МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

Национальный исследовательский университет ИТМО

МЕГАФАКУЛЬТЕТ ТРАНСЛЯЦИОННЫХ ИНФОРМАЦИОННЫЙ ТЕХНОЛОГИЙ

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПРОГРАММИРОВАНИЯ

ЛАБОРАТОРНАЯ РАБОТА №4

По дисциплине «Введение в цифровую культуру и программирование»

Исправление ошибок

Выполнил Шеин Максим Андреевич

(Фамилия Имя Отчество)

Проверила Страдина Марина Владимировна

(Фамилия Имя Отчество)

Санкт-Петербург, 2020г.

Оглавление

1)Задание

1. Предобработка текста
1.1Текст нужно разделить на слова.
1.2 Удалить следующие знаки препинания: ! ? , ; . : — « ( ) » Не удаляйте из слов дефисы. 1.3 Перевести все буквы в строчные (маленькие). Например, "Средний" - заменить на "средний".
Приводить слова к нормальной форме не нужно, так как в словаре присутствуют различные словоформы.
Например, которая которого которое которой котором которому которую которые который которым которыми которых.
Это всё разные словоформы. Всего в словаре 4772 разных словоформ, отсортированных по алфавиту.

2. Первичные расчёты
2.1 Посчитайте словоформы в своём тексте
2.2 Посчитайте разные словоформы
2.3 Посчитайте сколько разных словоформ из вашего текста присутствуют в словаре Обратите внимание, что в словаре после слова через пробел написано число – это частота встречаемости во всём тексте.

3. Поиск и исправление ошибок
3.1 Посчитайте, сколько словоформ не присутствует в словаре ("потенциальные ошибки")
3.2 Найдите для каждого из них редакторское расстояние до ближайшего слова. Редакторское расстояние – это минимальное количество разрешённых операций, необходимых для превращения одной строки в другую. В настоящем задании разрешены следующие операции: вставка одного символа, удаление одного символа и замена одного символа на другой. Допустимо в строку вставить символ «пробел», превратив строку в две.
3.3 В настоящем задании, если редакторское расстояние равняется 1 или 2, то словоформа в вашем тексте признаётся ошибочной и её нужно заменить на соответствующую словоформу из словаря. Если в словаре оказалось несколько словоформ с одинаковым редакторским расстоянием до ошибочной словоформы из текста, то нужно заменить на ту, у которой частота выше.

4. После поиска и исправления ошибок повторите расчёты:
4.1 Посчитайте словоформы в своём тексте
4.2 Посчитайте разные словоформы
4.3 Посчитайте сколько разных словоформ из вашего текста присутствуют в словаре

5. Выведите все "потенциальные ошибки" в порядке встречаемости в тексте в следующем виде: словоформа из текста - словоформа из словаря - редакторское расстояние. Если удалось исправить не все "потенциальные ошибки", то нужно вывести только неисправленное слово из текста с пометкой "не найдено".

2)Исходный текст

brain109.txt

https://d1b10bmlvqabco.cloudfront.net/paste/ke45e8wblop2qf/d5273e9bcadd86841c6e4ed5a90fcb1fe839f6085b39941b16d50b4acbc0b276/brain109.txt

3)Код функции для подсчёта редакторского расстояния

File Edit Format Run Options Window Help

```
from collections import Counter


def redactrast(str1, str2):
    length1 = len(str1)
    length2 = len(str2)
    editor = range(0, length1 + 1)
    for i in range(1, length2 + 1):
        prev = editor
        editor = [i] + [0] * length1
        for j in range(1, length1 + 1):
            ch = prev[j - 1]
            add = prev[j] + 1
            d = editor[j - 1] + 1
            if str1[j - 1] != str2[i - 1]:
                ch += 1
            editor[j] = min(ch, add, d)
    return editor[length1]
```

