УДК 004.021: 81'32

Катерина КУЗЬМА
katushke2017@gmail.com
ORCID: 0000-0002-0937-7299
м. Миколаїв

АЛГОРИТМ ПЕРЕВІРКИ ВІДПОВІДІ В СИСТЕМАХ ТЕСТУВАННЯ, ПОДАНОЇ У ТЕКСТОВІЙ ФОРМІ

В роботі досліджено методи перевірки відповіді, поданої у довільній текстовій формі, проаналізовано їх переваги та недоліки, що дозволило визначити найбільш ефективні й перспективні з точки зору практичного застосування в системах тестування. Під час розгляду алгоритмів «точного» порівняння відповіді із вірним зразком (паттерном) здійснено їх аналіз за базовими характеристиками. Відзначені обмеження застосування методів «точного» порівняння рядків в задачах перевірки відповідей, поданих в довільній текстовій формі.

Визначено, що подальшого дослідження потребують саме алгоритми «приблизного» порівняння шляхом обчислення найбільшої загальної підпослідовності двох рядків. За рахунок використання методів динамічного програмування здійснюється зіставлення коротких відповідей, які містять орфографічні помилки, що дозволяє вирішувати задачу перевірки відповіді, поданої у текстовій формі, як задачу «нечіткого» пошуку.

***Ключові слова**: відповідь, подана у текстовій формі, методи порівняння рядків, системи тестування, лінгвістичний аналіз тексту.*

Постановка проблеми

Існуючі системи тестового контролю за своєю функціональністю суттєво обмежують можливості неформалізованої побудови тестових завдань. Сучасні програмні засоби, які використовуються для тестування, дозволяють будувати питання лише певних типів. Переважно це питання типу «одне питання – декілька варіантів відповідей, серед яких один вірний», «одне питання – декілька варіантів відповідей, серед яких декілька вірних», «зіставлення варіантів відповідей».

Недостатньо формалізованим є подання відповіді на питання в довільній текстовій формі.

Таким чином, для автоматизації перевірки відповіді, поданої у текстовому форматі природною мовою, необхідно розробити ефективну методику порівняння такої відповіді зі зразком (зразками) правильної відповіді.

Аналіз останніх досліджень і публікацій

У зв'язку з постійним зростанням потреб у застосуванні механізмів природної мови в інформаційно-атоматизованих системах та людино-машинних системах особливого значення набули питання моделювання природної мови та мовлення. Це призвело до розроблення різноманітних лінгвістичних моделей, що могли б розв'язати практичні завдання лінгвістики, а саме: інформаційний пошук, машинний переклад, розуміння природної мови тощо.

Дослідженнями, розробкою моделей та методів лінгвістичного аналізу тексту займалися науковці: Комарницька О.І., Лесько О.М., Палагін О.В., Катеринчук І.С. та інші [1-4].

Постановка завдання

Метою роботи є аналіз методів перевірки відповіді, поданої у довільній текстовій формі, для виявлення їх переваг й недоліків під час використання в системах тестування.

Виклад основного матеріалу

Дослідження й опис природних мов у контексті автоматизованих інформаційних систем вимагає застосування математичних методів, серед яких: комбінаторні методи, методи математичної статистики, булевої алгебри, теорія графів, теорія нечітких множин; теорія ймовірності, методи штучного інтелекту (зокрема, нейромережі).

Розглядаючи алгоритми порівняння рядків тексту, здійснено їх класифікацію на два основних види: «точного» порівняння із зразком (паттерном), «приблизного» порівняння з «паттерном». При цьому шаблон (паттерн) пошуку може бути одиночним або множинним.

Зокрема в роботі [1] Комарницькою О.І. розроблено метод нечіткого семантичного порівняння, який базується на алгоритмі «приблизного порівняння» з множинними паттернами. Передбачається автоматизоване визначення лексичних одиниць тексту з подальшим здійсненням морфологічного, синтаксичного, семантичного та прагматичного аналізу. Метод, розроблений в роботі [1] для розпізнавання та виправлення слів, написаних з помилками (вставка, заміна, видалення, транспозиція), базується на удосконаленні метрики Левенштейна. Такий підхід передбачає, що чим більшою є відстань між рядками, тим більшою є відмінність.

Таким чином, в класичних підходах для порівняння рядків застосовуються метрики, які оцінюють мінімальну кількість дій (операцій редагування), необхідних для перетворення одного рядка в інший.

Згідно роботи [5] функція Левенштейна – це міра різниці двох послідовностей символів (рядків) відносно мінімального числа елементарних операцій редагування, необхідних для переведення одного рядка в інший у випадку, коли операції мають однакову вагу. Існує також модифікація відстані Левенштейна – відстань Левенштейна-Дамерау, де до безлічі елементарних операцій включено транспозиції символів. При цьому вимагається, щоб до транспонованих символів не застосовувалися інші операції редагування.

