченною. Тому записуємо нескінченність в перший рядок. Те ж саме стосується першого стовпчика. Якщо не приймати значення з T , відстань між ним та S також буде нескінченним. І відстань між 0 і 0 буде просто 0 .

Тепер для кожного кроку розглянемо відстань між кожними пунктами і додамо таку відстань, яка є мінімальною відстанню. Це дасть оптимальну відстань двох послідовностей до цієї позиції.

Для першого, $d(1, 1) = 0$, табличне значення $[0][0]$ є мінімумом. Таким чином, значення таблиці $[1][1]$ дорівнюватиме $0 + 0 = 0$. Для другого $d(1, 2) = 0$. Табличне значення $[1][1]$ являє собою мінімум також. Таблиця $[1][2] = 0 + 0 = 0$. Значення в таблиці $[7][6]$ є максимальною відстанню між цими двома заданими послідовностями. Тут 1 представляє максимальну відстань між зразком та тестом. Якщо продовжити цей шлях, після закінчення таблиці, вона буде виглядати так, як зображено на рис. 2.6.

Тепер, якщо пройти знову від останньої точки всі шляхи назад до стартової точки $(0, 0)$, отримаємо довгу лінію, яка переміщається горизонтально, вертикально і по діагоналі.

Кожен хід має своє значення. Горизонтальне переміщення являє собою видалення (тестова послідовність прискорилося протягом цього інтервалу).



Рисунок 2.6 — Приклад роботи DTW

Вертикальний хід являє собою вставку. Це означає, що тестова послідовність сповільнюється протягом цього інтервалу. Діагональний хід являє собою збіг. Протягом цього періоду тест і зразок були такими ж.

2.3.2 Способи покращення роботи алгоритму

Так як для визначення основи послідовності в динамічному програмуванні оптимальним є використання метод зворотного програмування, необхідно використовувати певний динамічний тип структури, який називається «стек». Подібно будь-якій динамічному алгоритму програмування, DWT має поліноміальну складність. Коли виконується робота з великими послідовностями, виникають дві незручності: запам'ятовування великих числових матриць та виконання великої кількості розрахунків відхилень.

Існує поліпшена версія алгоритму, FastDWT, яка вирішує дві вищевказані проблеми. Рішення полягає в розбитті матриці станів на 2, 4, 8, 16 і т.д. менших за розміром матриць, за допомогою повторюваного процесу розбиття

послідовності введення на дві частини. Таким чином, розрахунки відхилення виробляються тільки на цих невеликих матрицях, і шляхах деформації, розрахованих для невеликих матриць.

2.4 Висновки до другого розділу

Результатом створення другого розділу стала методика розпізнавання мовних команд, яка була створена за результатами аналізу у першому розділі. Вона базується на використанні під час розпізнавання лише кореню та суфіксу слова на відміну від аналогічних систем, що використовують тільки повні слова або систем з використанням ПММ, що розпізнають мовні морфеми, які зазвичай є складами або навіть меншими частинами слова.

Такий підхід до вхідних даних системи повинен забезпечити більшу достовірність у порівнянні з використанням повних команд через те, що префікси та закінчення дуже часто повторюються у різних словах на відміну від кореня слова. Він може повторюватись для декількох слів, але імовірність використання їх у системі ЛМВ досить низька. Видалення суфікса зі слова буде потребувати додаткових ресурсів за рахунок їх великої варіативності та можливої присутності у слові двох суфіксів.

Використання такого підходу дозволить також зменшити розмір бази даних за рахунок видалення з команди значної частини у вигляді префіксу та кореню. Кожна команда має деяку кількість аудіо файлів з еталонними прикладами промовляння. Отже звільнення щонайменше третини від загального об'єму бази даних покращить швидкість обробки даних та розпізнавання команд.

Основним методом для реалізації розпізнавання команди є DTW, який дозволяє провести оцінку збіжності вхідних даних до еталонних. Цей метод дозволяє адекватно оцінити збіжність аудіо даних різної довжини або швидкості промовляння слова та є одним з найкращих методів, що забезпечують високу достовірність розпізнавання безпосередньо роздільних від контексту слів.

| | | | | | | |
|-------|------|----------|--------|------|-------------------|------|
| | | | | | ІС КРМ 122 035 ПЗ | Лист |
| | | | | | | 2 |
| Змін. | Лист | № докум. | Підпис | Дата | | |