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 «Волгоград Онлайн»

Выполнили:

Губенко Д.Д.  
САПР – 1.4

Проверил:

Коробкин Д. М.

Волгоград, 2020 г.

Содержание

[1 Постановка задачи 3](#_Toc44033302)

[2 Парсер сайта 4](#_Toc44033303)

[Список источников 5](#_Toc44033304)

[Приложение А – Код парсера 6](#_Toc44033305)

[Приложение Б – Код работы с базой данных 8](#_Toc44033306)

# Постановка задачи

Для выполнения семестровой работы был выдан вариант № 2: Волгоград онлайн

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 Spark MlLib 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.

# Парсер сайта

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

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

Пример коллекции новостей представлен на Рисунке 1.

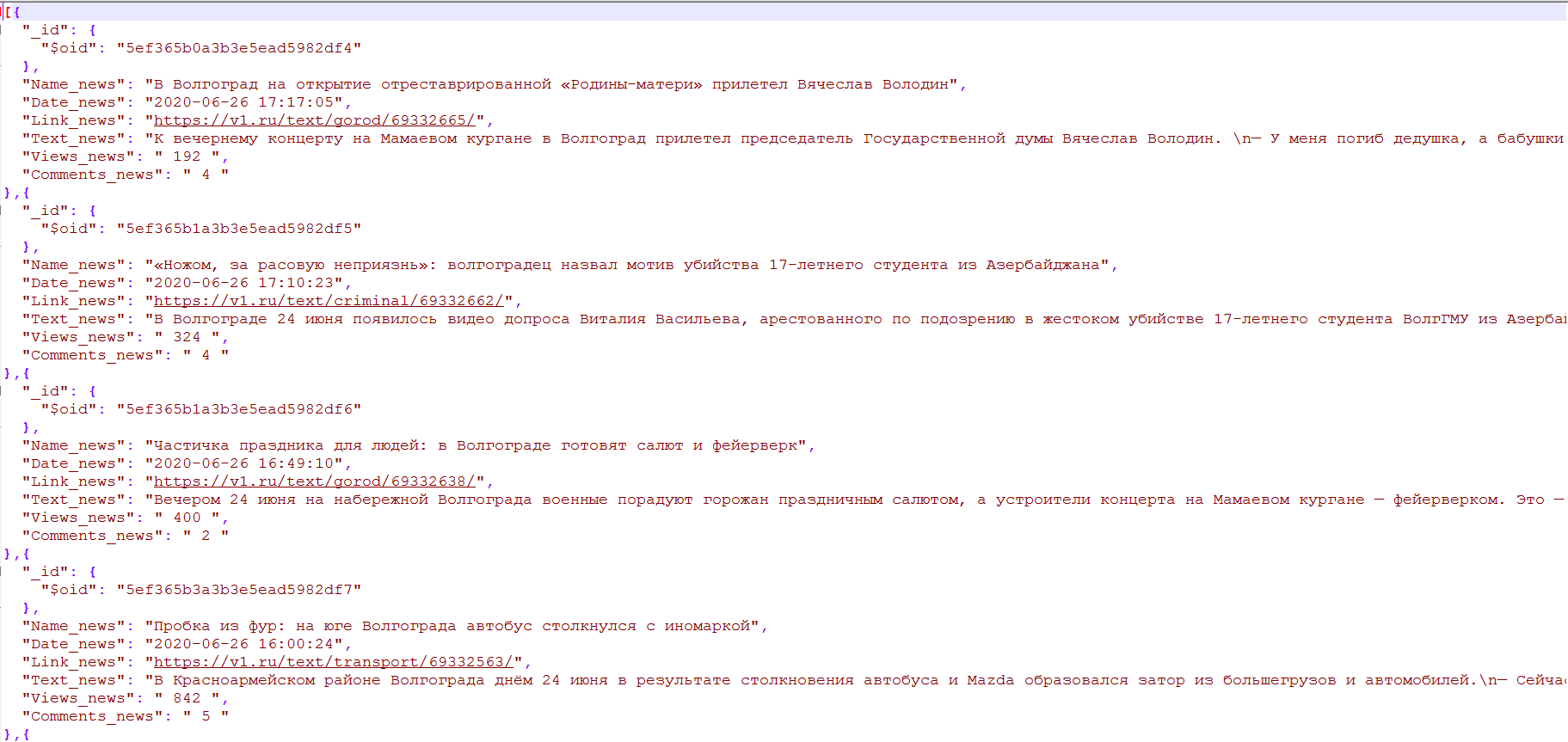


Рисунок 1 – Пример коллекции новостей

# Список источников

1. Beautiful Soup Documentation [Электронный ресурс] : офиц. сайт. – Режим доступа: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> (дата обращения 26.06.2020).
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 26.06.2020).

# Приложение А – Код парсера

import requests

from bs4 import BeautifulSoup

class parse:

def \_\_init\_\_(self):

self.get\_html = requests.get("https://v1.ru/text/").text

print(self.get\_html)

self.news = BeautifulSoup(self.get\_html, 'lxml').findAll('article', class\_='MNazv')

print(self.news)

self.name = ''

self.link = ''

self.date = ''

self.views = ''

self.comments = ''

self.text = ''

def Name\_News(self, idx):

self.name = self.news[idx].find('h2', class\_='MNb9').find('a').text

def Link\_News(self, idx):

self.link = 'https://v1.ru' + self.news[idx].find('h2', class\_='MNb9').find('a').get('href')

def Date\_news(self, idx):

self.date = self.news[idx].find('time').get('datetime')

def Views\_News(self, idx):

self.views = self.news[idx].find('div', class\_='LXch').find('span').text.replace("\xa0", "")

def Comments\_News(self, idx):

if self.news[idx].find('div', class\_='LXawl').findAll('span', class\_='LXbt')[1].text == " Обсудить ":

self.comments = "0"

else:

self.comments = self.news[idx].find('div', class\_='LXawl').findAll('span', class\_='LXbt')[1].text

def Text\_News(self):

txt = ''

for i in range(len(self.news)):

s = self.news[i].findAll('p')

for j in range(len(s)):

txt += s[j].text + '\n'

self.text = txt

# Приложение Б – Код работы с базой данных

from pymongo import MongoClient

class Date:

def \_\_init\_\_(self):

self.client = MongoClient()

self.db = self.client.news\_parse

self.new = self.db.news

def inset(self, date\_news):

self.new.insert\_one(date\_news)