|  |  |  |
| --- | --- | --- |
|  | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
| Лабораторная работа № 1 | | |
| по дисциплине «Программные системы статистического анализа» | | |
| **.** | | |
|  | | |
|  | Вариант: 52 | ПМИМ-51 БуданцеВ дмитрий |
| . |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Преподаватели | Тимофеева Анастасия Юрьевна |
|  |  |
| Новосибирск,2025 | | |

1. Постановка задачи

Цель

Сравнение метрик, реализованных для вычисления расстояния между строками в stringdist.

Задачи

**1**. **Часть 1**: Базовое сравнение на простых примерах

**2**. **Часть 2**: Анализ чувствительности к разным типам различий

1. Опечатки

2. Разная длина строк

3. Перестановки символов

4. Разный регистр символов

**3**. **Часть 3**: Производительность метрик

1. Теоретическая часть
2. Введение

Метрики расстояния меся фундаментальным инструментом в обработке естественного языка, анализе текстов и очистке данных. Они позволяют количественно оценить степень различия между двумя строковыми последовательностями, что особенно важно для задач поиска похожих строк, исправления опечаток и обнаружение дубликатов.

1. Расстояние Левенштейна
2. Определение

Расстояние Левенштейна - это минимальное количество операций редактирования, необходимых для преобразования одной строки в другую.

Формальное определение:

Для двух строк и длиной и соответственно, расстояние Левенштейна вычисляется по формуле:

где:

– расстояние между первыми  символами строки  и первыми  символами строки

– индикаторная функция, равная 0 если символы совпадают, и 1 в противном случае.

1. Операции редактирования

Расстояние Левенштейна учитывает три базовые операции:

1. **Вставка** - добавление символа

- Пример: "кот" -> "кот**а**"

2. **Удаление** - удаление символа

- Пример: "кот**а**" -> "кот"

3. **Замена** - замена одного символа на другой

- Пример: "кот" -> "к**а**т"

1. Пример вычисления

Для строк "kitten" и "sitting":

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | S | I | T | T | I | N | G |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| K | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| I | 2 | 2 | 1 | 2 | 3 | 4 | 5 | 6 |
| T | 3 | 3 | 2 | 1 | 2 | 3 | 4 | 5 |
| T | 4 | 4 | 3 | 2 | 1 | 2 | 3 | 4 |
| E | 5 | 5 | 4 | 3 | 2 | 2 | 3 | 4 |
| N | 6 | 6 | 5 | 4 | 3 | 3 | 2 | **3** |

Расстояние: 3

1. Расстояние Дамерау-Левенштейна
2. Определение

Расстояние *Дамерау-Левенштейна* является расширением классического расстояния Левенштейна, добавляя четвёртую операцию - **транспозицию** (перестановку двух соседних символов)

Формальное определение:

Для двух строк и длиной m и n соответственно, расстояние Левенштейна вычисляется по формуле:

1. Дополнительная операция

4. Транспозиция - перестановка двух соседних символов

- Пример: "**ab**cd" -> "**ba**cd"

1. Пример вычисления

Для строк "ca" и "abc"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | a | b | c |
|  | 0 | 1 | 2 | 3 |
| c | 1 | 1 | 2 | 2 |
| a | 2 | 1 | 2 | **3** |

Расстояние: 3

Примечание: Если попробовать получить результат от **stringdist**, то результат будет отличаться. Дело в том, что данный алгоритм реализован, через оптимальное расстояние выравнивания струны, а не через расстояние с соседними транспозициями из-за чего значение отличается от истинного.

1. Расстояние Хэмминга
2. Определение

Для двух строк одинаковой длины расстояние Хэмминга равно количеству позиции, в которых соответствующие символы различаются.

где – индикаторная функция, равная 1 если символы различны, и 0 в противном случае.

1. Расстояние Джакарда
2. Определение

Строки представляются как множества n-грамм (обычно биграмм и триграмм).

Формула коэффициента Джакарда:

Расстояние Джакарда:

где:

- множества n-грамм строк

- мощность пересечения множеств

- мощность объединения множеств

1. Пример вычисления

Для строк и с триграммами :

триграммы:

триграммы:

1. Параметр q и его влияние

Параметр определяет размер n-грамм:

Малые (1-2): лучше для коротких строк, чувствительно к опечаткам

Большие (3-4): лучше для длинных текстов, устойчивее к шуму

1. Косинусное расстояние
2. Определение

**Косинусное расстояние** основано на концепции векторного пространства и измеряет угол между векторами, представляющими строки.

Математическая формула:

где:

- – скалярное произведение векторов

- – нормы векторов

- – угол между векторами

1. Векторное представление строк

Строки представляются как векторы в пространстве n-грамм:

Каждая dimension соответствует уникальной n-грамме

Значения – частоты или бинарные индикаторы наличия n-грамм

Пример для "hello" (биграммы):

1. Q-граммное расстояние
2. Определение

**Q-граммное расстояние** измеряет разность между множествами n-грамм двух строк.

Формула:

где – множество q-грамм строки .

