

Московский государственный технический университет им. Н.Э. Баумана  
Факультет «Информатика и системы управления»  
Кафедра «Системы обработки информации и управления»



**Отчет**  
**Домашняя работа № 2**  
**По курсу «Проектирование интеллектуальных систем»**

Вариант 9

**ИСПОЛНИТЕЛЬ:**

Попов Илья Андреевич  
Группа ИУ5-23М

\_\_\_\_\_

"\_\_" \_\_\_\_\_ 2022 г.

**ПРЕПОДАВАТЕЛЬ:**

Канев А.И.

\_\_\_\_\_

"\_\_" \_\_\_\_\_ 2022 г.

Москва 2022

---

## **Задание**

На основе лабораторной работы 7 или 8 разработать телеграм бот на Python для генерации текста или машинного перевода соответственно.

## **Выполнение**

За основу была взята модель из лабораторной работы 7, задача предсказания следующего слова.

```

def tokenize(text):
    return re.findall("[A-Z]{2,} (?![a-z]) | [A-Z][a-z]+ (=[A-Z]) | [\\'\\w\\-]+", text)

def preprocess(text):
    tokens = [t.lower() for t in tokenize(text)]
    tokens = [SRC_SOS] + tokens + [SRC_EOS]
    src_indexes = [SRC_STOI.get(token, 0) for token in tokens]
    src_tensor = np.int64(src_indexes).reshape(1, -1)
    src_mask = (np.int64(src_indexes) != 1).reshape(1, 1, 1, -1)
    return src_tensor, src_mask

def get_trg_mask(trg_tensor):
    trg_pad_mask = (trg_tensor != 1).reshape(1, 1, 1, -1)
    trg_len = trg_tensor.shape[1]
    trg_sub_mask = np.tril(np.ones((trg_len, trg_len), dtype=np.bool))
    return trg_pad_mask & trg_sub_mask

def Conv1d(message):

def LSTM(message):

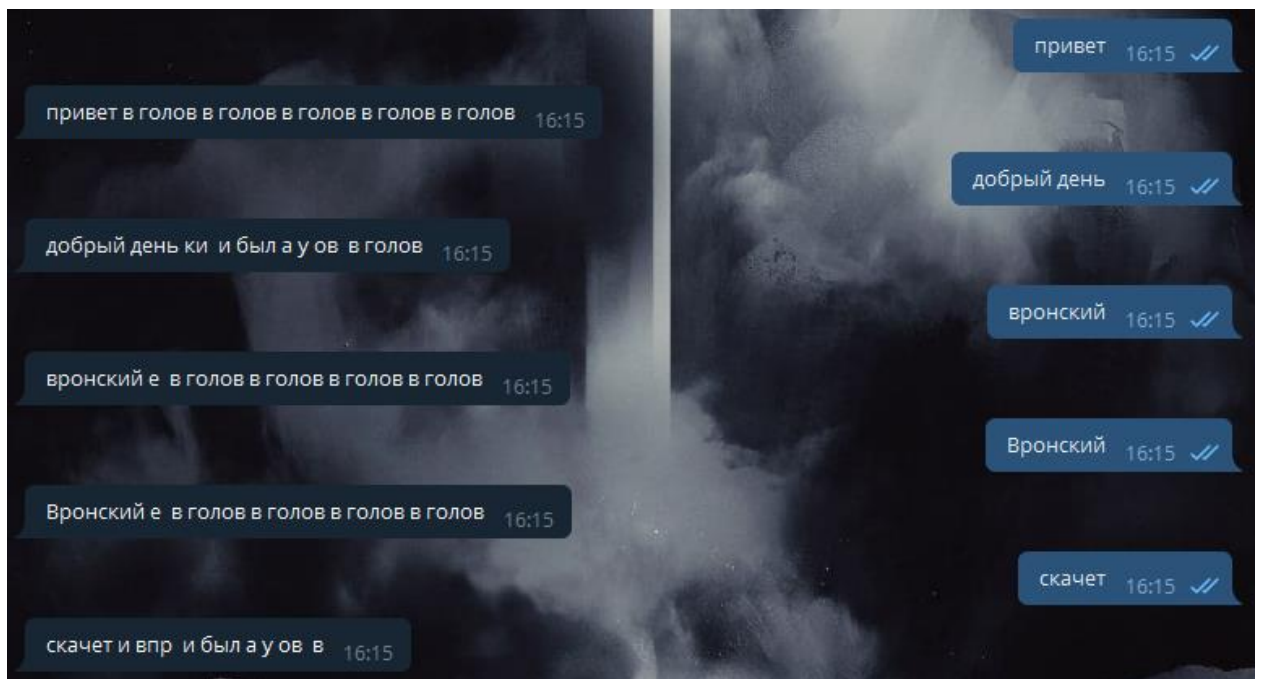
    inputs = sp.encode(message.text)[-16:]
    inputs = [0]*max(16 - len(inputs), 0) + inputs
    finalresult = message.text
    for i in range(100):
        token = sess1.run(None, {'input.1': np.array(inputs, dtype=np.int64).reshape(1, 16),
                                   'onnx::Slice_1': np.zeros((2, 16, 256), dtype=np.float32),
                                   'onnx::Slice_2': np.zeros((2, 16, 256), dtype=np.float32)})[0])
        inputs.pop(0)
        inputs.append(int(token[-1].argmax()))
        print(sp.decode([int(token[-1].argmax())]))
        finalresult = finalresult + ' ' + sp.decode([int(token[-1].argmax())])
    bot.send_message(message.from_user.id, finalresult)

@bot.message_handler(content_types=['text'])
def get_text_messages(message):
    global flag
    flag = 1
    if flag == 0:
        if message.text == "/start":
            bot.send_message(message.from_user.id, "Приветствую! Данный бот предназначен для генерации текста. Для справки введи команду: /help")
        elif message.text == "/help":
            bot.send_message(message.from_user.id, "Для перехода в режим генерации текста введи следующую команду: /text_generator")
        elif message.text == "/text_generator":
            bot.send_message(message.from_user.id, "Напиши текст длиной не более 16 слов, который мне нужно продолжить")
            flag = 1
        else:
            bot.send_message(message.from_user.id, "Не понимаю, что тебе нужно. Для справки введи команду: /help")
    elif flag==1:
        #выбрать нужную модель, остальное закомментировать
        #Conv1d(message)
        LSTM(message)
        #Transformer(message)
        #flag=0

bot.polling(none_stop=True, interval=0)

```

Исходный код Telegram-бота



Результат работы бота