4)Ответы на вопросы 2-5 и код

Вопрос №2

Количество словоформ: 1733

Количество разных словоформ: 841

Количество разных словоформ, присутствующих в словаре: 834

Вопрос №3

Количество разных словоформ, отсутствующих в словаре: 7

В коде для каждого слова было найдено и использовано редакторское расстояние

Словоформы, отсутствующие в словаре: влиание, гипоталамиса, деревняя, препринимать, эдектроэнцефадограмму, поседний, станавливает

Вопрос №4

Количество словоформ: 1733

Количество разных словоформ: 840

Количество разных словоформ, присутствующих в словаре: 840

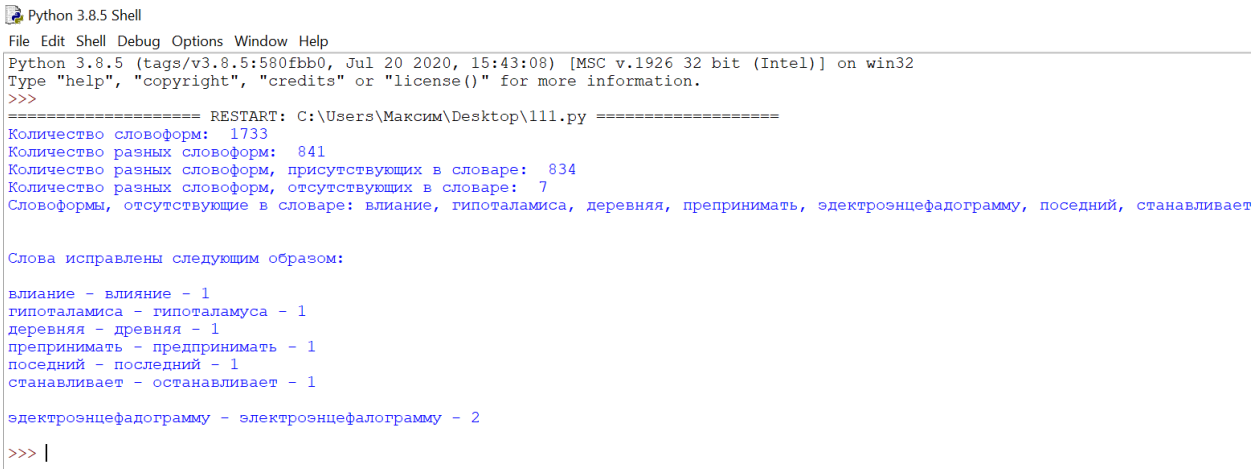
Вопрос №5

влиание - влияние - 1

гипоталамиса - гипоталамуса - 1

деревняя - древняя - 1
препринимать - предпринимать - 1
поседний - последний - 1
станавливает - останавливает – 1

