УДК 004.934

М.М. Шевчук[1], Я.О. Юсин[2], Т.М. Заболотня[3]
[1] НТУУ «КПІ», м. Київ, Україна, myte@ukr.net;
[2] НТУУ «КПІ», м. Київ, Україна, yusin.yakiv@gmail.com;
[3] НТУУ «КПІ», м. Київ, Україна, tatiana104@yandex.ua

FSS ПІДХІД ДО КОРЕКЦІЇ ПОМИЛОК В СИСТЕМАХ ГОЛОСОВОГО КЕРУВАННЯ З НЕОБМЕЖЕНИМ СЛОВНИКОМ

В статті запропоновано новий FSS підхід до корекції помилок в системах голосового керування з необмеженим словником. Визначено алгоритм, який реалізує даний підхід, та дві його можливі модифікації. Сформульовано способи підвищення ефективності роботи відповідних систем голосового керування за критеріями швидкодії та використання пам'яті.

ГОЛОСОВЕ КЕРУВАННЯ, РОЗПІЗНАВАННЯ МОВЛЕННЯ, НЕОБМЕЖЕНИЙ СЛОВНИК

Вступ

На сьогодні розпізнавання мовлення (або перетворення мовленнєвого сигналу на текстовий потік) є дуже поширеною науковою задачею [1], різноманітні рішення якої використовуються як в бізнес-проектах, так і в звичайних побутових приладах. Серед трьох основних типів програмно-апаратних систем, в яких застосовуються алгоритми розпізнавання мовлення, а саме *систем голосового керування* (взаємодії користувача з пристроями за допомогою введення голосом керуючих команд), *систем голосового введення тексту* та *систем голосового пошуку* (автоматичної передачі розпізнаного мовлення до стандартної системи пошуку базою даних), наразі, найбільш розповсюдженими є цифрові системи з *голосовим керуванням*, які використовуються, наприклад, в рішеннях для систем «розумного будинку», в настільних комп'ютерах, ноутбуках та мобільних телефонах, в автомобілях, в соціальних сервісах для людей з обмеженими можливостями тощо.

Проте, не дивлячись на масштабність поширення технологій голосового керування, вони досі мають недостатню точність розпізнавання команд, що залишає негативне враження від роботи з ними у користувачів відповідних систем. Таким чином, завдання збільшення точності розпізнавання команд в системах з голосовим керуванням не втрачає своєї актуальності і доцільним є продовження дослідницької діяльності в даному напрямку.

1. Постановка задачі

Загалом, системи голосового керування можна розділити на 2 різновиди:

- *системи, що використовують обмежений словник голосових команд* – характеризуються значною швидкодією та точністю розпізнавання команд;
- *системи з необмеженим словником* – на противагу попереднім системам, не мають явно обмеженого словника команд, що надає більшої свободи дій користувачу. Недоліком таких систем є нижча точність розпізнавання, а також необхідність користувачу самому вводити команди до словника.

З огляду на вищезазначене, а також з урахуванням сучасного динамічного розвитку цифрових технологій, автори вважають за доцільне звернути увагу на вдосконалення алгоритмів роботи систем з необмеженим словником, адже вони дозволяють більш гнучко реагувати на зміну потреб ринку та розширювати перелік підтримуваних команд. Таким чином, **метою** даної роботи стало підвищення точності розпізнавання команд шляхом розробки та дослідження варіантів реалізації нового підходу до корекції помилок в системах голосового керування з необмеженим словником.

Відповідно до вказаної мети в роботі поставлені і розв'язані такі **задачі**:

- вивчення методів розпізнавання голосового потоку та існуючих підходів до збільшення точності розпізнавання;
- розроблення нового підходу до корекції помилок в системах голосового керування з необмеженим словником для збільшення точності розпізнавання команд;
- формулювання алгоритму розпізнавання команд, що реалізує запропонований підхід;
- аналіз ефективності розробленого алгоритму за критеріями швидкодії та використання пам'яті.

2. Схема виконання голосової команди та місце в ній корекції помилок

В процесі виконання команди в системах голосового керування з необмеженим словником автори вважають за доцільне виділити наступні етапи.

1. Розпізнавання голосового потоку за допомогою традиційних методів та алгоритмів (результат - текстове подання вхідного потоку).