Якщо призначити одиничну вагу видаленню й вставці та подвоєну вагу заміні, отримаємо «відстань редагування». Якщо дозволяються тільки операції заміни із одиничною вагою, то розглядається відстань Хеммінга, яка визначається як кількість позицій, в яких рядки містять різні символи. Вона придатна для визначення відстані лише в тих випадках, коли рядки,які порівнюються, мають однакову довжину.

У разі, коли дозволені тільки операції видалення й вставки з вагою, рівною одиниці, можливо обчислити міру, яку називають найбільшою загальною підпослідовністю двох рядків (LCS – Longest Common Subsequence).

Серед інших підходів можна виділити алгоритми перевірки схожості звучання слів за допомогою фонетичного кодування (Soundex, Metaphone, NYSIIS і ін.) [6]. Зазвичай такі алгоритми мовнозалежні та погано працюють у разі, коли рядки різняться в першому символі або містять пробіли.

Ряд підходів також заснований на зіставленні лексем (схожість Джаккарда та ін.). У них робота ведеться з векторною моделлю документів, а текст представляється у вигляді набору слів. У деяких випадках замість слів в якості лексем виступають *n*-грами (загальні підрядки фіксованої довжини *n*). Основним недоліком цих методів, як правило, є низька ефективність роботи при порівнянні коротких рядків або при наявності орфографічних помилок в словах [7].

Відстань Левенштейна може бути обчислена за допомогою методу динамічного програмування Вагнера-Фішера [8]. Ідея методу полягає в тому, щоб послідовно оцінювати відстані між подовженими на кожному кроці префіксами рядків до отримання остаточного результату. Проміжні результати обчислюються ітеративно та зберігаються в масиві розмірності $(m+1)\cdot(n+1)$, що призводить до витрат часу та пам'яті $Q(m\times n)$, де m та n – довжини порівнюваних рядків. Для знаходження значення відстані потрібно обчислити $m\times n$ елементів матриці за допомогою методу динамічного програмування.

На основі методу динамічного програмування було розроблено багато алгоритмів, зокрема алгоритми Хешберга, Ханта-Шиманського, Вагнера-Фішера, Укконена-Майерса та інші. Більш докладний опис та дослідження цих алгоритмів можна знайти в роботах [8, 9].

Представлений далі алгоритм порівняння рядків призначений для зіставлення коротких рядків при наявності орфографічних помилок в словах (тестові питання в системі тестування, які відносяться до типу: «питання-коротка відповідь»).

Для обчислення довжини найбільшої загальної підпослідовності двох рядків для співставлення відповіді з еталоном була обрана одна з модифікацій методу динамічного програмування, запропонована Хешбергом. Вибір даного методу був обумовлений достатньою ефективністю та відносною простотою реалізації.

Втрати алгоритму щодо пам'яті та часу обчислення складають відповідно $Q(m+n)$ та $Q(m\times n)$, де m й n – довжини порівнюваних рядків. Алгоритм базується на рекурсії, при цьому на кожному кроці визначаються довжини найбільших спільних підпослідовностей у все більше подовжених префіксів рядків. Позначимо їх як $l[i,j]$:

$$l[i,j]=\left|LCS\left(x[1..i],y[1..j]\right)\right|.$$

Функція LCS (x, y) підраховує найбільшу загальну підпослідовність рядків x та y відповідно. Так як довжина найбільшої загальної підпослідовності будь-якого рядка та порожнього рядка дорівнює нулю, значення меж масиву задаються як $l[i,0]=l[0,j]=0$. У позиції $[i,j]$, тобто коли розглядаються префікси $x[1,i]$ та $y[1,j]$, якщо $x_i=y_j$, отримуємо нове значення функції LCS, додаючи даний символ до поточного значення LCS префіксів $x[1,i-1]$ та $y[1,j-1]$, звідки

$$l[i,j]=l(i-1,j-1)+1.$$

Інакше поточне значення LCS обчислюється як максимум із попередніх сусідніх значень:

$$l[i,j]=\max\{l[i-1,j],l[i,j-1]\}.$$

Зауважимо, що для обчислення рядка i потрібен тільки рядок $i-1$. Для зручності введемо вектор $ll[j]=l[m,j]$. Використовується масив h довжини $2(n+1)$, в якому нульовий та перший рядки виступають в якості рядків $i-1$ та i масиву l відповідно.

Граничні умови за j від 0 до n задаються як $h[1,j]=0$. Перед обчисленням кожного нового «рядка i» перший рядок зсувається вгору на місце нульового рядка. Для цього використовується цикл за i від 1 до m та за j від 0 до n, в якому $h[0,j]$ присвоюється значення $h[1,j]$. У циклі за j від 1 до n в позиції $[i,j]$ при $x_i=y_j$ вважаємо $h[1,j]=h[0,1]+1$. У іншому випадку

$$h[1,j]=\max\{h[1,j-1],h(0,j)\}.$$

На останньому етапі за всіма j від 0 до n відбувається копіювання результату $h[1,j]$ у вихідний вектор $ll[j]$.

