Министерство науки и высшего образования Российской федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Факультет «Электроники и вычислительной техники»

Кафедра «Системы автоматизированного проектирования и поискового конструирования»

КОНТРОЛЬНАЯ РАБОТА

по дисциплине: «Компьютерная лингвистика»

Выполнил:

студент 3 курса, гр. ИВТ-360

Брылев К.А.

Проверил:

доц. Коробкин Д.М.

Волгоград, 2022

Оглавление

[Первая часть 3](#_Toc102500697)

[Вторая часть 7](#_Toc102500698)

[Третья часть 9](#_Toc102500699)

## Первая часть

Поставленные задачи:

Проанализировать сайт из списка и вывести в web-интерфейсе данные согласно номеру задания. Поисковый робот считывает новостную ленту с первой страницы сайта. Периодичность повторения устанавливается пользователем. Данные заполняются в БД MongoDB.

База данных:

В качестве СУБД для парсинга новостей использовалась MongoDB.

Созданные коллекции в базе данных:

* news – коллекция, содержащая данные о распарсенных новостях, а также найденные в них достопримечательности и персоны.

Значение полей:

1. \_id – идентификатор записи в MongoDB;
2. title – название новости;
3. date – дата публикации новости на сайт;
4. link – ссылка на источник;
5. text – текст новости;
6. cmts – кол-во комментариев к новости
7. people – список персон
8. places – список достопримечательностей

* TomitaCollection – содержащая данные для 2 и 3 задания.

Значение полей:

1. Id – идентификатор записи в MongoDB;
2. Person – имя персоны(если была найдена);
3. Place – достопримечательность (если была найдена);
4. Sentence – предложение в котором было найдено упоминание;
5. Text – весь текст рассматриваемой новости;
6. originalId – Idрассматриваемой новости в коллекции news;
7. Tonality – тональность найденного упоминания.

Парсер новостей:

Парсер работает по стандартному алгоритму:

* Проход по заданной странице сайта и получение ссылок на конкретные новости в пределах данной страницы;
* Переход по найденным ссылкам и получение данных о новости, необходимых для добавления в БД (заголовок, текст новости, ссылка, кол-во комментариев, дата публикации);
* Добавление данных о новости в БД.

Скриншоты таблиц коллекции news, заполненной в результате парсинга:

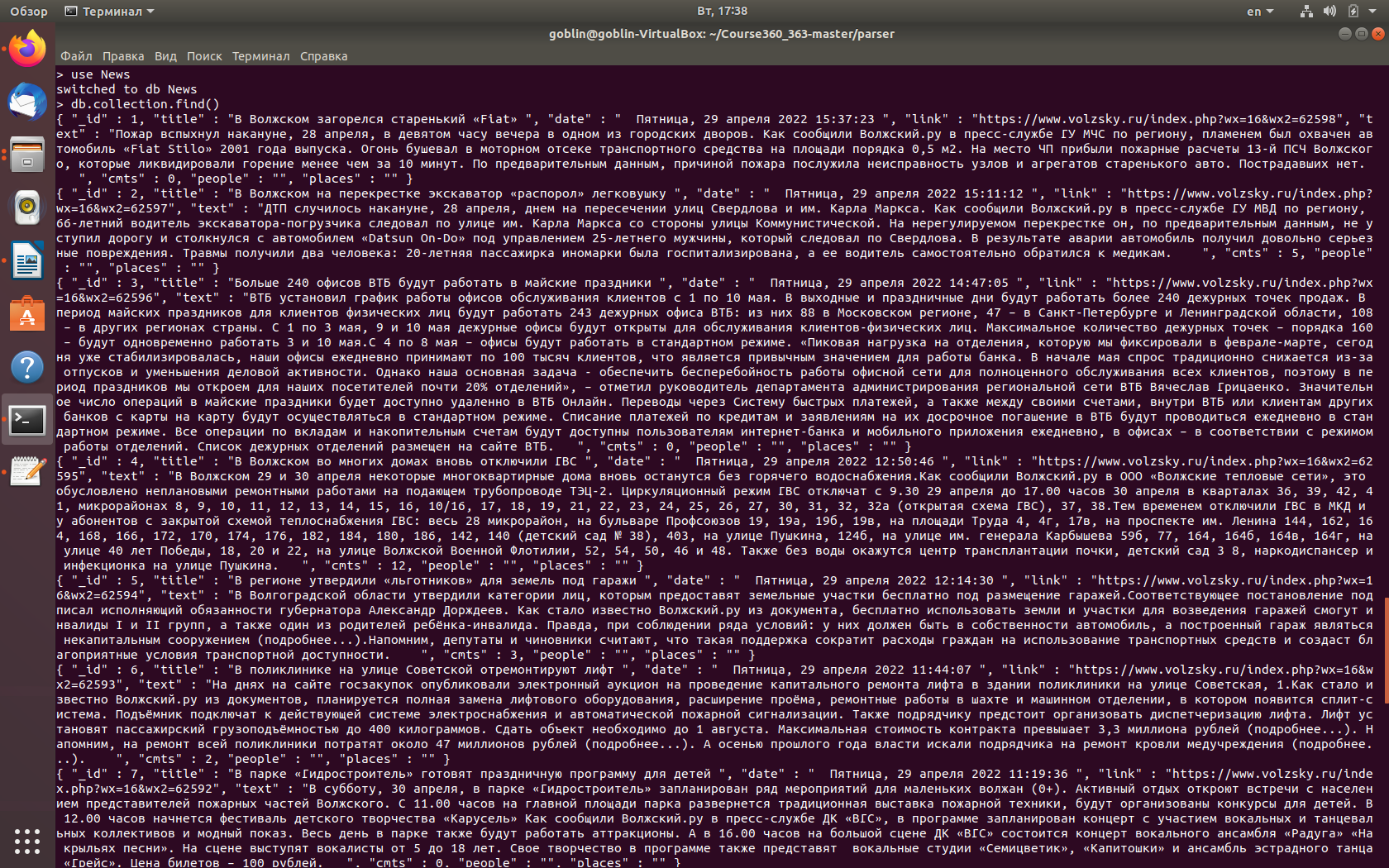


Рисунок 1 – содержимое news

Web-интерфейс:

Был реализован простой Web-интерфейс для отображения информации о каждой новости в коллекции, данные на интерфейсе соответствуют данным в БД.

Скриншоты Web-интерфейса:

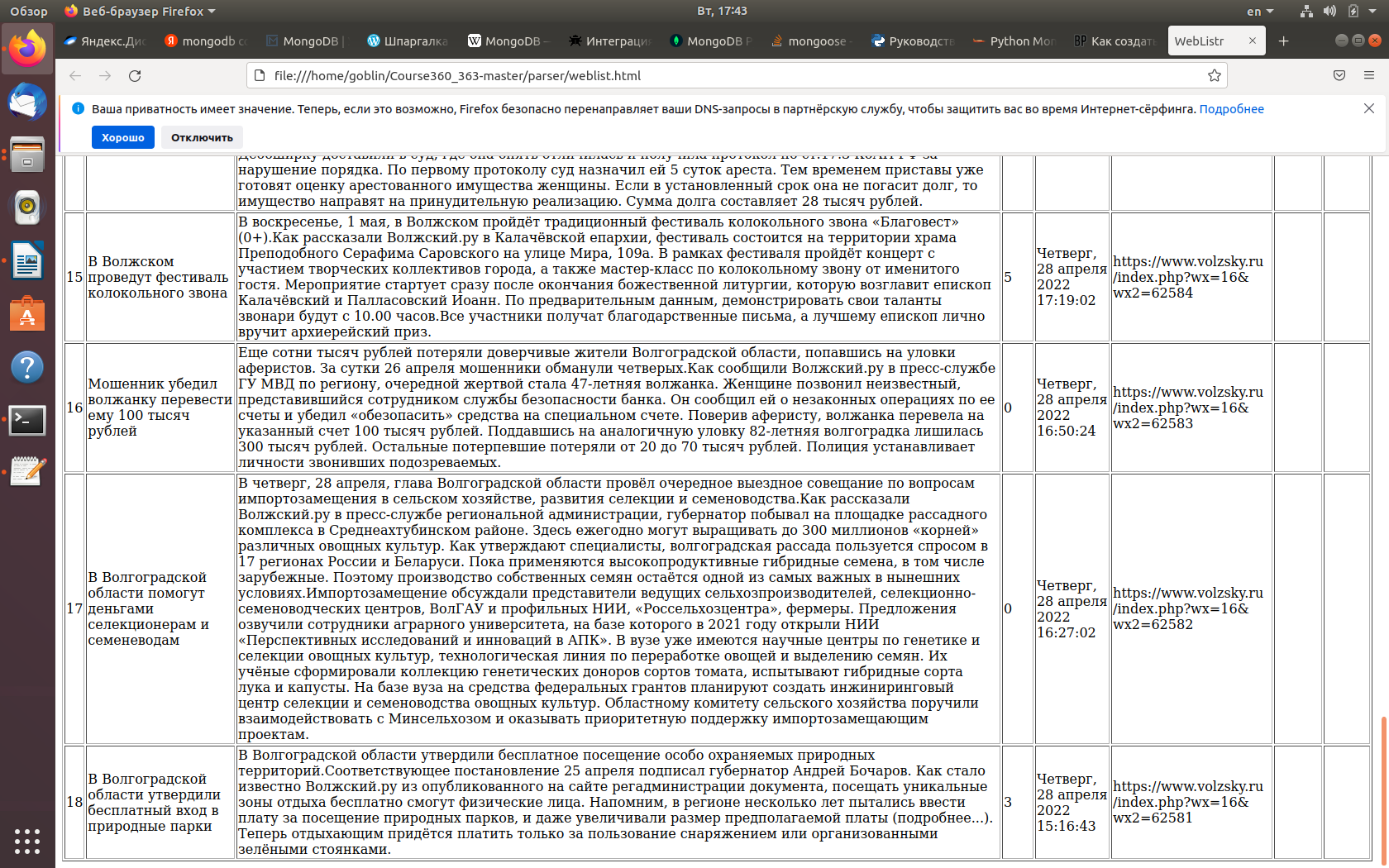


Рисунок 2 – пример Web-интерфейса

## Вторая часть

Томита-парсер:

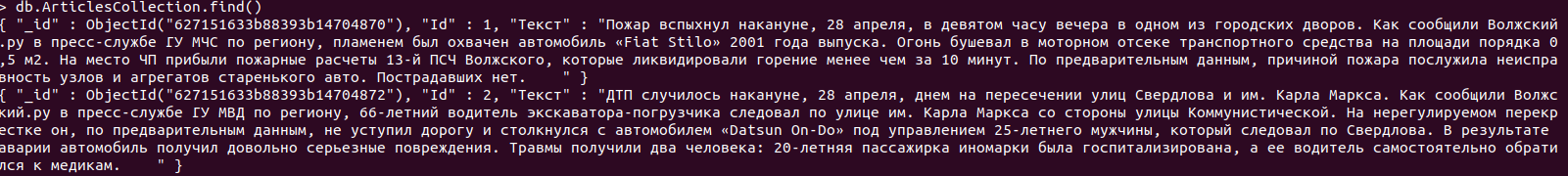


Рисунок 3 - Таблица ArticlesCollection

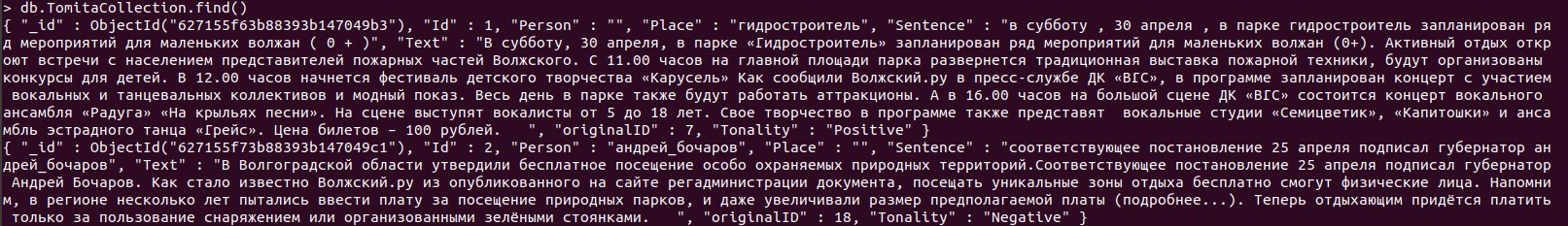


Рисунок 4 - Таблица TomitaCollection

Основной исполняемый файл – main.py. Файлы dbmodel.py и dbconfig.py являются вспомогательными – они осуществляют подключение и запись в БД информации. Исполняемый файл проверяет подключение, подключается к БД и применяет на содержимом новостей томита-парсер, обновляя БД и создавая ещё 2 коллекции со статьями и найденными персонами, достопримечательностями.

Для работы томита-парсера необходимо подключиться к БД. Программа создаёт файл textForCourseProject.txt в котором хранятся обнаруженные персоналии и достопримечательности. Полученные данные записываются в базу news с полями people и places. В коллекцию ArticlesСollection добавляются сами статьи, в коллекцию TomitaCollection добавляются записи новостей с обнаруженными персонами и достопримечательностями.

Word2Vec:

Для построения модели word2vec требуется подключение к БД. Программа собирает все статьи и записывает их в .txt-файлы, размещая в папке TxtFiles. Программа обучается на этих статьях с выделенными персонами и достопримечательностями

Были удалены стоп-слова, знаки препинания, после чего сгенерирована модель.

Сохранив модель, создаем словарь, с помощью которого будет произведен поиск по контекстным синонимам и словам, с которыми упоминались в тексте персоны и достопримечательности Волгоградской области, в частности Волжского.

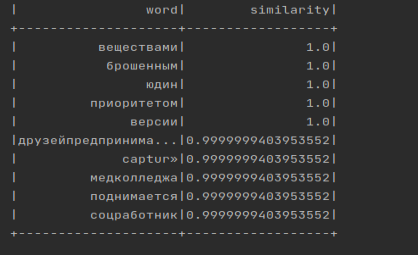


Рисунок 5 - Таблица синонимов к слову «Коронавирус»

# Третья часть

Поставленные задачи:

Создать программный модуль для выявления тональности высказываний по отношению к персонам Волгоградской области и достопримечательностям.Можно использовать либо подход на основе правил и словарей, либо методы машинного обучения.

Определение тональности

Для статей, в которых упоминались значимые персоны и достопримечательности области, был создан программный модуль, определяющий тональность высказывания.

Пример хранения в базе данных:

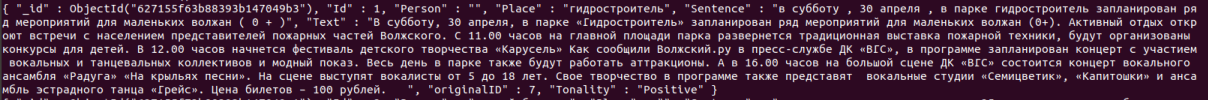


Рисунок 6 – Статья, тональность которой определена как положительная