Московский Авиационный Институт
(Научный Исследовательский Институт)

Факультет прикладной математики и физики
Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу

«Обработка естественно-языковых текстов»

Студент: Мхитарян С.А.

Преподаватель: Кухтичев А.А.

Группа: М8О-206М

Дата:

Оценка:

Подпись:

Москва, 2021

ЛР1: Токенизация

Задание

Реализовать процесс разбиения текстов документов на токены, который потом будет использоваться при индексации. Для этого потребуется выработать правила, по которым текст делится на токены. Необходимо описать их в отчёте, указать достоинства и недостатки

выбранного метода. Привести примеры токенов, которые были выделены неудачно, объяснить, как можно было бы поправить правила, чтобы исправить найденные проблемы.

Метод решения

1. Изучение способов токенизации текста с использованием разных библиотек python.
2. Установка необходимых библиотек на ПК.
3. Выделение текста статей из скачанного дампа при помощи nltk и собственных наработок.
4. Написание и отладка кода, выполняющего разделение текста на отдельные токены.
5. Сбор и анализ статистических данных.

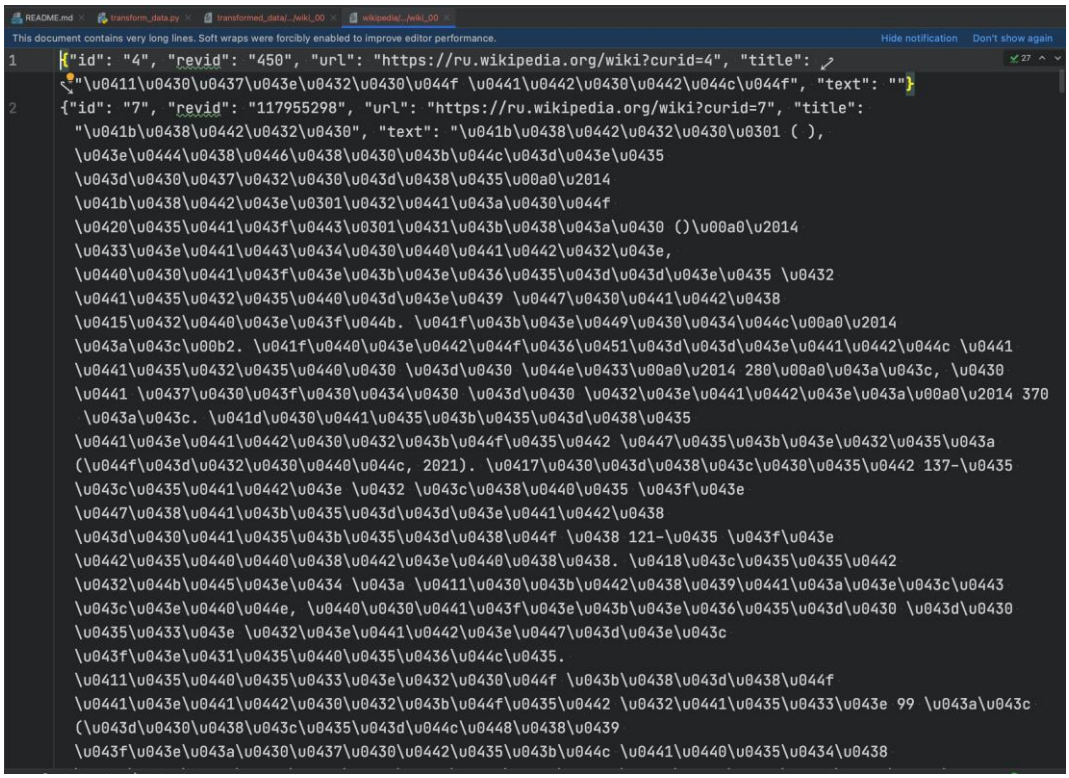
Журнал выполнения

№	Действие	Проблема	Решение
1	Установка nltk библиотеки python на ПК	-	
2	Выполнение программы на данных, которые были получены лабораторной работы 2 по информационному поиску	Время - можно оптимизировать с помощью потоков и более быстрого языка программирования	