Граничний показник подібності відповіді з еталонною (правильною) необхідно підбирати в процесі тестування алгоритму (діапазон значень від 50 % до 100 %). Використання алгоритму направлено на відповіді, представлені в короткій формі (до 10 слів) з можливими орфографічними помилками.

Висновки і перспективи досліджень

Таким чином, враховуючи обмеження «точних» методів в задачах перевірки відповідей, представлених у довільній текстовій формі, подальшого дослідження потребують методи «нечіткого» порівняння, які базуються на використанні методів динамічного програмування для визначення найбільшої загальної підпослідовності двох рядків (LCS).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Комарницька, О.І. Алгоритм нечіткого семантичного порівняння текстової інформації [Текст] / О.І. Комарницька, Т.В. Ваколюк // Збірник наукових праць Військового інституту Київського Національного університету ім. Т. Шевченка. – К., 2013. – № 39. – С. 163-168.
2. Лесько, О.М. Использование онтологий для анализа семантики естественно-языковых текстов [Текст] / О.М. Лесько, Ю.В. Рогушина // Проблеми програмування. – 2009. – № 3. – С. 59-65.

3. Палагин, А.В. К проектированию онтологоуправляемой информационной системы с обработкой естественно-языковых объектов [Текст] / А.В. Палагин, Н.Г. Петренко // Математичні машини і системи. – 2008. – № 2. – С. 14-23.
4. Катеринчук, І.С. Інтелектуальна система автоматизованого контролю знань студентів вищих навчальних закладів [Текст] / І.С. Катеринчук, Р.В. Рачок, В.В. Кравчук, В.М. Кулик // Інформаційні технології в освіті: збірник наукових праць. – 2009. – Вип. 4. – Херсон: Вид-во ХДУ. – С. 139-147.
5. Stephen Graham A. String Searching Algorithms [Text] / Graham A. Stephen // Lecture Notes Series On Computing. – Vol.3. – London: World Scientific, 1994. – 256 p.
6. Soundex [Електронний ресурс]. – Режим доступу до ресурсу: http://en.wikipedia.org/wiki/Soundex
7. Цыганов, Н.Л. Исследование методов поиска дубликатов веб-документов с учетом запроса пользователя [Текст] / Н.Л. Цыганов, М.А. Циканин // Интернет-математика 2007: Сб. работ участников конкурса. – 2007. – Екатеринбург: Изд-во Урал. ун-та. – С. 211-222.
8. Смит, Б. Методы и алгоритмы вычислений на строках: пер. с англ [Текст] / Б. Смит. – Москва: ООО «И.Д. Вильямс», 2006. – 496 с.
9. Navarro, G. A guided tour to approximate string matching [Text] / G. Navarro // ACM Computing Surveys. – 2001. – 33(1). – P. 31-88.

Kateryna KUZMA
Mykolaiv

THE ALGORITHM OF VERIFICATION THE ANSWER IN TESTING SYSTEMS, SUBMITTED IN A TEXT FORM

The methods of checking the answer submitted in an text form, their advantages and disadvantages, which allowed to determine the most effective and promising for practical use in testing systems have been considered. While investigating algorithms of «strict» comparison the answer with a correct pattern, their basic characteristics have been analyzed. Restrictions in use the «strict» comparison strings methods in tasks of verification the answers submitted in an text form have been noted.

It was determined that further researching requires the algorithms of «approximate» comparison with multiple patterns by calculating the longest common subsequence of two strings. By using the dynamic programming methods, the comparison of short answers is carried out, that allowed to solve the problem of checking the answers, presented in textual form, as a problem of «fuzzy» search.

Keywords: *the answer submitted in text form, methods of the string comparison, testing system, linguistic analysis of text.*

Екатерина КУЗЬМА
Николаев

АЛГОРИТМ ПРОВЕРКИ ОТВЕТА В СИСТЕМАХ ТЕСТИРОВАНИЯ, ПРЕДСТАВЛЕННОГО В ТЕКСТОВОЙ ФОРМЕ

В работе исследованы методы проверки ответа, представленного в произвольной текстовой форме, проанализированы их преимущества и недостатки, что позволило определить наиболее эффективные и перспективные с точки зрения практического применения в системах тестирования. При рассмотрении алгоритмов «точного» сравнения ответа с правильным образцом (паттерном) осуществлен их анализ по базовым характеристикам. Отмечены ограничения применения методов «точного» сравнения строк в задачах проверки ответов, представленных в произвольной текстовой форме.

Определено, что дальнейшего исследования требуют именно алгоритмы «приблизительного» сравнение путем вычисления наибольшей общей подпоследовательности двух строк. За счет использования методов динамического программирования осуществляется сопоставление коротких ответов, что позволяет решать задачу проверки ответа, представленного в текстовой форме, как задачу «нечеткого» поиска.

Ключевые слова: *ответ, представленный в текстовой форме, методы сравнения строк, системы тестирования, лингвистический анализ текста.*

Стаття надійшла до редколегії 23.10.2018