

ВИКОРИСТАННЯ ВІДСТАНІ ЛЕВЕНШТЕЙНА ПРИ РОЗРОБЦІ ГОЛОСОВОГО АСИСТЕНТА

Касумов А.І.

Науковий керівник – к.н.т., доц. Шафроненко А.Ю.

Харківський національний університет радіоелектроніки, каф. ІНФ,
м. Харків, Україна

e-mail: artur.kasumov@nure.ua

This research is devoted to exploring and implementing voice recognition methods utilizing neural networks for the development of a desktop application aimed at computer control through voice commands. Specifically, it investigates the application of the Levenshtein algorithm in the development of a voice assistant. The Levenshtein algorithm, renowned for its capability to calculate the edit distance between two strings, plays a pivotal role in enhancing the accuracy and robustness of voice recognition systems. By integrating the Levenshtein algorithm into the voice assistant, the application aims to improve its ability to accurately interpret and respond to a wide range of voice commands, thus enhancing user experience and usability.

З розвитком голосових технологій та зростанням попиту на голосові інтерфейси виникає необхідність в розробці та вдосконаленні систем розпізнавання голосу. При впровадженні голосових інтерфейсів виникають ряд складнощів, включаючи точність розпізнавання голосу та обробку різноманітних голосових команд.

Однією з основних проблем у галузі розпізнавання голосу є забезпечення високої точності розпізнавання при різних умовах оточуючого середовища.

Для вирішення цих проблем існують різні алгоритми та нейронні мережі. Нейронні мережі, навчені на обширних наборах даних, здатні адаптуватися до різних умов та забезпечувати високу точність розпізнавання. Для боротьби з проблемою нечіткого порівняння застосовуються різні алгоритми.

Розглянемо випадок, коли користувач вимовляє фразу "відкрити документ". Через акцент чи фоновий шум система може неправильно розпізнати команду як "відкрити домент". За допомогою відстані Левенштейна система порівнює відстань між двома фразами і визначає найбільш ймовірний варіант, у даному випадку "відкрити документ", і виконує відповідну дію.

Відстань Левенштейна (редакційна відстань) – метрика, яка вимірює по модулю різницю між двома рядками. Вона визначається як мінімальна кількість односимвольних операцій (вставки, видалення, заміни), необхідних для перетворення одного рядка на інший. Кожній операції (вставка, заміна, видалення) можна поставити у відповідність вартість (числовий ек-

вівалент витрат за її виконання). $w(a,b)$ – вартість заміни символу a на символ b ; $w(e,b)$ – вартість вставки символу b ; $w(a,e)$ – вартість видалення символу a . Зазвичай при розрахуванні відстань Левенштейна використовує наступна вартість операцій: $w(a,b) = 1$; $w(e,b) = 1$; $w(a,e) = 1$.

Знаходження відстань Левенштейна зводиться до знаходження такої послідовності операцій сума вартості яких буде мінімальною.

Для знаходження відстані Левенштейна існує декілька алгоритмів, одним з них є алгоритм Вагнера-Фішера (1), суть якого зводиться до побудови матриці розміру $N+1$, $M+1$ та заповнення її елементів на основі рекурентної формули наведеної нижче [0]:

$$D(i, j) = \begin{cases} 0, i = 0, j = 0, \\ i, j = 0, i > 0, \\ j, i = 0, j > 0, \\ \min \begin{cases} D(i-1, j) + 1, \\ D(i, j-1) + 1, \\ D(i-1, j-1) + m(S_1[i], S_2[j]), \end{cases} & i, j > 0 \end{cases} \quad (1)$$

де S_1 та S_2 – строки, що порівнюються, $S_1[i]$ та $S_2[j]$ – символи, що відповідають елементам матриці, $m(S_1[i], S_2[j])$ – результат порівняння $S_1[i]$ та $S_2[j]$: 0 якщо різні символи та 1, якщо однакові:

$$D(i, j) = \min \begin{cases} 0, \\ D(i-1, j) + 1, \\ D(i, j-1) + 1, \\ D(i-1, j-1) + 1, \\ D(i-1, j-1) + m(S_1[i], S_2[j]), \\ D(i-2, j-2) + 1, \quad i, j > 1, \\ S_1[i] = S_2[j-1], \\ S_1[i-1] = S_2[j], \end{cases} \quad (2)$$

де S_1 та S_2 – строки, що порівнюються, $S_1[i]$ та $S_2[j]$ – символи, що відповідають коміркам матриці, $m(S_1[i], S_2[j])$ – результат порівняння $S_1[i]$ та $S_2[j]$: 0 якщо різні символи та 1, якщо однакові. Даний метод застосовується в області обробки тексту, де важливо виявляти та виправляти помилки, пов'язані з порядком символів [2]. Алгоритм Дамерау-Левенштейна використовується коли треба виявляти та коригувати помилки, пов'язані з порядком символів в рядках. У інших випадках можна використовувати алгоритм Вагнера-Фішера, так як він більш швидкий за рахунок меншого обсягу обчислень [3]:

$$D_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{якщо } \min(i, j) = 0 \\ \min \begin{cases} D_{a,b}(i-1, j) + 1 \\ D_{a,b}(i, j-1) + 1 \\ D_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \\ D_{a,b}(i-2, j-2) + 1 \end{cases} & \text{якщо } i, j > 1 \text{ та } a_i = b_{j-1} \text{ і } a_{i-1} = b_j \\ \min \begin{cases} D_{a,b}(i-1, j) + 1 \\ D_{a,b}(i, j-1) + 1 \\ D_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{інакше.} \end{cases} \quad (3)$$

Результати роботи алгоритмів наведені в табл. 1 нижче. Проаналізувавши результати наведені в таблиці, можна зробити висновок, що алгоритм Вагнера-Фішера працює швидше в півтори рази.

Таблиця 1 – Результати роботи алгоритмів

Порівнюваний набір символів	Параметри		Алгоритм Вагнера-Фішера	Алгоритм Дамерау-Левенштейна
cat / cot	Кількість операцій: 10000	Швидкість	0.0479с	0.0647с
	Редакційна відстань		1	1
open document / open docume	Кількість операцій: 10000	Швидкість	0.5213с	0.8121с
	Редакційна відстань		3	3

Реалізація методу розпізнавання голосу на основі нейронних мереж у поєднанні з алгоритмами обробки тексту представляє собою важливий крок у розвитку голосових інтерфейсів. Такий підхід забезпечує ефективне та точне розпізнавання голосових команд при різних умовах експлуатації.

Список використаних джерел:

1. Мінайло А.Ю., Турчина В.А. Використання відстані Левенштейна для аналізу подібності даних. Питання прикладної математики і математичного моделювання. 2015.

2. Bodyanskiy, Ye V., A. Yu Shafronenko, and I. N. Klymova. "Online fuzzy clustering of incomplete data using credibilistic approach and similarity measure of special type." Radio Electronics, Computer Science, Control 1 (2021): 97-104.