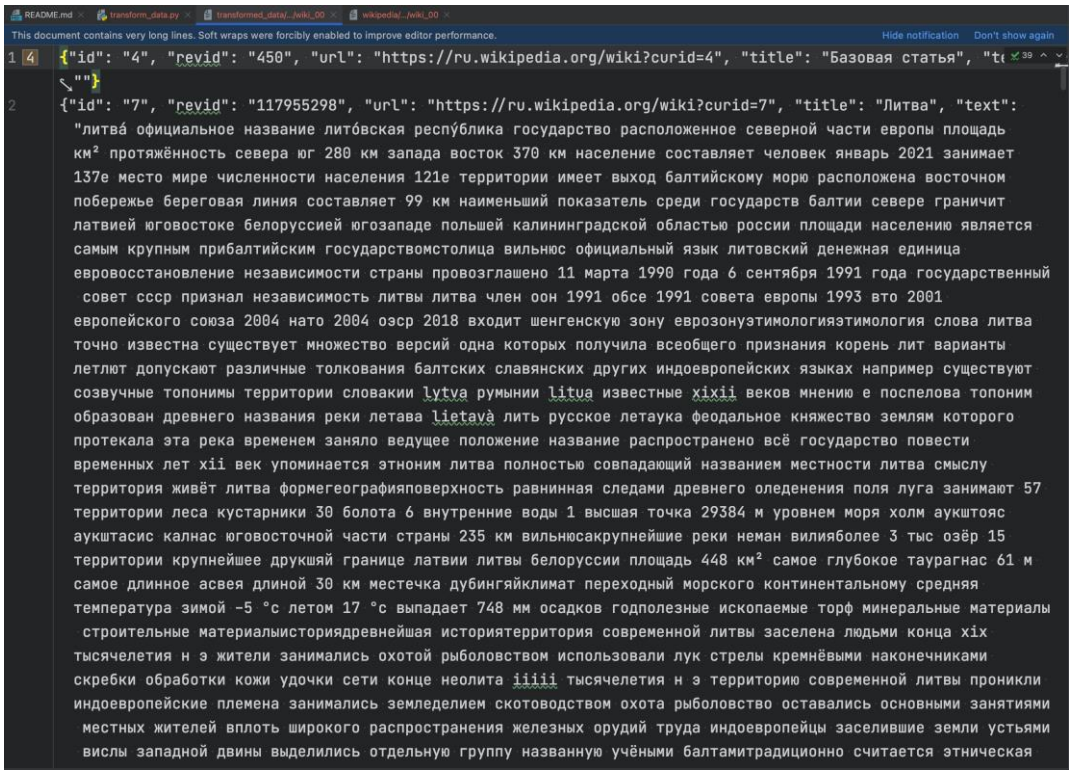
Результаты выполнения

Данные	Размер до токенизации	Размер после токенизации	Средняя длина токена	Время
Все статьи в формате json	18 ГБ	6.2ГБ	6.45	1:21:12.494894

Исходные данные



Полученный результат



Положительные стороны:

- можно быстро и удобно реализовать на языке Python

- библиотека с русскими словами уже готова, не нужно писать «велосипеды»

Отрицательные стороны:

- использования языка python (можно выбрать более оптимальный язык для работы с текстами)

- не все «лишние знаки» были подчищены

Исходный код

```python
import ssl
import json
import logging
import os.path
import threading
from os import listdir
from pathlib import Path
from datetime import datetime

import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize


# Avoid an error with the ssl certificate
try:
    _create_unverified_https_context = ssl._create_unverified_context
except AttributeError:
    pass
else:
    ssl._create_default_https_context = _create_unverified_https_context

nltk.download('stopwords')
nltk.download('punkt')
russian_stopwords = stopwords.words('russian')
```

```python
name = os.path.splitext(os.path.basename(__file__))[0]
home_path = '/Users/samvel/projects/MAI-IR'
wikipedia_data_path = os.path.join(home_path, 'IR', 'laboratory_work_1', 'wikipedia')
transformed_data_path = os.path.join(home_path, 'NLPT', 'laboratory_work_1', 'transformed_data')

bad_chars = {
    ')', '(', '[', ']', '{', '}', '\n', '\t', '\\', '/', '"', '\'', '.',
    ',', '*', '#', '@', '$', '%', '_', '-', '!', '?', '<', '>', '=', '+',
    '–', ':', '?', '|', '~', ';', '±', '&', '^', '№', '—', '»', '«', '€',
    '°c', '’', '“', '„', '…', 'SHY'
}


def transform_data(file_path, logger):
    logger.info(f'threadId={threading.current_thread().ident}, transform_file={file_path}')
    with open(file_path) as file:
        lines = file.readlines()

    data = ''
    for line in lines:
        json_data = json.loads(line)
        transformed_data = json_data['text'].lower()

        for bad_char in bad_chars:
            transformed_data = transformed_data.replace(bad_char, '')

        tokens = word_tokenize(transformed_data, language="russian")
        json_data['text'] = " ".join(token for token in tokens if token not in russian_stopwords)
        data += f'{json.dumps(json_data, ensure_ascii=False)}\n'

    folder_name = os.path.basename(os.path.dirname(file_path))
    file_name = os.path.splitext(os.path.basename(file_path))[0]

    article_path = os.path.join(transformed_data_path, folder_name, file_name)
    with open(article_path, 'w', encoding='utf8') as new_file:
        new_file.write(data)


if __name__ == '__main__':
    fmt = '%Y-%m-%d'
    logger = logging.getLogger(name)
    ch = logging.StreamHandler()
    formatter = logging.Formatter(fmt='%(asctime)s %(levelname)s %(name)s: %(message)s', datefmt='%y/%m/%d %H:%M:%S')
    ch.setFormatter(formatter)
    logger.addHandler(ch)
    logger.setLevel(logging.DEBUG)

    articles_paths = []
    for folder in sorted(listdir(wikipedia_data_path)):
        if not folder.startswith('.'):
            Path(os.path.join(transformed_data_path, folder)).mkdir(parents=True, exist_ok=True)
            articles_paths.append(os.path.join(wikipedia_data_path, folder))

    start = datetime.now()

    for articles_path in articles_paths:
        articles_path_folders = sorted(listdir(articles_path))
        articles = [os.path.join(articles_path, folder) for folder in articles_path_folders]

        threads = []
        for number, article in enumerate(articles):
            thread = threading.Thread(target=transform_data, name=f'thread_{number}', args=(article, logger))
            thread.start()
            threads.append(thread)

        for thread in threads:
            thread.join()

    end = datetime.now()
    logger.info(end - start)
```

Выводы

При выполнении лабораторной работы по обработке естественно-языковых текстов были токенизированы тексты из статей русской Википедии. Хочется отметить, что в этом помогла такая полезная python-библиотека, как nltk, в которой есть русские стоп слова. Также использование библиотек threading на Python помогло увеличить скорость токенизации примерно на 10%.