

Минобрнауки России
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Сыктывкарский государственный университет имени Питирима Сорокина»
(ФГБОУ ВО «СГУ им. Питирима Сорокина»)
Институт точных наук и информационных технологий
Кафедра математического моделирования и кибернетики

Алгоритмы нечеткого сравнения строк

Выполнил:
студент гр. 1200 К.С. Петренко

Сыктывкар 2018

Оглавление

Редакционные расстояния	3
Расстояние Хэмминга	3
Расстояние Левенштейна.....	3
Применение редакционного расстояния для проверки ответов студентов	5
Алгоритм проверки ответов студентов с применением редакционного расстояния.....	5
Ограничения для входных данных	7
Использование программы для проверки заданий студентов	7
Заключение	11

Редакционные расстояния

Редакционным расстоянием для двух строк является минимальное количество операций вставки, удаления или замены одного символа на другой, необходимых для превращения одной строки в другую.

Расстояние Хэмминга

Расстояние Хемминга применяется для строк одинаковой длины любых q -ичных алфавитов и служит метрикой различия (функцией, определяющей расстояние в метрическом пространстве) объектов одинаковой размерности.

Так как данное расстояние сравнивает строки только одинаковой длины то не учитываются варианты, когда пропущена буква или вместо одной записано 2 и более. То есть данное расстояние имеет узкую направленность.

Так же можно заметить, что для нахождения данного расстояния будет достаточно провести посимвольное сравнение двух строк и увеличивать счетчик, если символы будут не совпадать. Конечное значение счетчика и будет расстоянием Хэмминга для данных слов.

Расстояние Левенштейна

Для того чтобы получить расстояние Левенштейна для строк s и t (длиной m и n соответственно, индексация начинается с нуля) и редакционное предписание (какие именно правки нужно вносить), рассчитывается матрица расстояний D (размерность $m+1 * n+1$). Каждый элемент $D[i, j]$ содержит дистанцию между первыми i символами строки s и первыми j символами строки t .

Матрица дистанций Левенштейна для строк $s='ABC'$ и $t='ABF'$ (добавлены последние символы подстрок, чтобы соответствие было явно видно):

$$\begin{bmatrix}
 & A & B & F \\
 & 0 & 1 & 2 & 3 \\
 A & 1 & 0 & 1 & 2 \\
 B & 2 & 1 & 0 & 1 \\
 C & 3 & 2 & 1 & 1
 \end{bmatrix}$$

Рис 1. Матрица дистанций Левенштейна для строк «ABC» и «ABF»

Столбцы соответствуют подстрокам строки t , а строки матрицы — подстрокам s . Строка и столбец с нулевым индексом соответствуют пустым подстрокам s и t . Каждый элемент этой матрицы содержит расстояние между подстроками соответствующих его индексам. Например, $D[3,2] = 1$ это расстояние между ABC и AB (всего одна правка — удалить C). Таким образом, $D[3,3] = 1$ это и есть искомая дистанция между ABC и ABF. (замена C на F).

Кроме дистанции эта матрица содержит в себе информацию о тех правках, которые необходимо внести в строку s чтобы получить строку t - редакционное предписание.

Чтобы построить редакционное предписание, проще всего запоминать выбранные операции во время построения матрицы. А также, можно проанализировать матрицу, двигаясь от правого нижнего угла к левому верхнему по минимальным весам и запоминая ходы: ход влево — I(вставка), вверх — D(удаление), влево-вверх — R(замена), если символы различаются, иначе M(совпадение).

M M R
A B C
A B F

Рис 2. Редакционное предписание для слов «ABC» и «ABF»

Применение редакционного расстояния для проверки ответов студентов

Для проверки ответов, необходимо проверить каким будет редакционное расстояние для пар слов $[A[i], B[j]]$, где $A[i]$, слово из ответа студента, а $B[j]$ - ключевое слово.

Алгоритм проверки ответов студентов с применением редакционного расстояния

Входными данными для программы являются:

1. Файл с ключевыми словами. Данный файл имеет n -строк, где n - количество вопросов в проверочной работе.
2. Файл с ответом студента. Также имеет n -строк. В каждой строке ответ на вопрос.
3. Порог редакционного расстояния при котором слово из ответа студента будет считаться программой правильным. Назовем переменную которая будет хранить это значение как `value`.

