Минобрнауки России

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Факультет Электроники и вычислительной техники

Кафедра С[истемы автоматизированного проектирования и поискового конструирования](http://www.vstu.ru/university/fakultety-i-kafedry/fakultet-elektroniki-i-vychislitelnoy-tekhniki/kafedry/kafedra-sapr/)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе (проекту)**

по дисциплине Компьютерная Лингвистика

на тему Программный модуль для анализа новостей из БД \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент Онищенко Владислав Андреевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

Группа ИВТ-364

Руководитель работы (проекта) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Коробкин Д.М.

(подпись и дата подписания) (инициалы и фамилия)

Члены комиссии:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата подписания) (инициалы и фамилия)

Волгоград 2020 г.

Содержание

[1.](#_heading=h.26in1rg) Цель проекта 3

[2.](#_heading=h.lnxbz9) Задача проекта 3

[3.](#_heading=h.35nkun2) Описание парсера 3

[3.1.](#_heading=h.1ksv4uv) Описание полей 4

[4.](#_heading=h.44sinio) Формат хранения информации в базе данных MongoDB 4

[5.](#_heading=h.2jxsxqh) Формат выполнения действий 5

[6.](#_heading=h.z337ya) Результаты выполнения программы: 6

[Заключение 8](#_heading=h.3j2qqm3)

[Список использованной литературы 9](#_heading=h.1y810tw)

[Приложение А 10](#_heading=h.4i7ojhp)

[Программный код модуля 10](#_heading=h.2xcytpi)

1. Цель проекта

Необходимо разработать программный модуль для анализа новостей из БД с помощью Томита-парсера и программный модуль для проведения с помощью Spark MlLib анализа модели word2vec на всем объеме новостных статей из БД.

1. Задача проекта

Написать программу в интегрированной среде разработки PyCharm 2019 Professional на языке python c помощью Томита-парсера, а также встроенных библиотек, которая будет выполнять выделение упоминаний в тексте значимых персон Волгоградской области и достопримечательностей и заносить их в базу данных, а также программный модуль, анализирующий модели word2vec на всех новостях из базы данных Запуск программы осуществить с помощью операционной среды Linux.

1. Описание программного модуля

В проекте создается программный модуль для анализа новостей из БД. Выделение с помощью Томита-парсера упоминаний в тексте значимых персон Волгоградской области и достопримечательностей. Происходит фиксирование в БД предложения с их упоминанием для дальнейшего анализа тональности.

Также создается 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.

Для реализации программы, информация для программного модуля будет браться из базы данных MongoDB в формате json. Для этого необходимо конвертировать хранимую информацию в коллекциях базы данных в файл json.

Команда конвертирования данных: mongoexport -d parserCL -c parser-o C:/test/news.json

Создадим файлы для Томита-парсера, которые будут отвечать за нахождение Персон и Достопримечательностей Волгоградской области.

В них будут хранится словари, с помощью которых будут определяться ФИО персон и имена собственные достопримечательностей.   
Полученные предложения с упоминанием фиксируются в базе данных для дальнейшей работы третьим студентом.

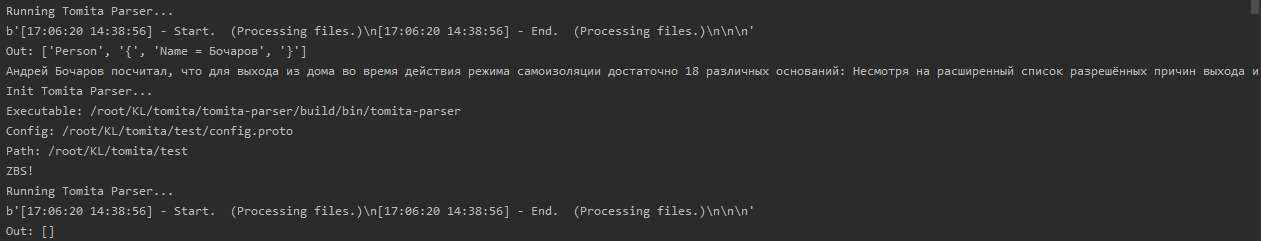


Рисунок 1. Пример выполнения томита-парсера

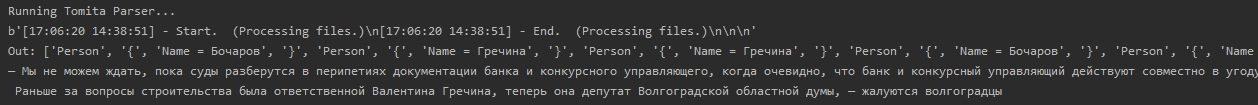


Рисунок 2. Пример выделяемых предложений

## Word2Vec

Модель word2vec была обучена на новостных статьях из базы данных, объем которой составлял примерно 1600 статей. Модель находится в папке ./news\_analyser/w2v/model/.