эдектроэнцефадограмму - электроэнцефалограмму – 2



Программный код на Python

```
from collections import Counter


def redactrast(str1, str2):
    length1 = len(str1)
    length2 = len(str2)
    editor = range(0, length1 + 1)
    for i in range(1, length2 + 1):
        prev = editor
        editor = [i] + [0] * length1
        for j in range(1, length1 + 1):
            ch = prev[j - 1]
            add = prev[j] + 1
            d = editor[j - 1] + 1
            if str1[j - 1] != str2[i - 1]:
                ch += 1
            editor[j] = min(ch, add, d)
    return editor[length1]


def char(str1, ch, idx):
    if idx >= len(str1):
        result = str1 + ch
```

```
        return result
    result = ''
    i = 0
    while i < len(str1):
        if i == idx:
            result += ch
            result += str1[i]
            i += 1
        else:
            result += str1[i]
            i += 1
    return result


def resize_char(str1, idx):
    result = ''
    for i in range(len(str1)):
        if i == idx:
            result += str1[i].upper()
        else:
            result += str1[i]
    return result


text_file = open("brain109.txt", "r", encoding = "utf8")
dict_file = open("dict1.txt", "r")
fout = open("Исправленный_текст.txt", "w", encoding = "utf8")
text = text_file.read()
text_copy = text
text = text.replace(".", "")
text = text.replace(",", "")
text = text.replace("?", "")
text = text.replace("!", "")
text = text.replace(":", "")
text = text.replace(";", "")
text = text.replace("(", "")
text = text.replace(")", "")
text = text.replace("»", "")
text = text.replace("«", "")
text = text.replace("{", "")
text = text.replace("}", "")
text = text.replace("—", "")
text = text.lower()
text_list = text.split()
frequency = Counter(text_list)
print("Количество словоформ: ", len(text.split()))
print("Количество разных словоформ: ", len(frequency))

key = []
```

```
value = []
for string in dict_file:
    a, b = string.replace("\n", "").split(' ')
    key.append(a)
    value.append(b)
dict_of_dict = {key[i]: value[i] for i in range(len(key))}
text_el = {str1: frequency[str1] for str1 in frequency if str1 in dict_of_dict}
print("Количество разных словоформ, присутствующих в словаре: ", len(text_el))
print("Количество разных словоформ, отсутствующих в словаре: ", len(frequency) -
len(text_el))

words_not_in_dict = []
for word in text_list:
    if word not in dict_of_dict:
        words_not_in_dict.append(word)
print("Словоформы, отсутствующие в словаре: ", end='')
print(*words_not_in_dict, sep=", ")
print('\n')

corrected_words_1 = {}
for word in words_not_in_dict:
    for i in range(1, len(word)):
        if word[0:i] in dict_of_dict.keys() and word[i:len(word)] in dict_of_dict.keys():
            temp = corrected_words_1.get(word, [])
            if len(temp) == 0:
                corrected_words_1[word] = [word[0:i], word[i:len(word)]]
            elif (frequency[word[0:i]] + frequency[word[i:len(word)]]) > (frequency[temp[0]] +
frequency[temp[1]]):
                corrected_words_1[word][0] = word[0:i]
                corrected_words_1[word][1] = word[i:len(word)]
    for ch in dict_of_dict.keys():
        if redactrast(word, ch) == 1:
            temp = corrected_words_1.get(word, [])
            if len(temp) == 0:
                corrected_words_1[word] = [ch]
            elif len(temp) == 1:
                if frequency[ch] > frequency[temp[0]]:
                    corrected_words_1[word][0] = ch
            elif len(temp) == 2:
                if frequency[ch] > (frequency[temp[0]] + frequency[temp[1]]):
                    corrected_words_1[word] = [ch]

uncorrected_words_1 = []
for word in words_not_in_dict:
    if word not in corrected_words_1.keys():
        uncorrected_words_1.append(word)

corrected_words_2 = {}
for word in uncorrected_words_1:
```

```
    for i in range(1, len(word)):
        for ch in dict_of_dict.keys():
            if word[0:i] in dict_of_dict.keys() and redactrast(word[i:len(word)], ch) == 1:
                temp = corrected_words_2.get(word, [])
                if len(temp) == 0:
                    corrected_words_2[word] = [word[0:i], ch]
                elif (frequency[word[0:i]] + frequency[ch]) > (frequency[temp[0]] +
frequency[temp[1]]):
                    corrected_words_2[word][0] = word[0:i]
                    corrected_words_2[word][1] = ch
            elif redactrast(word[0:i], ch) == 1 and word[i:len(word)] in dict_of_dict.keys():