1. Расстояние Jaro
2. Алгоритм вычисления

**Шаг 1**: Поиск соответствующих символов

- Символы считаются соответствующими, если они одинаковы и находятся в пределах позиций

**Шаг 2**: Подсчет транспозиций

- Транспозиции - это соответствующие символы в разном порядке

**Формула Jaro**:

где:

* + – количество совпадающих символов
  + – половина количества транспозиций

1. Расстояние Jaro-Winkler
2. Формула вычисления

где:

* – расстояние **Jaro**
* – длина общего префикса (максимум 4 символа)
* – фактор масштаба (обычно 0.1)

1. Расстояние LCS
2. Определение

**Расстояние на основе самой длинной общей подстроки**(Longest Common Substring distance) – это метрика, которая измеряет различие между двумя строками на основе длины их наибольшей общей непрерывной подпоследовательности.

**Формальное определение:**

Для двух строк и , расстояние вычисляется как:

где:

- – длины строк и

-  – длина самой длинной общей подстроки

1. Пример вычисления

Для строк и :

**Шаг 1:** Поиск LCS

- Общие подстроки: (длина 2), (длина 2)

- Longest Common Substring: или (длина 2)

**Шаг 2:** Вычисление расстояния

В **stringdist** другая интерпретация, поэтому результат другой.

1. Практическая часть

Часть 1: Базовое сравнение на простых примерах

**Таблица 1.** Результаты сравнения метрик расстояния

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Pair** | **L** | **DL** | **H** | **J Q2** | **J Q3** | **CosQ2** | **CosQ3** | **QGQ2** | **QGQ3** | **Jaro** | **JaroW** | **LCS** |
| **"kitten" "sitting"** | 3 | 3 | NA | 0.778 | 0.875 | 0.635 | 0.776 | 7 | 7 | 0.254 | 0.254 | 5 |
| **"hello" "hell"** | 1 | 1 | NA | 0.25 | 0.333 | 0.134 | 0.184 | 1 | 1 | 0.067 | 0.04 | 1 |
| **"abc" "acb"** | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 4 | 2 | 0.444 | 0.4 | 2 |
| **"book" "back"** | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 6 | 4 | 0.333 | 0.3 | 4 |
| **"data" "date"** | 1 | 1 | 1 | 0.5 | 0.667 | 0.333 | 0.5 | 2 | 2 | 0.167 | 0.117 | 2 |
| **"martha" "marhta"** | 2 | 1 | 2 | 0.75 | 0.857 | 0.6 | 0.75 | 6 | 6 | 0.056 | 0.039 | 2 |
| **"example" "samples"** | 3 | 3 | 7 | 0.5 | 0.571 | 0.333 | 0.4 | 4 | 4 | 0.19 | 0.19 | 4 |
| **"programming" "program"** | 4 | 4 | NA | 0.4 | 0.444 | 0.225 | 0.255 | 4 | 4 | 0.121 | 0.073 | 4 |
| **"identical" "identical"** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **"completely" "different"** | 8 | 8 | NA | 1 | 1 | 1 | 1 | 17 | 15 | 0.567 | 0.567 | 15 |

**Таблица 2.** Матрица корреляции между метриками

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **L** | **DL** | **H** | **J Q2** | **J Q3** | **CosQ2** | **CosQ3** | **QGQ2** | **QGQ3** | **Jaro** | **JaroW** | **LCS** |
| **L** | 1 | 0.875 | 0.853 | 0.638 | 0.652 | 0.526 | 0.558 | 0.773 | 0.739 | 0.488 | 0.530 | 0.857 |
| **DL** | 0.875 | 1 | 0.906 | 0.383 | 0.400 | 0.279 | 0.291 | 0.552 | 0.554 | 0.368 | 0.417 | 0.943 |
| **H** | 0.853 | 0.906 | 1 | 0.163 | 0.181 | 0.050 | 0.060 | 0.377 | 0.472 | 0.192 | 0.261 | 0.731 |