Запуск модуля осуществляется запуском программы ./news\_analyzer/w2v/main.py.

Модуль записывает контекстные синонимы в БД и осуществляет следующий вывод:

'Бочаров Андрей Иванович'

бочаров

андрей

губернатор

попков

куприков

'------------------------------'

'Мержоева Зина Османовна'

зина

мержоева

вицегубернатор

эксвицегубернатор

выстропов

Заключение

В результате проделанной работы был разработан программный модуль, осуществляющий выделение упоминаний в тексте значимых персон Волгоградской области и достопримечательностей и заносить их в базу данных со страницы новостей и занесение информации в базу данных MongoDB. Разработка осуществлялась с помощью операционной системы Linux, что позволило приобрести ряд навыков для написания проекта.

Список использованной литературы

1. Установка и использование оболочки Linux – Режим доступа: <https://www.comss.ru/page.php?id=4897> (дата обращения 08.06.2020).
2. Основы работы в операционной системе Linux С.А.Немнюгин – Режим доступа: <http://www.cph.phys.spbu.ru/documents/Second/unix_SNv2.pdf> (дата обращения 09.06.2020)
3. Наташа — библиотека для извлечения структурированной информации из текстов на русском языке – Режим доступа: https://habr.com/ru/post/349864/ (дата обращения 12.06.2020)
4. Natasha - Документация <https://natasha.readthedocs.io/ru/latest/> (дата обращения 12.06.2020)
5. Ознакомления с работой с MongoDBиPyMongo – Режим доступа: <https://api.mongodb.com/python/current/tutorial.html> (дата обращения 11.06.2020)

Приложение А

Программный код модуля

import os

import subprocess

import re

import xml.etree.ElementTree as ElementTree

from pymongo import MongoClient

client = MongoClient()

db = client.parserCLV1ru

parser = db.parser

tomita = db.tomita

class TomitaParser(object):

def \_\_init\_\_(self, executable, config, debug=True, validate=False):

self.debug\_mode = debug

self.debug("Init Tomita Parser...")

self.executable = os.path.expanduser(executable)

if not os.path.exists(self.executable):

raise Exception("Tomita executable not found at: %s" % self.executable)

self.debug("Executable: %s" % self.executable)

self.config = os.path.expanduser(config)

if not os.path.exists(self.config):

raise Exception("Config file not found at: %s" % self.config)

self.debug("Config: %s" % self.config)

self.path = self.config[:self.config.rfind("/")]

self.debug("Path: %s" % self.path)

self.debug("ZBS!")

if validate:

self.validate\_config()

def validate\_config(self):

is\_xml = False

with open(self.config, "r") as f:

for line in f.readlines():

line = line.strip()

if line:

if line.startswith("File"):

raise Exception("This library uses STDIN and STDOT for communicating with Tomita Parser, "

"please remove all File = \"...\" from Input and Output sections in config.")

if line.startswith("Format") and "xml" in line:

is\_xml = True

if not is\_xml:

raise Exception("This library working only with XML output, "

"please add \"Format = xml;\" to the Output section.")

def run(self, text = "", with\_facts=True, with\_leads=True):

self.debug("Running Tomita Parser...")

with open('/root/KL/tomita/example/input.txt', 'w', encoding='utf-8') as inputFile:

inputFile.writelines(text)

pipe = subprocess.Popen(

[self.executable, self.config],

stdout=subprocess.PIPE,

stdin=subprocess.PIPE,

stderr=subprocess.PIPE,

cwd=self.path

)

pipe.wait()

out, err = pipe.communicate(input=text.encode("utf-8"))

self.debug(err)

output = []

with open("/root/KL/tomita/example/output.txt", 'r', encoding='utf-8') as outputFile:

readedNews = outputFile.readlines()

appending = False

for news in readedNews:

if "Person" in news or "Place" in news:

appending = True

if appending:

output.append(news.strip())

if "}" in news:

appending = False

return output

def debug(self, text):

if self.debug\_mode:

print(text)

def getPropWihtCoincidence(match, text, start = 0):

out = None

indexMatch = text.find(match, start)

split\_regex = re.compile(r'[.|!|?|…]')

if(indexMatch is not -1):

tt = text[start:]

for prop in split\_regex.split(tt):

ind = prop.find(match)

if ind is not -1:

print(prop)

out = prop

break

return out

if \_\_name\_\_ == "\_\_main\_\_":

config = "/root/KL/tomita/test/config.proto"

for i in parser.find():

tomita = TomitaParser("/root/KL/tomita/tomita-parser/build/bin/tomita-parser", config, debug=True)

output = tomita.run(i["text"])

print("Out:", output)

srt = 0

for s in output:

ind = s.find("Name = ")

if ind is not -1:

prop = getPropWihtCoincidence(s[ind+7:], i['text'], start=srt)

toBD = {

'match': s[ind+7:],

'prop': prop,

}

out = parser.insert\_one(toBD)

srt = ind+7