Считываем значения из каждого файла по строкам, строки разбиваем на слова, используя пробел как разделитель, и записываем слова в двумерные массивы. Где первый индекс это номер вопроса, а второй номер слова в строке ключей или ответов к вопросу.

```
q.open("key.txt");
string key_line;
while(std::getline(q, key_line))
{
    keys[c1] = key_line;
    c1++;
}
q.close();
```

Рис 3. Считывание строк ключевых слов для вопроса

```
for (i = 0; i < c1; i++){
    j = 0;
    while(keys[i].find(' ') != string::npos){
        key_words[i][j] = keys[i].substr(0,keys[i].find(' '));
        keys[i].erase(0,keys[i].find(' ')+1);
        j++;
    }
    key_words[i][j] = keys[i];
}
```

Рис 4. Разбиение строк ключей на отдельные слова и запись в двумерный массив

После этого мы имеем 2 двумерных массива, один с ключевыми словами, второй с ответом студента.

Далее обходим массив ключевых слов в двойном цикле и для каждого $K[i][j]$ – значения из массива ключевых слов, проходимся по массиву $A[i]$.

Для каждого $A[i][p]$ - слова из ответа студента на вопрос под индексом i , находим редакционное расстояние для слов $K[i][j]$ и $A[i][p]$ - R . Если R меньше либо равно $value$, увеличиваем счетчик правильных слов для ответа $valid_word$.

После того как для ключевого слова $K[i][j]$ и всех значений массива слов из ответа $A[i]$ найдены расстояния, проверяем значение $valid_word$.

Обрабатываются 2 случая:

1. $valid_word$ больше нуля, значит $K[i][j]$ при данном $value$ содержится в ответе, увеличиваем счетчик количества присутствующих ключевых слов в ответе $valid_count$.
2. $valid_word$ меньше или равен нулю, значит $K[i][j]$ при данном $value$ не содержится в ответе, ничего не делаем.

Переходим к следующему ключевому слову.

После того как мы прошли по всем ключевым словам для вопроса, выводим процентное соотношение $(valid_count/j)*100$ присутствующих ключевых слов и увеличиваем счетчик i отвечающий за номер вопроса.

```

while(key_words[i][j].length() != 0)
{
    while(key_words[i][j].length() != 0)
    {
        valid_word = 0;
        while (answer_words[i][p].length() != 0)
        {
            if(levensteinInstruction(answer_words[i][p], key_words[i][j], 1, 1, 1) <= value){valid_word ++;};
            p++;
        }
        if(valid_word > 0){valid_count++;}
        j++;
        p=0;
    }
    cout<<i+1<<" ". "<<(double)valid_count/j*100<<"% ключевых слов найдено в ответе"<<endl;
    i++;
    j = 0;
    valid_count = 0;
}

```

Рис 5. Обход ключевых слов и ответов студента

Ограничения для входных данных

Ключевые слова должны удовлетворять некоторым характеристикам чтобы избежать некорректной работы программы:

1. Должны быть достаточной длины, чтобы не спутать ключевое слово с вспомогательными частями речи
2. Ключевых слов для ответа должно быть достаточное количество, чем больше ключевых слов, тем более точную характеристику ответа мы получим
3. Ключевые слова должны быть уникальны в рамках одного вопроса

Исходя из данных полученных Макаровым Андреем в ходе его ВКР, будем использовать редакционное расстояние меньшее или равное 3.

Использование программы для проверки заданий студентов

В ходе исследования была проведена проверочная работа для студентов группы 111пми Сыктывкарского Государственного Университета им. Питирима Сорокина.

Цели тестирования:

1. Получение статистических данных для отладки программы
2. Проверка знаний студентов в ходе педагогической практики
3. Поиск критических уязвимостей и их устранение

Заданиями для проверочной работы были выбраны вопросы по основам информатики и программирования.