2. Пошук розпізнаного тексту в словнику команд (результат - факт наявності команди у словнику).

3. При успішному завершенні пошуку відбувається виконання знайденої (розпізнаної) команди, в іншому випадку – пропозиція користувачу створити нову команду й додати її до словника.

Ілюстрація даного процесу наведена на рис. 1.

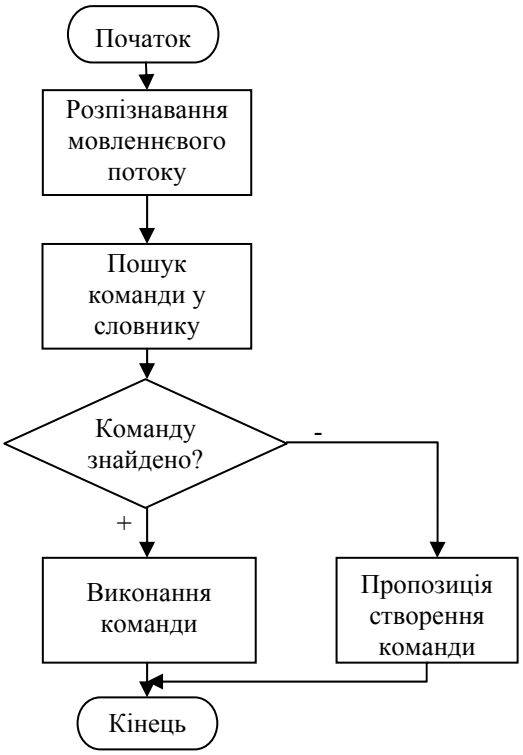

Рис. 1. Блок-схема процесу виконання голосової команди

Основними методами розпізнавання мовлення, що використовуються на першому етапі є [2]:

- часові динамічні алгоритми (dynamic time warping, DTW) – сімейство алгоритмів для вимірювання подібності між двома часовими послідовностями, котрі можуть змінюватися в часі або швидкості; передбачають порівняння мовленнєвого потоку з еталонами для знаходження максимально схожих записів; результат виконання алгоритмів не залежить від швидкості потоку;
- прихована марковська модель (hidden Markov model, HMM) – це статистична марковська модель, у якій система, що моделюється, розглядається як марковський процес з неспостережуваними (прихованими) станами. HMM використовується для розпізнавання мовлення, тому що мовленнєвий потік можна розглядати як кусково-стаціонарний сигнал або стаціонарний сигнал на короткому проміжку часу [3];
- нейронні мережі (neural networks) – на відміну від HMM базуються не на статистичному розподілі, а на природному навчанні. Це дозволяє досягти значної ефективності за критерієм точності розпізнавання під час класифікації короткочасних одиниць, таких як окремі фонеми та слова, проте ефективність значно падає зі збільшенням тривалості потоку [4];
- глибинні нейронні мережі (deep neural networks, DNN) – відрізняються від звичайної нейронної мережі декількома прихованими шарами нейронів. Завдяки цьому значно збільшується потенціал створення та навчання складних моделей мовних даних. DNN є найбільшим популярним типом акустичної моделі для розпізнавання мовлення [5].

На сьогодні окреме застосування та комбінування вищезгаданих методів розпізнавання при обробці мовленнєвого потоку дозволяє досягти точності розпізнавання в 92% [6]. Для збільшення значення даного показника до першого ж етапу виконання команди прийнято відносити і основні кроки щодо корекції помилок при розпізнаванні команд. Але оскільки самі методи розпізнавання мовленнєвого потоку є витратними і потребують залучення значних обчислювальних потужностей, виконання додаткових корекцій на цьому етапі є небажаним.

Таким чином, з огляду на вищезазначене, можна сформулювати гіпотезу про доцільність перенесення додаткової корекції помилок на другий етап виконання команди для збільшення точності її розпізнавання. Враховуючи, що на цьому етапі відбувається робота не з мовленнєвим потоком, а з текстовими даними, реалізація цього підходу до виправлення помилок має потребувати менше обчислювальних потужностей від платформи, де відбувається розпізнавання.

