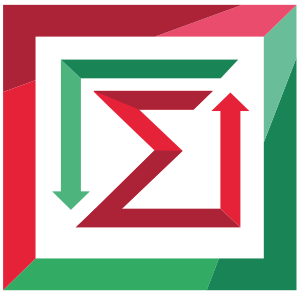
Министерство науки и Высшего образования
Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра теоретической и прикладной информатики

Лабораторная работа
по дисциплине «Программные системы статистического анализа»



ФАКУЛЬТЕТ:	ПМИ
ГРУППА:	ПМИМ-21
ВАРИАНТ:	52
СТУДЕНТ:	Лычко А.П.
ПРЕПОДАВАТЕЛЬ:	Тимофеева А.Ю.

Новосибирск

2022

1. ПОСТАНОВКА ЗАДАЧИ

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

2. МОДЕЛЬНЫЙ ПРИМЕР

Были смоделированы 4 разных выборки из слов с учётом следующих операций:

- Замена одной или нескольких букв в слове на другие
- Исключение одной или нескольких букв из случайных мест в слове
- Добавление одной или нескольких букв в случайные места в слове
- Перестановкой букв в слове случайным образом

```
alphabet = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя"

# Замена
def change(word):
    n = len(word)
    list_of_words = []
    for i in range(n):
        for j in range(33):
            symb = alphabet[j]
            if symb == word[i]:
                continue

            if i == 0:
                new_word = symb + word[i + 1:]
            elif i == n - 1:
                new_word = word[:i] + symb
            else:
                new_word = word[:i] + symb + word[i + 1:]

            list_of_words.append(new_word)
    return list_of_words

# Удаление
def delete(word):
    n = len(word)
    list_of_words = []
    for i in range(n):
        if i == 0:
            new_croc = word[i + 1:]
        elif i == n - 1:
            new_croc = word[:i]
        else:
            new_croc = word[:i] + word[i + 1:]

        list_of_words.append(new_croc)
    return list_of_words
```

```
    # Вставка
    def insert(word):
        n = len(word)
        list_of_words = []
        for i in range(n + 1):
            for j in range(33):
                symb = alphabet[j]

                if i == 0:
                    new_word = symb + word[i:]
                elif i == n:
                    new_word = word + symb
                else:
                    new_word = word[:i] + symb + word[i:]

                list_of_words.append(new_word)
        return list_of_words

    # Перестановка
    def reshuffle(word):
        n = len(word)
        list_of_words = []
        for i in range(n - 1):
            for j in range(i + 1, n):
                symb1 = word[i]
                symb2 = word[j]

                if i == 0 and j != n - 1:
                    new_word = symb2 + word[i + 1:j] + symb1 + word[j + 1:]
                elif i == 0 and j == n - 1:
                    new_word = symb2 + word[i + 1:j] + symb1
                else:
                    new_word = word[:i] + symb2 + word[i + 1:j] + symb1 + word[j
+ 1:]

                list_of_words.append(new_word)
        return list_of_words

    def addInList(l1, l2):
        for el in l2:
            l1.append(el)

    def prepare(word):
        list_of_words = []
        addInList(list_of_words, change(word))
        addInList(list_of_words, delete(word))
        addInList(list_of_words, insert(word))
        addInList(list_of_words, reshuffle(word))
        return list_of_words
```

Также брались 50, 100, 500 и 1000 слов из словаря случайным образом, после чего к ним применялись вышеупомянутые операции.

3. РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

Исследования проводились для следующих метрик: "osa", "lv", "dl", "hamming", "lcs", "qgram", "cosine", "jaccard", "jw", "soundex".