Вопрос	Правильный ответ	Ключевые слова
1.Понятие алгоритма	Слово алгоритм сравнительно новое Оно произошло от algorism которое использовалось для обозначения выполнения арифметических операций в позиционной десятичной системе счисления	алгоритм арифметическая операция позиционная десятичная система счисления
2.Какие бывают алгоритмические конструкции	Любой алгоритм может быть представлен комбинацией трех базовых структур следование ветвление цикл Характерной особенностью базовых структур является наличие в них одного входа и одного выхода	базовые структуры следование ветвление цикл вход выход
3.Какие компоненты Фон Нейман выделил и детально описал для вычислительной системы	Центральное арифметико-логическое устройство Центральное устройство управления Запоминающее устройство Устройство ввода данных Устройство вывода данных	арифметико-логическое устройство управление запоминающее ввод данные вывод
4.Какие положения фон Нейман зафиксировал для вычислительной системы	ВС должна быть электронным устройством работать с двоичными числами выполнять операции последовательно Программа выполнения этих операций должна храниться в памяти ВС совместно с данными	ВС электронное устройство двоичные числа операции последовательно программа хранить данные
5.Сколько было поколений ЭВМ	Всего на данный момент времени пять поколений эвм	Пять
6.Что такое машинный язык	Множество всех машинных	

	команд называется языком машины	последовательность команд множество языков машины
7.Что такое глобальная переменная	Глобальная переменная это переменная к которой можно обратиться в любой точке тела программы	тело программа переменная глобальная доступ
8.Что такое локальная переменная	Локальная переменная это переменная которая определяется в каком либо цикле функции или процедуре и может быть использована только там	цикл функции процедура использована только там
9.Что такое рекурсия	Под рекурсией понимается такой способ решения при котором программа вызывает сама себя	программа вызывает себя
10.Что такое бит	Бит это минимальная единица информации	минимальная единица информации
11.На каком языке программирования пишутся макросы для excel	Visual basic	Visual basic
12.Чем известен Евклид	Евклид является отцом геометрии	отец геометрии
13.Какими трудами известен Кнут	автор классических трудов Искусство программирования Конкретная математика концепции грамотное программирование	Искусство программирования Конкретная математика грамотное программирование
14.Чем известен Эдсгер Вие Дейкстра	создатель алгоритма Семафоров Дейкстры один из основателей структурного программирования	семафоры структурное программирование
15.Опишите как работает цикл while	Цикл работает до того момента пока условие верно	условие верно

16. Чем характерно первое поколение ЭВМ	Работало на лампах	лампа
17. Чем характерно второе поколение ЭВМ	Работало на транзисторах	транзистор
18. Чем характерно третье поколение ЭВМ	Работало на интегральных микросхемах	Интегральная микросхема
19. Чем характерно четвертое поколение ЭВМ	Значительное снижение стоимости и размеров эвм	снижение стоимость размер эвм
20. Чем характерно пятое поколение ЭВМ	Работает на микропроцессорах	микропроцессор

Таблица 1. Вопросы, правильные ответы и ключевые слова для тестирования студентов

В проверочной работе поучаствовали 9 студентов. Правильность ответа выводится в процентах.

Вопрос	1	2	3	4	5	6	7	8	9	Среднее значение
1.	0,00	28,57	0	14,28	0	0	28,57	0	14,28	9,52
2.	42,85	42,85	42,85	0	28,57	0	0	0	28,57	20,63
3.	71,42	0	0	0	57,14	14,28	28,57	0	14,28	20,63
4.	80,00	0	0	0	60	10	10	0	0	17,78
5.	100	0	100	100	100	0	100	100	0	66,67
6.	60	40	40	40	40	20	0	20	0	28,89
7.	60	60	80	60	80	80	0	0	0	46,67
8.	66,66	16,67	50	50	66,66	83,33	0	0	0	37,04
9.	66,66	33,33	33,33	33,33	33,33	33,33	0	33,33	0	29,63
10.	100	33,33	66,66	66,66	100	66,66	0	66,66	0	55,55
11.	0,00	0	0	0	100	0	0	0	0	11,11