3. FSS підхід

Зазвичай, в системах голосового керування з необмеженим словником на другому етапі виконання команди використовуються методи, що передбачають пошук повного збігу команди із записами в словнику. В той же час в системах з обмеженим словником для корекції помилок на цьому ж етапі традиційно використовується реалізація *відстані Левенштейна* [7] або інших алгоритмів нечіткого пошуку в текстових даних, і до виконання на третьому етапі приймається команда з найменшою відстанню до заданого текстового фрагменту. Подібний підхід використовується в системах цифрового введення даних на мобільних платформах [8] та робототехніці [9].

У даній статті пропонується ***FSS підхід*** (Fuzzy String Search) до корекції помилок в системах голосового керування з необмеженим словником, який полягає у використані алгоритмів нечіткого пошуку на другому етапі процесу виконання голосової команди – при пошуку команд у словнику, а також базується на припущенні, що операція виконання існуючої команди зі словника (розширюваного користувачем) виконується набагато частіше, ніж операція створення і додавання команди до словника.

Припущення, що знаходиться в фундаменті підходу, було перевірено за допомогою проведення опитування серед користувачів голосових систем «розумного будинку». Близько 60% опитаних використовують одну команду протягом дня до 10 разів, 35% - від 10 до 20 разів та 5% більше 20 разів.

В такий спосіб замість пошуку введеної команди у словнику і наступних дій щодо виявлення помилок у команді (в разі неуспішного пошуку) пропонується виконання одразу нечіткого пошуку команди у словнику, що дозволить збільшити показники точності

розпізнавання команд без відчутного погіршення часових характеристик роботи системи.

Визначимо основні кроки алгоритму, що реалізує *FSS підхід* та отримує на вхід текстовий потік.

1. Визначити метрику для оцінювання схожості текстових даних (наприклад, відстань Левенштейна, відстань Дамерау-Левенштейна тощо), а також деяку цілочисельну константу, котру будемо називати *порогом збігу* - $T_з$.
2. Отримати перелік існуючих команд зі словника.
3. Обчислити значення відстані від кожної існуючої команди до заданого текстового потоку (відповідно до метрики, визначеної в п.1) та вибрати ту команду, відстань до якої є мінімальною.
4. Якщо відстань від обраної команди до текстового потоку є меншою або дорівнює порогу збігу, вважаємо, що команда знайдена в словнику, і переходимо до наступного етапу виконання команди. В іншому випадку вважаємо, що команди не знайдено.

Можна виділити два різновиди підходу, котрі відрізняються стратегією вибору порогу збігу:

- *модифікація зі статичним порогом збігу* – поріг збігу емпірично обирається розробником з урахуванням призначення системи та не залежить від вхідного текстового потоку і змісту словника;
- *модифікація із динамічним порогом збігу* – поріг збігу визначається на початку алгоритму як значення деякої функції, котра залежить від довжини текстового потоку. Наприклад, якщо розпізнавання мовленнєвого потоку виконується за допомогою HMM, можливим є визначення порогу як лінійної функції (чим менша довжина текстового потоку – тим менший поріг використовується).

Розглянемо декілька популярних існуючих метрик нечіткого пошуку текстових даних, котрі можуть використовуватись в конкретній реалізації запропонованого підходу: відстань Левенштейна; відстань Левенштейна з ваговими коефіцієнтами; відстань Дамерау-Левенштейна.

Відстань Левенштейна (функція Левенштейна, алгоритм Левенштейна, відстань редагування) у теорії інформації і комп'ютерній лінгвістиці є мірою відмінності двох послідовностей символів (рядків) і обчислюється як мінімальна кількість операцій вставки, видалення і заміни, необхідних для перетворення однієї послідовності символів на іншу [10]. Дана метрика має також властивості, котрі можуть бути використані для покращення ефективності алгоритму за критерієм швидкодії (див. п.4):

а) вона не є меншою, ніж різниця довжини рядків, що порівнюються;

б) вона не є більшою довжини найдовшого рядка;

в) вона дорівнює 0 тоді і тільки тоді, коли рядки є однаковими (містять однакові символи на однакових позиціях);

г) для рядків однакової довжини верхньою межею відстані редагування є відстань Гемінга (число позицій, у яких відповідні символи двох рядків однакової довжини є різними [11]).

Відстань Левенштейна з ваговими коефіцієнтами є узагальненням попередньої метрики, де кожна операція (вставка, видалення, заміна) має свій ваговий коефіцієнт для урахування ймовірності різних помилок.