                temp = corrected_words_2.get(word, [])
                if len(temp) == 0:
                    corrected_words_2[word] = [ch, word[i:len(word)]]
                elif (frequency[ch] + frequency[word[i:len(word)]]) > (frequency[temp[0]] +
frequency[temp[1]]):
                    corrected_words_2[word][0] = ch
                    corrected_words_2[word][1] = word[i:len(word)]
    for ch in dict_of_dict.keys():
        if redactrast(word, ch) == 2:
            temp = corrected_words_2.get(word, [])
            if len(temp) == 0:
                corrected_words_2[word] = [ch]
            elif len(temp) == 1:
                if frequency[ch] > frequency[temp[0]]:
                    corrected_words_2[0] = ch
            elif len(temp) == 2:
                if frequency[ch] > (frequency[temp[0]] + frequency[temp[1]]):
                    corrected_words_2 = [ch]

uncorrected_words_2 = []
for word in uncorrected_words_1:
    if word not in corrected_words_2.keys():
        uncorrected_words_2.append(word)

print("Слова исправлены следующим образом:\n")
for word in corrected_words_1.keys():
    print(word, '-', *corrected_words_1[word], '- 1')
print('')
for word in corrected_words_2.keys():
    print(word, '-', *corrected_words_2[word], '- 2')
print('')
for word in uncorrected_words_2:
    print(word, '-', "не найдено", '- >2')

text_copy_list = text_copy.split('\n')
text_result = ""
for string in text_copy_list:
    string_list = string.replace('\n', '').split()
```

```
for word in string_list:
    dict_ch = {}
    for ch in range(len(word)):
        if (word[ch] == '?') or (word[ch] == '!') or (word[ch] == '(') or (word[ch] == ')') or (
                word[ch] == ':') or (word[ch] == ';') or (word[ch] == '.') or (word[ch] == ',') or (
                word[ch] == '«') or (word[ch] == '»') or (word[ch] == '}') or (word[ch] == '{'):
            dict_ch[word[ch]] = dict_ch.get(word[ch], [])
            dict_ch[word[ch]].append(ch)
        elif ord(word[ch]) >= 1040 and ord(word[ch]) <= 1071:
            dict_ch['Upper'] = dict_ch.get('Upper', [])
            dict_ch['Upper'].append(ch)
    word = word.lower()
    word = word.replace("?", "")
    word = word.replace("!", "")
    word = word.replace("»", "")
    word = word.replace("«", "")
    word = word.replace(":", "")
    word = word.replace(";", "")
    word = word.replace(",", "")
    word = word.replace(".", "")
    word = word.replace("(", "")
    word = word.replace(")", "")
    word = word.replace("}", "")
    word = word.replace("{", "")
    if word in corrected_words_1.keys():
        final_word = ''
        for i in range(len(corrected_words_1[word])):
            if i == 0:
                final_word = corrected_words_1[word][i] + ' '
            else:
                final_word = final_word + corrected_words_1[word][i]
        for ch in dict_ch.keys():
            if ch != 'Upper':
                for pos in dict_ch[ch]:
                    final_word = char(final_word, ch, pos)
            else:
                for pos in dict_ch[ch]:
                    final_word = resize_char(final_word, pos)
        text_result += final_word + ' '
    elif word in corrected_words_2.keys():
        final_word = ''
        for i in range(len(corrected_words_2[word])):
            if i == 0:
                final_word = corrected_words_2[word][i] + ' '
            else:
                final_word = final_word + corrected_words_2[word][i]
        for ch in dict_ch.keys():
            if ch != 'Upper':
                for pos in dict_ch[ch]:
```

```
                final_word = char(final_word, ch, pos)
            else:
                for pos in dict_ch[ch]:
                    final_word = resize_char(final_word, pos)
        text_result = text_result + final_word + ' '
    else:
        final_word = word
        for ch in dict_ch.keys():
            if ch != 'Upper':
                for pos in dict_ch[ch]:
                    final_word = char(final_word, ch, pos)
            else:
                for pos in dict_ch[ch]:
                    final_word = resize_char(final_word, pos)
        text_result += final_word + ' '
text_result = text_result[0:len(text_result) - 1] + '\n'
fout.write(text_result)
```

5)Текст с исправленными ошибками

В связи с тем, что текст очень массивный, я оставлю ссылку на него.
https://pastebin.com/gkDU3cde