12.	50	50	0	0	0	0	0	50	0	16,67
13.	50	0	0	33,33	50	0	0	0	0	14,81
14.	0	0	0	0	0	0	0	0	0	0,00
15.	50	50	100	0	50	100	0	50	0	44,44
16.	100	0	100	0	100	0	0	0	0	33,33
17.	100	0	100	0	0	0	0	0	0	22,22
18.	50	0	0	0	50	0	0	0	0	11,11
19.	0	0	0	0	0	0	0	0	0	0,00
20.	100	0	0	0	0	0	0	0	0	11,11
Оценка за тест в процентах	57,38	17,74	35,64	19,88	45,79	20,38	8,36	16,00	2,86	

Таблица 2. Процентное соотношение правильности ответов студентов на вопросы тестирования

Исходя из полученных данных, можно заметить, что наилучший результат среди всех студентов показал вопрос номер 10. На него ответили 7 из 9 студентов, с средним процентом правильности 55,55%. Наихудшим вопросом стал, вопрос номер 19. На него никто не смог ответить, за данный промежуток времени.

Заключение

Использование, какого то конкретного метода не даст полной уверенности в правильности ответа студента. Решением данной проблемы может стать комбинирование методов. Однако даже в этом случае остается необходимость ручной проверки сомнительных результатов. Также встает вопрос в выборе вопроса, чтобы ответ был как можно однозначнее, для корректного выбора ключевых слов.

Список литературы

1. Вирт Н. Алгоритмы + структуры данных = программы // Издательство М. Мир 1985г. 406с.
2. Выявление плагиата [Электронный ресурс], URL: https://ru.wikipedia.org/wiki/Выявление_плагиата (дата обращения: 16.12.2016)
3. Юрий Жиловец @gatoazul Система поиска плагиата [Электронный ресурс], URL: <https://habrahabr.ru/post/199190/> (дата обращения: 16.12.2016)
4. Зеленков Ю.Г., Сегалович И.В. Сравнительный анализ методов определения нечетких дубликатов для WEB-документов // Труды 9-ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» RCDL'2012: сб. работ участников конкурса – Переславль-Залесский, 2012.
5. Москаленко Е.Н., Слесарев Ю.Н. Методы проверки текстовых документов на уникальность // Современные научные исследования и инновации. 2016. № 6 [Электронный ресурс]. URL: <http://web.snauka.ru/issues/2016/06/69137> (дата обращения: 1.06.2017).
6. Алгоритмы поиска в строке [Электронный ресурс], URL: <https://habrahabr.ru/post/111449/> (дата обращения 1.06.2017)
7. Расстояние Хэмминга [Электронный ресурс], URL: https://ru.wikipedia.org/wiki/Расстояние_Хэмминга (дата обращения 1.07.2017)
8. Расстояние Левенштейна [Электронный ресурс], URL: https://ru.wikipedia.org/wiki/Расстояние_Левенштейна (дата обращения 1.07.2017)
9. Расстояние Дameraу — Левенштейна [Электронный ресурс], URL: [https://ru.wikipedia.org/wiki/ Расстояние_Дameraу_ —_ Левенштейна](https://ru.wikipedia.org/wiki/Расстояние_Дameraу_—_Левенштейна) (дата обращения 1.07.2017)

- 10.Алгоритм Вагнера-Фишера [Электронный ресурс], URL:
<http://algotlist.manual.ru/search/lcs/vagner.php>(дата обращения 1.07.2017)
- 11.Желудков А. В., Макаров Д. В., Фадеев П. В. Особенности алгоритмов нечёткого поиска // Инженерный вестник. 2014. № 12. С. 501-511
- 12.Андреев А.М. Автоматизация обнаружения и исправления опечаток в названиях географических объектов для системы семантического контроля документов электронной библиотеки / А.М. Андреев, Д.В. Березкин, А.С. Нечкин, К.В. Симаков, Ю.Л. Шаров // НПЦ «Интеллект плюс». – 2007. – № 25
- 13.Бирюкова М.В. Определение опечаток в ответе студента на тестовый вопрос при известном правильном ответе / Бирюкова М.В., Мамонтов Д.П., Сычев О.А. // Открытое образование. - 2015. - № 5. - С. 11-15.