Відстань Дамерау-Левенштейна – модифікація відстані Левенштейна, де до дозволених операцій вставки, видалення та заміни символу додана операція транспозиції (перестановки двох сусідніх символів).

У випадку, коли ймовірності появи окремих типів помилок в текстовому потоці є невідомими, очевидною є доцільність використання базової відстані Левенштейна для отримання більш точних результатів. Тому в наступному розділі розглянемо шляхи підвищення ефективності алгоритму, що реалізацує *FSS підхід*, за критеріями використання пам'яті та швидкодії саме з використанням цієї метрики.

4. Варіанти реалізації FSS підходу з використанням відстані Левенштейна

4.1 Спрощення обчислення відстані Левенштейна.

Найінтенсивніше використання ресурсів часу та пам'яті відповідно до алгоритму, що реалізує розроблений підхід, відбувається при обчисленні відстані між заданим текстовим потоком та командами зі словника. Базовий варіант знаходження відстані редагування має часову складність $O(mn_i)$ та використовує $O(mn_i)$ пам'яті, де m – довжина текстового потоку та n_i – довжина i-ї команди зі словника. Але завдяки особливостям роботи базового алгоритму, можна зменшити використання пам'яті до показника $O(\min(m,n_i))$. *Оскільки* в *FSS підході* не є важливим точне значення метрики, якщо воно більше порогу збігу, можна оптимізувати часову складність алгоритму з відстанню Левенштейна до $O((T_з+1)\min(m,n_i))$ (використовуючи відсікання Укконена [12]).

4.2 Перевірка необхідності знаходження відстані редагування.

Додатково до запропонованого вище удосконалення відстані Левенштейна, перед використанням останньої можна провести попередні перевірки, що базуються на її властивостях, описаних в п.3.

Якщо довжини текстового потоку та i-ї команди зі словника є рівними, доцільною є перевірка їхньої рівності (часова складність $O(m)$). Якщо вони є рівними, можна зупинити виконання алгоритму на i-й ітерації (відстань Левенштейна між такими рядками буде мінімальною і рівною 0).

Враховуючи властивість 1) відстані редагування (вона не є меншою, ніж різниця довжини рядків, що порівнюються) можна порівняти різницю довжин рядків, і якщо вона є більшою за поріг збігу, здійснити перехід до наступної ітерації.

Описані два способи прискорення виконання алгоритму не впливають на точність результатів реалізації *FSS підходу*. Також можна використати оцінки зверху (властивості 2 та 4) замість точного значення, якщо вони є меншими або рівними порогу збігу. Проте це зменшить точність знаходження команди зі словника, відстань для якої є мінімальною.

5. Результати використання підходу

Запропонований у статті *FSS підхід* реалізований в мобільному застосунку для керування системою «розумний будинок» на основі бібліотеки розпізнавання мовлення від компанії Google. Як міру близькості текстового потоку до команд зі словника використано відстань Левенштейна в модифікації зі статичним порогом збігу. Оскільки метою даного дослідження є збільшення точності розпізнавання команд, варіанти реалізації *FSS підходу*, спрямовані на підвищення значень інших показників ефективності роботи системи голосового керування, при тестуванні не аналізувались.

Тестування проводилось на множині з 50 найбільш поширених голосових команд, отриманій від користувачів голосових систем «розумний будинок».

До реалізації підходу за результатами вимірювання була отримана точність розпізнавання в 92%, тобто 46 вірно розпізнаних команд. Завдяки ж реалізації *FSS підходу* приріст точності розпізнавання склав 4%, досягнувши 96%, або 48 розпізнаних голосових команд. Невірно розпізнаними залишились 2 команди, що не були попередньо додані до словника, натомість в ньому були наявні подібні до них команди. Після виконання корекції алгоритм переходив до виконання існуючих подібних команд, а не визначав відсутність даної команди в словнику. Це могло спричинити завищене значення порогу збігу при тестуванні.

Висновки

Таким чином, в статті показана доцільність розробки нового підходу до корекції помилок в системах голосового керування з необмеженим словником, обґрунтовано місце проведення корекції в загальному процесі виконання команди, запропоновано новий *FSS підхід* до корекції помилок в згаданих системах голосового керування, визначено основні кроки алгоритму, що реалізує *FSS підхід,* та наведено дві можливі його базові модифікації.