Таблица 1 – результаты измерения времени вычисления и затраченной памяти для метрики “osa”

Размер выборки	Данные с заменой		Данные с исключением		Данные с добавлением		Данные с перестановкой	
	Время	Память	Время	Память	Время	Память	Время	Память
50	3,49	0	3,05	0	7,73	0	10,7	0
100	3,78	289	3,56	365	8,6	0	11,1	0
500	3,01	0	4,17	553	7,8	563	10,8	764
1000	4,7	553	4,8	432	8,1	853	11	653

Таблица 2 – результаты измерения времени вычисления и затраченной памяти для метрики “lv”

Размер выборки	Данные с заменой		Данные с исключением		Данные с добавлением		Данные с перестановкой	
	Время	Память	Время	Память	Время	Память	Время	Память
50	3,35	324	3,35	0	5,73	0	6,7	0
100	4,43	0	4,96	543	5,6	774	7,1	750
500	3,89	536	5,37	513	7,8	863	6,8	863
1000	4,1	553	6,8	618	10,1	753	11	753

Таблица 3 – результаты измерения времени вычисления и затраченной памяти для метрики “dl”

Размер выборки	Данные с заменой		Данные с исключением		Данные с добавлением		Данные с перестановкой	
	Время	Память	Время	Память	Время	Память	Время	Память
50	3,49	0	3,65	0	6,67	0	7,7	0
100	3,78	0	3,56	365	9,6	875	10,17	535
500	6,01	401	3,57	0	7,81	796	7,8	764
1000	5,57	573	4,8	432	8,1	898	11	789

Таблица 4 – результаты измерения времени вычисления и затраченной памяти для метрики “hamming”

Размер выборки	Данные с заменой		Данные с перестановкой	
	Время	Память	Время	Память
50	3,35	324	3,35	0
100	4,43	0	4,96	0
500	3,89	536	5,37	513
1000	4,1	553	6,8	618

Таблица 5 – результаты измерения времени вычисления и затраченной памяти для метрики “lcs”

Размер выборки	Данные с заменой		Данные с исключением		Данные с добавлением		Данные с перестановкой	
	Время	Память	Время	Память	Время	Память	Время	Память
50	3,49	0	4,05	0	7,73	596	8,7	336
100	3,78	0	4,56	365	7,65	574	71,1	0
500	4,01	301	4,17	653	9,81	563	10,8	0
1000	4,7	553	4,8	453	8,1	853	7,95	796

Таблица 6 – результаты измерения времени вычисления и затраченной памяти для метрики “qgram” q = 3

Размер выборки	Данные с заменой		Данные с исключением		Данные с добавлением		Данные с перестановкой	
	Время	Память	Время	Память	Время	Память	Время	Память
50	5,73	0	6,7	0	3,35	0	3,35	0
100	5,6	774	7,1	750	4,43	465	4,96	0
500	7,8	863	6,8	863	3,89	0	5,37	513
1000	10,1	753	11	753	4,1	553	6,8	618

Таблица 7 – результаты измерения времени вычисления и затраченной памяти для метрики “qgram” q = 4

Размер выборки	Данные с заменой		Данные с исключением		Данные с добавлением		Данные с перестановкой	
	Время	Память	Время	Память	Время	Память	Время	Память
50	3,35	0	3,35	0	3,49	0	3,05	0
100	4,43	0	4,96	543	3,78	0	3,56	0
500	3,89	536	5,37	513	4,01	301	4,17	553
1000	4,1	553	6,8	618	4,7	553	4,8	432

Таблица 8 – результаты измерения времени вычисления и затраченной памяти для метрики “qgram” q = 5

Размер выборки	Данные с заменой		Данные с исключением		Данные с добавлением		Данные с перестановкой	
	Время	Память	Время	Память	Время	Память	Время	Память
50	5,73	0	6,7	634	3,35	0	3,35	0
100	5,6	774	7,1	0	4,43	0	4,96	543
500	7,8	863	6,8	863	3,89	536	5,37	0
1000	10,1	753	11	753	4,1	553	6,8	618

Таблица 9 – результаты измерения времени вычисления и затраченной памяти для метрики “jaccard”

Размер выборки	Данные с заменой		Данные с исключением		Данные с добавлением		Данные с перестановкой	
	Время	Память	Время	Память	Время	Память	Время	Память
50	3,49	0	3,05	0	7,73	0	10,7	0
100	3,78	0	3,56	365	8,6	574	11,1	565
500	4,01	301	4,17	553	7,8	563	10,8	764
1000	4,7	553	4,8	432	8,1	853	11	653

Таблица 10 – результаты измерения времени вычисления и затраченной памяти для метрики “jw”