Продолжение таблицы

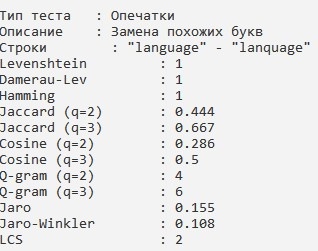
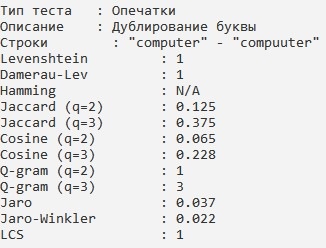
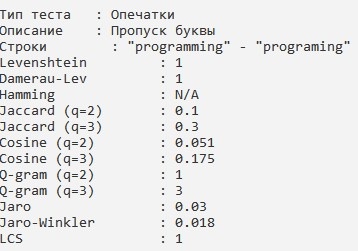
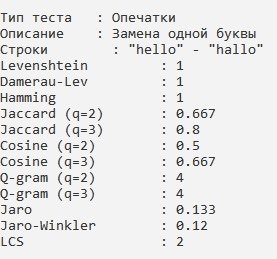
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **L** | **DL** | **H** | **J Q2** | **J Q3** | **CosQ2** | **CosQ3** | **QGQ2** | **QGQ3** | **Jaro** | **JaroW** | **LCS** |
| **J Q2** | 0.638 | 0.383 | 0.163 | 1 | 0.983 | 0.976 | 0.995 | 0.846 | 0.566 | 0.802 | 0.783 | 0.613 |
| **J Q3** | 0.652 | 0.400 | 0.181 | 0.983 | 1 | 0.923 | 0.976 | 0.850 | 0.626 | 0.739 | 0.708 | 0.626 |
| **CosQ2** | 0.526 | 0.279 | 0.050 | 0.976 | 0.923 | 1 | 0.981 | 0.773 | 0.428 | 0.842 | 0.828 | 0.523 |
| **CosQ3** | 0.558 | 0.291 | 0.060 | 0.995 | 0.976 | 0.981 | 1 | 0.820 | 0.529 | 0.786 | 0.760 | 0.542 |
| **QGQ2** | 0.773 | 0.552 | 0.377 | 0.846 | 0.850 | 0.773 | 0.820 | 1 | 0.897 | 0.431 | 0.440 | 0.720 |
| **QGQ3** | 0.739 | 0.554 | 0.472 | 0.566 | 0.626 | 0.428 | 0.529 | 0.897 | 1 | 0.045 | 0.056 | 0.633 |
| **Jaro** | 0.488 | 0.368 | 0.192 | 0.802 | 0.739 | 0.842 | 0.786 | 0.431 | 0.045 | 1 | 0.994 | 0.517 |
| **JaroW** | 0.530 | 0.417 | 0.261 | 0.783 | 0.708 | 0.828 | 0.760 | 0.440 | 0.056 | 0.994 | 1 | 0.544 |
| **LCS** | 0.857 | 0.943 | 0.731 | 0.613 | 0.626 | 0.523 | 0.542 | 0.720 | 0.633 | 0.517 | 0.544 | 1 |

**Вывод:**

Исходя из преведённых таблиц можно заметить, что пары алгоритмов: (*Levenshtein, Damerau-Levenshtein, LCS*), (*Jaccard, Cossine, Q-gram*), (*Jaro, Jaro-Winkler*) имеют между друг другом высокую корреляцию. Для идентичных строк ("identical - identical") все метрики корректно возвращают 0. Для полностью различных строк ("completely" "different") метрики достигают максимумов. Для строк разной длины Hamming не применим.

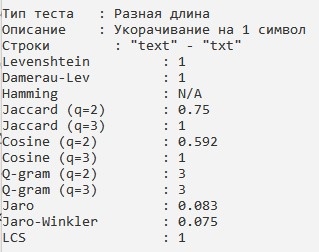
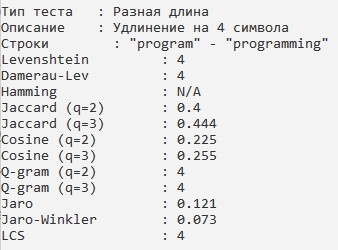
Часть 2: Анализ чувствительности к разным типам различий

Чувствительность к опечаткам



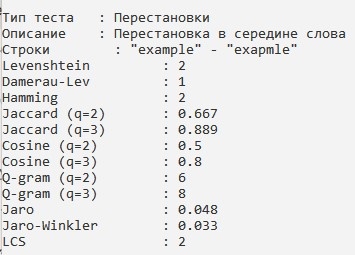
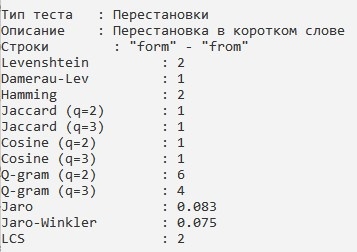
Таким образом, наиболее чувствительными к опечаткам оказались метрики редактирования (Levenshtein, Damerau-Levenshtein), а также Hamming для строк с одинаковой длинной.

Чувствительность к разной длине строк



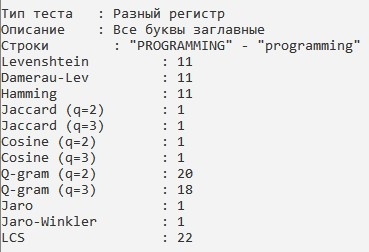
Таким образом, к разной длине строк наиболее чувсвительны все метрики, кроме Hamming, который не работает с строками имеющие разные длину.

Чувствительность к перестановкам символов



Таким образом, к перестановкам символов наиболее чувствителен Damerau-Levenstein, поскольку его задумка в этом.

Чувствительность к разному регистру



Часть 3: Производительность метрик