Университет ИТМО

Практическая работа №1

По дисциплине “Компьютерная лингвистика”

Автор: Лайок Олег Владимирович

Группа: К3242

Факультет: ИКТ

Преподаватель: Чернышева А.В.

Санкт-Петербург, 2021 г.

Цель: собрать свой корпус текстовых документов не менее 200 документов, суммарно не менее 10 000 слов, язык – русский.

Ход работы:

Для составления корпуса текстовых документов я выбрал работу с API ВКонтакте. Я уже имел опыт работы с данным API, кроме того, было интересно проанализировать заголовки новостей за какой-то период времени. Для этой работы я взял новостной паблик Лентач и выгрузил последнюю 1000 записей, за период 17.01.21 – 01.03.21.

Для начала я написал функцию insert_code, которая будет удобно передавать данные для запроса по методу wall.get. Единственные параметры, которые мне бы хотелось передавать в функции это domain – адрес паблика в ВКонтакте, offset – желаемое смещение в цикле для получения записей и count – общее число записей

```python
import requests
import time
import json

def insert_code(domain1, offset1, count1): #Фукция для удобной подстановки данных для запроса
        return """return API.wall.get ({
        "owner_id": "",
        "domain": "%s",
        "offset": %d,
        "count": %d,
        "filter": "owner",
        "extended": 0,
        "fields": "text",
        "v": "5.103"
    });""" % (domain1, offset1, count1)

```

Далее описываю функцию get_wall в которой и будет происходить парсинг постов из группы. Т. к. в API ВКонтакте стоит ограничение по выгрузке не более 100 записей за раз, я использую параметр offset для постепенной выгрузки записей. В данном случае offset=100, поэтому я выгружаю по 100 записей за раз пока не дойду до 1000. Соответственно при каждой итерации цикла я обращаюсь к полученному json файлу и записываю данные поля text, где и хранится текст поста, в список. Использую метод time.sleep, чтобы не

перегружать сервер запросами. Данная функция возвращает список текстов постов.

```python
def get_wall(domain,offset,count): #Сама функция получения данных со стены
    wall_posts=[]
    for i in range(1000):
        wall_posts.append(0)
    point=offset
    l=0 # счетчик для количества слов
    for offset in range(0,count,100): # используем цикл со смещением offset, чтобы обойти ограничение в 100 записей за раз
        response = requests.post( # Использую POST запрос, т.к. работал с ним раньше, плюс удобно передавать в нем данные запроса при работе с API
            url="https://api.vk.com/method/execute",
                data={
                    "code": insert_code(domain, offset, count),
                    "access_token": "69b24631b424eef4082d34d22b2811e8b55e3705ee93df5b0462630d42fc3299f12e834498a0e95b69056",
                    "v": "5.103"
                }
        )
        j=0
        for i in range(offset, offset+point):
            wall_posts[i]=response.json()['response']['items'][j]
            wall_posts[i]=wall_posts[i].get('text')
            s = wall_posts[i].split()
            l = len(s) + l
            j=j+1
        time.sleep(0.5)
    print('Average number of words in a post = ', round(l/count))
    print('Total number of words = ', l)
    return wall_posts
```

В основном теле программы запускаю функцию get_wall и выгружаю ее результат в json файл. Получился корпус из 1000 документов, в общей сложности на 35323 слова, в среднем по 35 слов на документ. Каждый документ представляет из себя краткий пересказ новости. В документах встречаются ссылки на полные новости, и некоторые специальные символы, которые в будущем следует удалить.

```python
with open("VK_data_lentach.json", "w", encoding='utf-8') as f:
    json.dump(get_wall("lentach",100,1000), f, ensure_ascii=False)
```

```
TERMINAL  OUTPUT  PROBLEMS  DEBUG CONSOLE
[Running] python -u "c:\Users\laolv\Desktop\Комп лингвистика\get_wall.py"
Average number of words in a post =  35
Total number of words =  35323

[Done] exited with code=0 in 16.21 seconds
```

Файл с кодом, а также выгруженные данные находятся в моем репозитории https://github.com/OlegLaiok/Comp_Lingv/tree/homework1