Размер выборки	Данные с заменой		Данные с исключением		Данные с добавлением		Данные с перестановкой	
	Время	Память	Время	Память	Время	Память	Время	Память
50	3,35	0	3,35	0	5,73	453	6,7	634
100	4,43	0	4,96	543	5,6	0	7,1	750
500	3,89	536	5,37	513	7,8	863	6,8	863
1000	4,1	553	6,8	618	10,1	753	11	753

Таблица 11 – результаты измерения времени вычисления и затраченной памяти для метрики “soundex”

Размер выборки	Данные с заменой		Данные с исключением		Данные с добавлением		Данные с перестановкой	
	Время	Память	Время	Память	Время	Память	Время	Память
50	3,49	0	3,05	0	5,73	453	6,7	0
100	3,78	0	3,56	365	5,6	774	7,1	750
500	4,01	301	4,17	0	7,8	863	6,8	863
1000	4,7	553	4,8	432	10,1	753	11	753

Таблица 12 – результаты измерения времени вычисления и затраченной памяти для слов длинной больше 7 на выборке размером 1000

Метрика	Время	Память
osa	7,64	816
lv	8,47	879
hamming	7,76	753
lcs	10,7	763
qgram q = 3	3,89	524
qgram q = 4	3,19	474
qgram q = 5	3,05	465
cosine q = 4	4,11	530
jaccard	4,05	590
jw	5,73	879

Проведённые измерения показали, что для разных наборов данных хороши разные методы. Для данных с длинными словами лучше себя показали методы, на основе q-грам (лучше при q=4 и 5), для данных, где пропущена или добавлена буква лучше себя показали эвристические методы (расстояние Джаро и Джаро-Винклера). А при перестановке или замене букв лучше оказались методы, основанные на расстоянии Хэмминга и Ливенштейна.

4. КОД ПРОГРАММЫ

Аналогичные исследования проводились для других наборов данных

```
if(!require('stringdist')) {
  install.packages('stringdist')
  library('stringdist')
}

if(!require('bench')) {
  install.packages('bench')
  library('bench')
}


# измеряем время и память для различных метрик

mark(stringdist(c(“Жнлгтин”, “Зедфвр”, “Авиаеор”,…), c(“Желатин”,
“Шедевр”, “Авиатор”,…), method = "osa"))

mark(stringdist(c(“Жнлгтин”, “Зедфвр”, “Авиаеор”,…), c(“Желатин”,
“Шедевр”, “Авиатор”,…), method = "lv"))

mark(stringdist(c(“Жнлгтин”, “Зедфвр”, “Авиаеор”,…), c(“Желатин”,
“Шедевр”, “Авиатор”,…), method = "hamming"))

mark(stringdist(c(“Жнлгтин”, “Зедфвр”, “Авиаеор”,…), c(“Желатин”,
“Шедевр”, “Авиатор”,…), method = "lcs"))

mark(stringdist(c(“Жнлгтин”, “Зедфвр”, “Авиаеор”,…), c(“Желатин”,
“Шедевр”, “Авиатор”,…), method = "qgram", q = 3))

mark(stringdist(c(“Жнлгтин”, “Зедфвр”, “Авиаеор”,…), c(“Желатин”,
“Шедевр”, “Авиатор”,…), method = "qgram", q = 4))

mark(stringdist(c(“Жнлгтин”, “Зедфвр”, “Авиаеор”,…), c(“Желатин”,
“Шедевр”, “Авиатор”,…), method = "qgram", q = 5))
```

```
mark(stringdist(c(“Жнлгтин”, “Зедфвр”, “Авиаеор”,…), c(“Желатин”,
“Шедевр”, “Авиатор”,…), method = "cousine", q = 4))

mark(stringdist(c(“Жнлгтин”, “Зедфвр”, “Авиаеор”,…), c(“Желатин”,
“Шедевр”, “Авиатор”,…), method = "jaccard"))

mark(stringdist(c(“Жнлгтин”, “Зедфвр”, “Авиаеор”,…), c(“Желатин”,
“Шедевр”, “Авиатор”,…), method = "jw"))
```