Також запропоновано додаткові способи прискорення виконання алгоритму, що реалізовує *FSS підхід*, для забезпечення ефективної роботи системи за критеріями швидкодії та використання пам’яті.

Серед напрямів подальшого вивчення та розвитку запропонованого підходу можна виділити такі: створення програмних бібліотек для популярних мов програмування; розроблення рекомендацій щодо вибору модифікацій підходу та порогу збігу для популярних систем та методів розпізнавання мовлення.

Програмна реалізація підходу виконана мовою C#.

Список літератури: 1. Розпізнавання мовлення. – Режим доступу: https://uk.wikipedia.org/wiki/ Розпізнавання_мовлення. – 2.2.2016. **2.** Furtuna, T. Dynamic Programming Algorithms in Speech Recognition / T. Furtuna. // Informatica Economică. – 2008. – №2. – С. 94. **3.** Распознавание речи от Яндекса. Под капотом у Yandex.SpeechKit. – Режим доступу: https://habrahabr.ru/company/yandex/blog/198556/ – 2.2016. **4.** Zahorian, S.A. Classification for Computer based Visual Feedback for Speech Training for the Hearing Impaired / S.A. Zahorian, A.M. Zimmer, F.M. Vowel // ICSLP 2002. – 2002. **5.** Deep Neural Networks for Acoustic Modeling in Speech Recognition – The shared views of four research groups / G. Hinton, L. Deng, D. Yu and others // IEEE Signal Processing Magazine. – 2012. – №6. – С. 82–97. **6.** Novet, J. Google says its speech recognition technology now has only an 8% word error rate / J. Novet. – Режим доступу: http://goo.gl/. – 09.05.2016. **7.** Zgank, A.Predicting the Acoustic Confusability between Words for a Speech Recognition System using Levenshtein Distance / A. Zgank, Z. Kacic // Elektronika ir Elektrotechnika. – 2012. – №8. – С. 81–84. **8.** Halim, D. Implementation Levenshtein Distance Algorithm For Voice Control In Calorie Tracker Application / D. Halim. – Режим доступу: http://library.umn.ac.id/ eprints/2266/2/abstrakeng.pdf. – 13.04.2016. **9.** Shokhirev, R. Voice control of robots and mobile machinery / R. Shokhirev // Proceedings of the Spring/Summer Young Researchers’ Colloquium on Software Engineering. – 2013. – №7. – С. 155–158. **10.** Відстань Левенштейна. – Режим доступу: https://uk.wikipedia.org/wiki/Відстань_Левенштейна. – 6.2.2016. **11.** Відстань Геммінга. – Режим доступу: https://uk.wikipedia.org/wiki/Відстань_Геммінга. – 6.2.2016. **12.** Ukkonen, E. Finding approximate patterns in strings / E. Ukkonen // Journal of Algorithms. – 1985. – №6. – С. 132-137.

Поступила до редколегії 12.05.2016

УДК 004.934

FSS подход к коррекции ошибок в системах голосового управления с неограниченным словарём / М.М. Шевчук, Я.А. Юсин, Т.Н. Заболотняя // Бионика интеллекта: научн.-техн. журнал. – 2016. – № 1(86). – С. 13-16.

В данной работе предложен новый FSS (fuzzy string search) подход к коррекции ошибок при распознавании команд в системах голосового управления с неограниченным словарем, реализация которого способствует повышению точности работы данных систем. Разработан алгоритм реализации подхода и две его возможные модификации. В соответствии с новым подходом определены способы увеличения эффективности работы систем голосового управления по критериям использования памяти и быстродействия.

Ил.1. Библиогр.: 12 назв.

UDC 004.934

The FSS approach to the error correction in a voice control systems with an unlimited vocabulary / M. Shevchuk, Y. Yusyn, T. Zabolotnia// Bionica Intellecta: Sci. Mag. – 2016. – № 1(86). – P. 13-16.

In this work the new FSS (fuzzy string search) approach to the error correction during commands recognition in a voice control systems with a unlimited vocabulary is proposed. It’s implementation helps to improve the accuracy of these systems' functioning. The algorithm of the approach realization and two it’s possible modifications are developed. The ways of increasing the efficiency of a voice control systems functioning in accordance with the new approach by the criterias of memory and speed are defined.

Fig. 1. Ref.: 12 items.