**Федеральное агентство связи**

**Государственное бюджетное образовательное учреждение высшего образование**

**Ордена Трудового Красного Знамени**

**«Московский технический университет связи и информатики»**

**Кафедра “Математической кибернетики и информационных технологий”**

Отчет по учебной практике

Natural language processing

Выполнила студентка группы БФИ1801

Калатанова Е.С.

Москва 2020

Natural Language Processing (NLP) — одно из направлений искусственного интеллекта, которое работает с анализом, пониманием и генерацией живых языков, обеспечивающее взаимодействие с компьютерами и устно, и письменно, используя естественные языки вместо компьютерных.

NLP применяется в Machine translation (Google Translate), Natural language generation, поисковые системы, спам-фильтры, Sentiment Analysis, чат-боты и др.

Named Entity Recognition (NER) – одна из популярных задач NLP. Суть сводится к извлечению именованных сущностей. Классическими сущностями выступают имена людей, наименования компаний, адреса, географические объекты, даты и т.д., необходимые для понимания текста и решения последующих задач.

Задачами настоящего исследования являлись:

1) Разработка функции на Python, которая принимает строку и число N и выдает “True”, если длина строки больше числа N, “False”, если длина строки меньше числа N.

2) Разработка функции на Python, которая принимает строку и выдает кортеж, который содержит “True”, если данный вопрос связан с количеством баллов на направление, и номер направления, если он указан в вопросе, или “False”, в противном случае.

3) Запуск модели Named Entity Recognition (NER) ner\_rus DeepPavlov.

В целях решения задач было выполнено:

1) Сравнение значения длины строки и числа N, если больше, то выдаем “True”, иначе “False”. Для этого изучено построение функций в Python, работа со строками и условиями.

2) Осуществлен поиск подстроки в строке «Балл» или «балл», при нахождении данной подстроки осуществляется поиск направления с помощью шаблона \d{2}.\d{2}.\d{2}. В случае если найдена только подстрока, то кортеж будет состоять из “True”, если еще и направление, то будет состоять из “True” и номера направления, иначе “False”. Изучены кортеж в Python, регулярные выражения и модуль re, который ищет номер направления с помощью шаблона.

3) Осуществлен запуск и активация виртуального окружения venv, установлен DeepPavlov и модель ner\_rus. Успешная установка позволила работать с моделью с помощью кода, указанного в листинге 3. Изучена работа с виртуальным окружением в Python и запуск модель DeepPavlov.

3. Листинг программы

1)a = input("Введите строку: ")

b = input("Введите число: ")  
 def strt(a,b):  
 if (len(a)>int(b)):  
 d = True  
 else:  
 d = False  
 return d  
 print(strt(a,b))

2)import re

vopros = input("Введите вопрос: ")

def proverka(vopros):

if (vopros.find("балл") != -1) or (vopros.find("Балл") != -1):

d = (True,);

result = re.findall('\d{2}.\d{2}.\d{2}', vopros)

if str(\*result) != "":

d = (True, \*result);

else:

d = (False,);

return d

print(proverka(vopros))

3)python -m venv env  
 .\env\Scripts\activate.bat  
 pip install DeepPavlov   
 python -m deeppavlov install ner\_rus  
 from deeppavlov import configs,build\_model  
 ner\_model = build\_model(configs.ner.ner\_rus, download=True)

4. Результат работы

Результат работы первой программы представлен на рисунке 1.

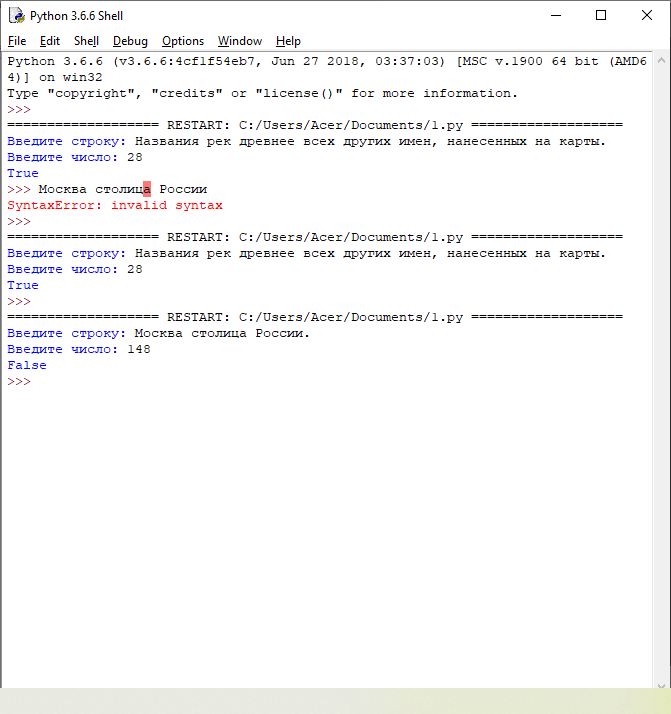


Рисунок 1 – Работа первой программы

Результат работы второй программы представлен на рисунке 2.

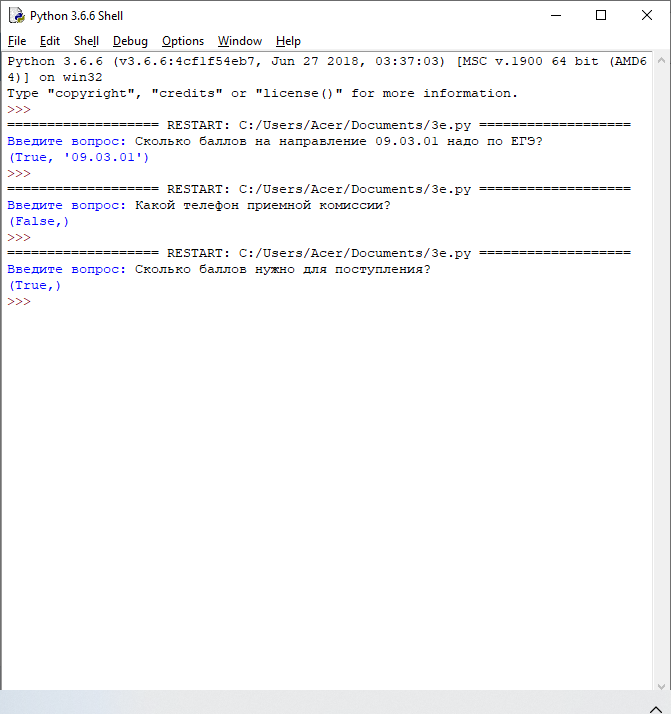


Рисунок 2 – Работа второй программы

Результат работы третьей программы представлен на рисунке 3.

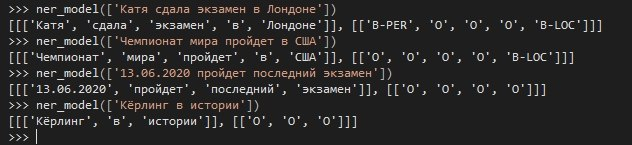


Рисунок 3 – Работа третьей программы

Вывод

Применение программ, извлекающих сущности, из текстовой, звуковой и других источников является перспективным направлением в различных сферах. В ходе проведенного исследования были созданы программы:

Первая программа использовалась для изучения работы Python и не имеет практического назначения;

Вторая программа разработана для приемной комиссии, имела недостаток, заключавшийся в грубом поиске связанных с баллами вопросов;

Для третьей программы использовалась готовая модель ner\_rus, извлекающая сущности, но тестирование показало выявление не всех возможных сущностей.