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»

Выполнили:

студенты 3 курса, гр. ИВТ-363

Ильин И.В.

Яковлев М.П.

Якушина А.Ю.

Проверил:

доц. Коробкин Д.М.

Волгоград, 2020

Содержание

[Первая часть 3](#_Toc43168758)

[База данных 3](#_Toc43168759)

[Парсер 4](#_Toc43168760)

[Вторая часть 9](#_Toc43168761)

[Томита-парсер 9](#_Toc43168762)

[Word2Vec 10](#_Toc43168763)

[Третья часть 12](#_Toc43168764)

[Анализ тональности 12](#_Toc43168765)

[Руководство системного администратора 15](#_Toc43168766)

[Установка библиотек для Python 15](#_Toc43168767)

[Установка и настройка PyMongo 15](#_Toc43168768)

[Установка томита-парсера 15](#_Toc43168769)

[Руководство пользователя 17](#_Toc43168770)

[Работа с web-интерфейсом: 17](#_Toc43168771)

[Использование парсеров: 17](#_Toc43168772)

[Использование word2vec: 18](#_Toc43168773)

[Использование анализатора тональности: 18](#_Toc43168774)

# Первая часть

## База данных

Используется СУБД Mongodb.

Описание коллекций базы данных:

Tomito — предложения, в которых томита-парсер нашел персон и достопримечательности

news — все новости

tonality — предложения, для которых проанализирована тональность

Word2vecv2 – хранит синонимы

## Парсер

Парсер (crawler.py) работает в 2 этапа:

Первый этап: Selenium — проходит по новостной ленте и сохраняет ссылки на новостные статьи.

Второй этап: Библиотека Requests — проходит по сохраненным ссылкам на новости, отправляя запрос на каждую страницу со статьей и получая с нее данные, которые записываются в базу. Даты проходят через функцию date\_switch и нормализуются к виду "%Y %m %d".

Скриншоты таблиц заполненной парсерами базы данных:

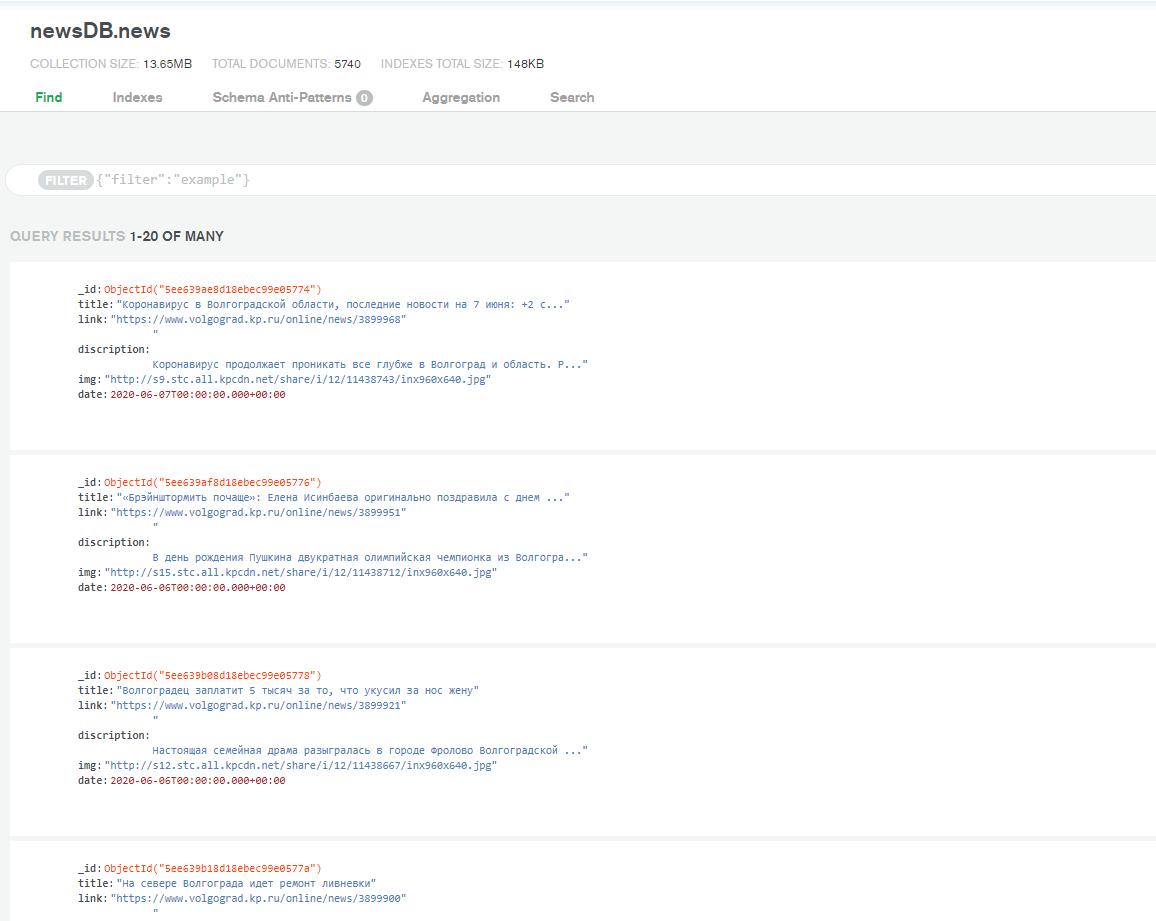


Рисунок 1. Таблица news

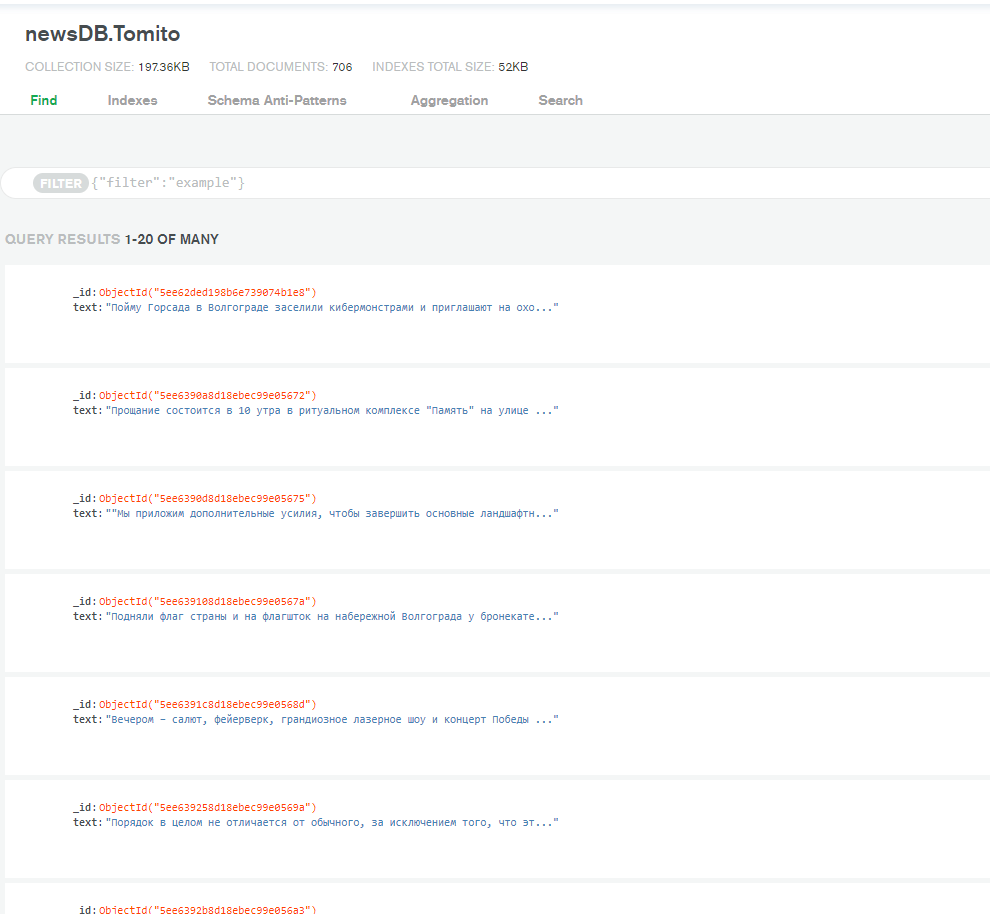


Рисунок 2. Таблица Tomito

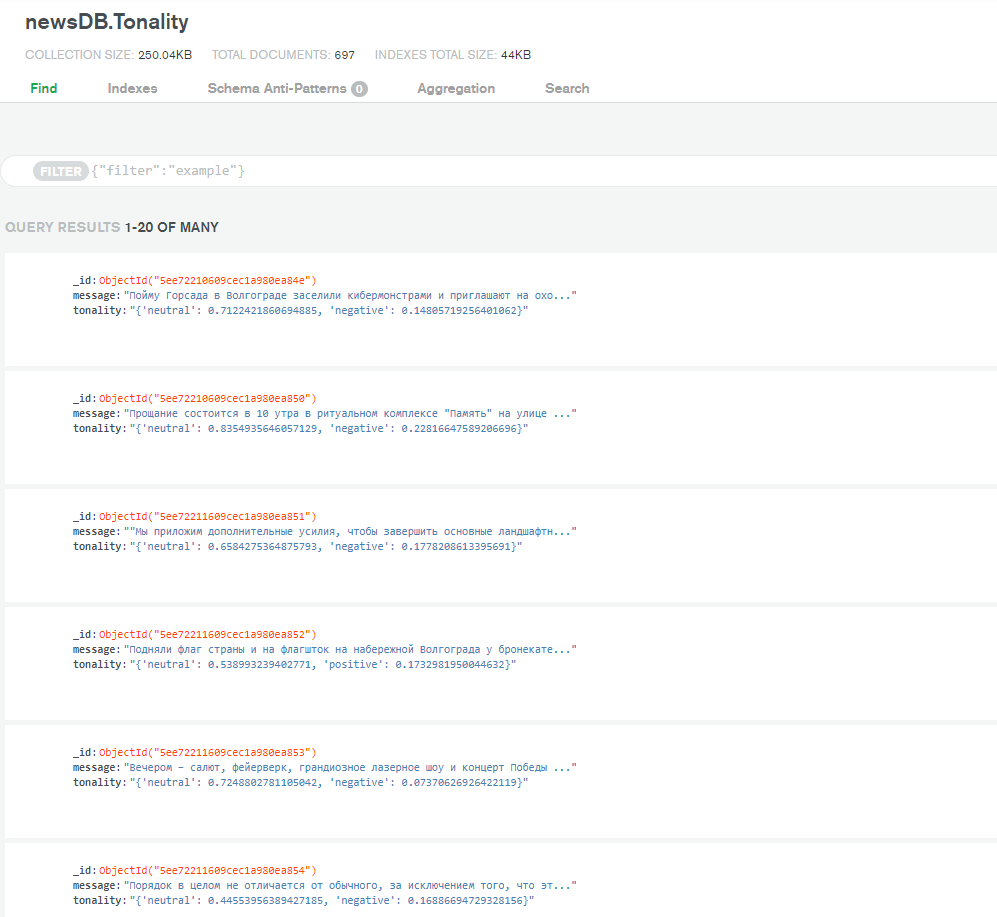


Рисунок 3. Таблица Tonality

Web-интерфейс

Web-интерфейс главной страницы выглядит следующим образом, на нем представлена ссылки на статьи с сайта Volgograd.kp.ru, их названия и содержание:

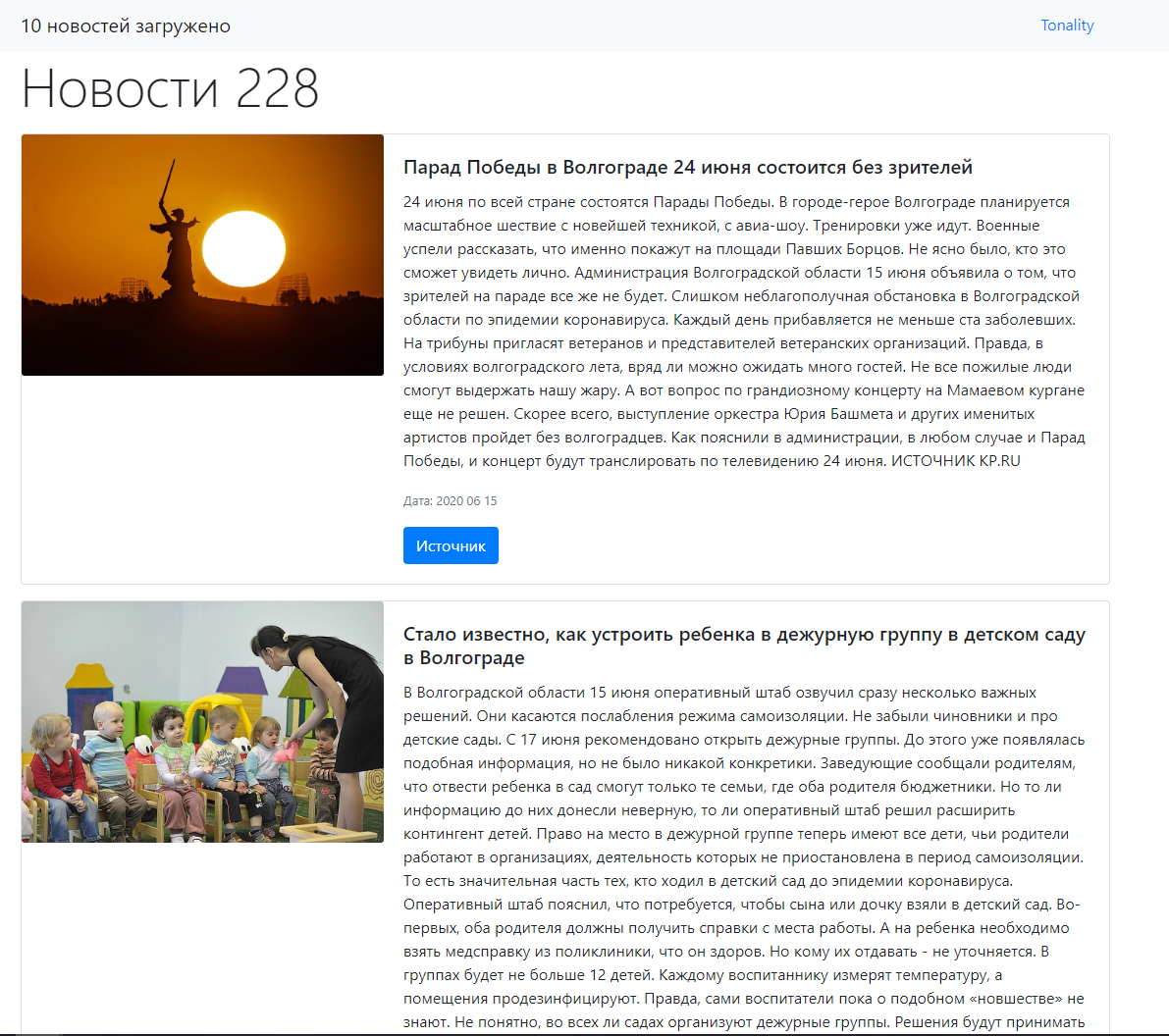


Рисунок 5. Интерфейс главной страницы

# Вторая часть

## Томита-парсер

Грамматики для томита-парсера генерировались на основе данных полученных с сайтов: <https://global-volgograd.ru/person> и https://avolgograd.com/sights?obl=vgg

Выделение персон и достопримечательностей и помещение их в таблицу Tomito выполняется в скрипте crawler.py по ходу записи новостей в коллекцию news.

Перед запуском необходимо его поместить в папку с файлами people.txt, places.txt, dic.gzt и config.proto(По умолчанию: лежат вместе в ветке мастер).

## Word2Vec

Модель word2vec была обучена на новостных статьях из базы данных, объем которой составлял примерно ~6000 статей. Модель находится в папке ./models/w2v/. Всю папку нужно перенести в /home/vagrant/. Запуск модуля осуществляется запуском программы /word2vec/[sinonim.py](https://github.com/asuka1211/ComputationalLinguistics/blob/master/word2vec/sinonim.py). Модуль записывает контекстные синонимы в БД и осуществляет следующий вывод:

'Бочаров Андрей'

бочаров

андрей

губернатор

озвучил

глава

…

Модуль имеет следующий web-интерфейс, на нем представлено имя персоны или достопримечательности с контекстным синонимом:

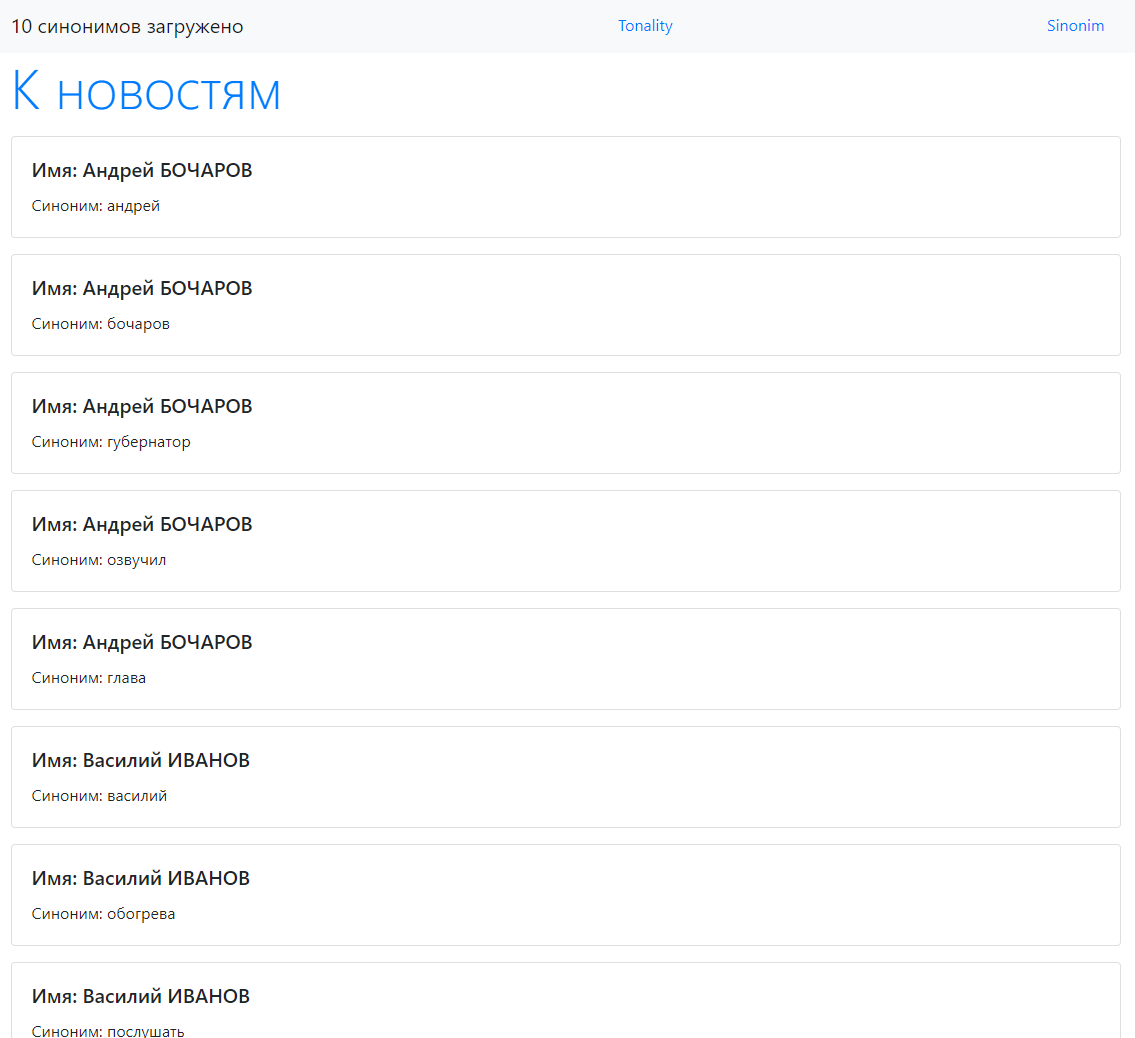


Рисунок 7. Страница контекстных синонимов

# Третья часть

## Анализ тональности

Анализ тональности производится у высказываний/предложений, которые были определены томита-парсером на втором этапе работы. Для определения тональностей предложений была использована библиотека Dostoevsky (https://github.com/bureaucratic-labs/dostoevsky). Её необходимо скачать на свой ПК ($ pip3 install dostoevsky) и установить ($ python3 -m dostoevsky download fasttext-social-network-model). Выполнение модуля осуществляется по ходу записей новостей в коллекцию Tonality модулем crawler.py. Результатом выполнения данного кода является определение тональности каждого высказывания/предложения, которые были выявлены томита-парсером. В web-интерфейсе представлены предложения, выявленные на втором этапе и их тональность:

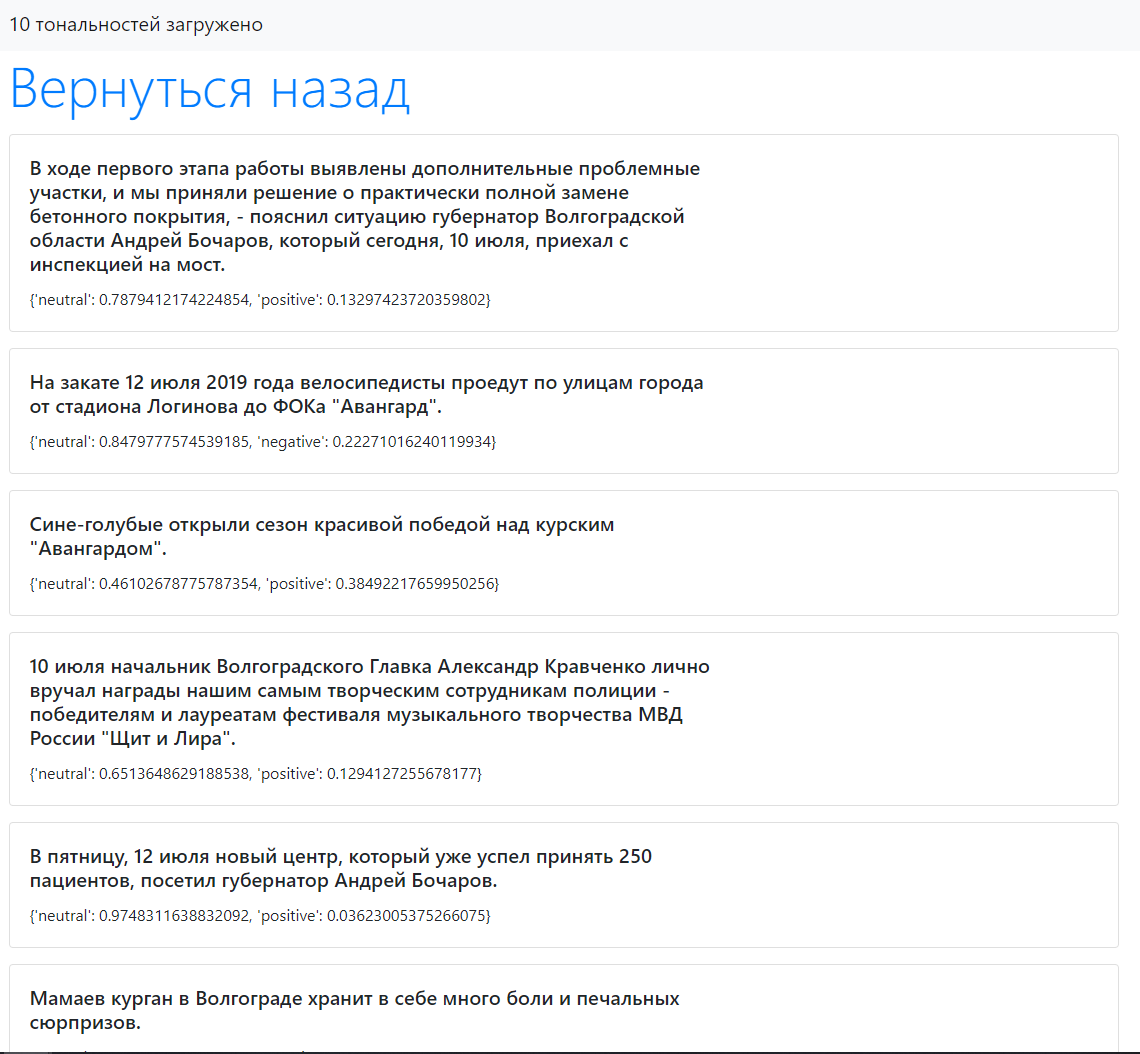


Рисунок 10. Интерфейс предложений и их тональности

Коллекция tonality:



Рисунок 11. tonality

# Руководство системного администратора

## Установка библиотек для Python

Установить python 3.7 и pip (если их нет)

sudo apt install python3.7

python3.7 -m pip install -U pip

Установить библиотеки

python3.7 -m pip install -r requrements.txt

## Установка и настройка PyMongo

Private Key: 40a23d68-ae98-4243-b8f3-55d89a9e3a8f

## Установка томита-парсера

cd ~

git clone <https://github.com/yandex/tomita-parser>

sudo apt-get install build-essential cmake lua5.2

cd tomita-parser && mkdir build && cd build

cmake ../src/ -DCMAKE\_BUILD\_TYPE=Release

make

Переместить файл libmystem\_c\_binding.so в ~/tomita-parser/build/bin/

Установка приложения

Загрузить с github:

git clone

Установка pyspark

mkdir spark245

cd spark245

wget <http://apache-mirror.rbc.ru/pub/apache/spark/spark2.4.5/spark-2.4.5-bin-hadoop2.7.tgz>

sudo tar -zxvf spark-2.4.5-bin-hadoop2.7.tgz

Установим значения переменных среды:

sudo nano /etc/environment

Добавить новую строку JAVA\_HOME="/usr/lib/jvm/java-8-openjdk-amd64" Применим изменения source /etc/environment

Чтобы PySpark запускался с использованием Python 3.x, необходимо создать следующие переменные окружения для текущего пользователя:

source /etc/environment

export PYSPARK\_PYTHON=/usr/bin/python3

export PYSPARK\_DRIVER\_PYTHON=python3

export SPARK\_HOME="/home/vagrant/spark245/spark-2.4.5-binhadoop2.7" export PATH="$SPARK\_HOME/bin:$PATH"

Чтобы создать новые переменные окружения для текущего пользователя, необходимо добавить в конец скрытого файла .bashrc, находящегося в домашнем каталоге пользователя, приведенные выше строки. Для того, чтобы убедиться в корректной установке и запуске PySpark, необходимо выполнить в терминале следующую команду:

source ~/.bashrc

pyspark

# Руководство пользователя

## Работа с web-интерфейсом:

Работа web-интерфейса осуществляется запуском в папке через консоль командой python3 start2.py. После запуска откроется окно (Рисунок 5), на нем можно просмотреть статьи с сайта Volgograd.kp.ru и их названия и перейти. Пользователь может перейти к просмотру тональностей предложений нажав кнопку «Tonality», откроется окно (Рисунок 10), в котором пользователь может видеть предложения выделенных томита парсером.

## Использование парсеров:

Для сбора данных необходимо сначала собрать список ссылок на новости, потом скачивать сами новости.

crawler.py заполняет все поля коллекции news, только новыми новостями, которые раньше присутствовали в базе. Для работы требует указать количество проходов по ленте, selenium будет загружать новые новости указанное количество раз. Дублирование записываемой информации исключается заголовком новости.

Примеры команды запуска:

python3 crawler.py

Данная команда загружает ленту указанное число раз.

Для скачивания новостей используется selenium webdrive, скачивающий новости, которые были найдены. Далее, библиотека Requests открывает найденные ранее ссылки на новости и парсит их содержимое в базу. Дублирование записываемой информации исключается уникальным полем заголовка новости.

## Использование word2vec:

Если проект скачан из репозитория, то в нем уже имеется обученная на ~6000 статей модель (./models/w2v/), готовая к выявлению контекстных синонимов. Если же вы хотите получить свою модель на основе данных из БД, то запустите программу ./word2vec/main.py, начнется процесс подготовки и обучения модели. После окончания обучения вы можете запустить /word2vec/sinonim.py произойдет запрос к файлам достопримечательностей и персон, после чего будет осуществляться нахождение контекстных синонимов этих слов. По окончании работы программы слова и их синонимы выводятся в консоль и записываются в БД.

## Использование анализатора тональности:

Для того, чтобы выявить тональность высказываний, была использована библиотека Dostoevsky (https://github.com/bureaucratic-labs/dostoevsky). Её необходимо скачать на свой ПК ($ pip3 install dostoevsky) и установить ($ python3 -m dostoevsky download fasttext-social-network-model). После установки необходимо запустить код ComputationalLinguistics/tonality.py. Результатом выполнения данного кода является определение тональности каждого высказывания/предложения, которые были выявлены томита-парсером.