Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«ВОЛГОГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ВолгГТУ)

Факультет электроники и вычислительной техники

Кафедра «САПР и ПК»

Контрольная работа

по дисциплине «Компьютерная лингвистика и анализ текста»

Тема «Высота 102»

Выполнили:

Ложеницина А. В.  
САПР – 1.4

Проверил:

Коробкин Д. М.

Волгоград, 2020 г.

Содержание

[1 Постановка задачи 3](#_Toc43232184)

[2 Парсер сайта 5](#_Toc43232185)

[Список источников 6](#_Toc43232186)

[Приложение А – Код парсера 7](#_Toc43232187)

[Приложение Б – Код загрузки новостей в базу данных 9](#_Toc43232188)

1 Постановка задачи

Для выполнения семестровой работы был выдан вариант № 2 «Высота 102».

Задание для первого студента.

Распарсить сайт из списка и вывести в web-интерфейсе данные согласно номеру задания. Краулер должен считывать новостную ленту с первой страницы сайта. Периодичность повторения устанавливается пользователем. Данные заполняются в БД MongoDB. Обязательные поля для текста новости:

- название новости;

- дата новости;

- ссылка на новость;

- текст новости;

- ссылка на видео (если есть);

- количество просмотров новости (если есть);

- количество комментариев новости (если есть);

Задание для второго студента.

Создать программный модуль для анализа новостей из БД. Выделить с помощью Томита-парсера упоминание в тексте значимых персон Волгоградской области и достопримечательностей. Зафиксировать в БД предложения с их упоминанием для дальнейшего анализа тональности.

Создать программный модуль для проведения с помощью SparkMlLib анализ модели word2vec на всем объеме новостных статей из БД. Для персон Волгоградской области и достопримечательностей определить контекстные синонимы и слова, с которыми они упоминались в тексте.

Задание для третьего студента.

Создать программный модуль для выявления тональности высказываний по отношению к персонам Волгоградской области и достопримечательностям.

В данной работе реализован вариант исполнения задания для первого студента.

2 Парсер сайта

Парсер реализован на языке Python. Для парсинга сайта использована библиотека BeautifulSoup [1] (Приложение А).

Для хранения данных используется MongoDB (в частности MongoDBAtlas), для работы с ней использована библиотека MongoClient [2] (Приложение Б).

Пример коллекции новостей представлен на рисунке 1.

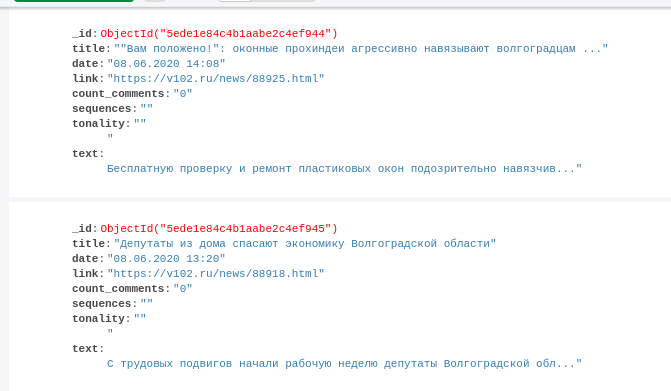


Рисунок 1 – Пример коллекции новостей

Список источников

1. BeautifulSoupDocumentation [Электронный ресурс] : офиц. сайт. – Режим доступа: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> (дата обращения 25.05.2020).
2. PyMongo 3.9.0 documentation [Электронный ресурс] : офиц. сайт. – Режим доступа: https://api.mongodb.com/python/current/api/pymongo/mongo

\_client.html (дата обращения 25.05.2020).

Приложение А – Код парсера

from bs4 import BeautifulSoup as bs

import requests as req

fromParser.Get\_session import fetch\_date

importaiohttp

importasyncio

defParse\_Site(site):

resp = req.get(site)

soup = bs(resp.text, 'html.parser')

articles = soup.find\_all(class\_="new-article")

lst\_articles = []

fori in articles:

lst\_articles.append({

'title': i.h3.get\_text(),

'date': i.find(class\_="mobile-date").get\_text(),

'link': i.find(class\_="detail-link")['href'],

'count\_comments': i.find(class\_="comment-icon").get\_text()

})

return articles

asyncdefdownload\_news(queue):

site = "https://v102.ru"

url\_form = "https://v102.ru/center\_line\_dorabotka\_ajax.php?page={num}&category={category}"

category = 0

async with aiohttp.ClientSession() as session:

fori in range(1, 2):

url = str.format(url\_form, num=i, category=category)

html = await fetch\_date(url, session)

news = Parse\_Site(html)

articles = await asyncio.gather(

\*[fetch\_date(link, session) for link in map(lambda article: site + article['link'], news)]

)

text\_articles = map(parse\_article, articles)

for new, new\_text in zip(news, text\_articles):

new['text'] = new\_text

new['link'] = site + new['link']

awaitqueue.put(news)

defparse\_article(html\_article):

html = bs(html\_article, 'html.parser')

returnhtml.find(class\_='n-text').get\_text()

Приложение Б – Код загрузки новостей в базу данных

importasyncio

import sys

fromParser.Parsing import download\_news

importpymongo

asyncdefAddDateBase(database, queue):

while True:

x = await queue.get()

db = database['news']

db.insert\_many(x)

queue.task\_done()

asyncdef main():

client = pymongo.MongoClient("localhost", 27017)

db = client['news\_date']

queue = asyncio.Queue()

downloader = asyncio.create\_task(download\_news(queue))

printer = asyncio.create\_task(AddDateBase(db, queue))

awaitasyncio.gather(downloader)

printer.cancel()

awaitasyncio.gather(printer, return\_exceptions=True)

if \_\_name\_\_ == '\_\_main\_\_':

asyncio.run(main())