



СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Москат Н.А.

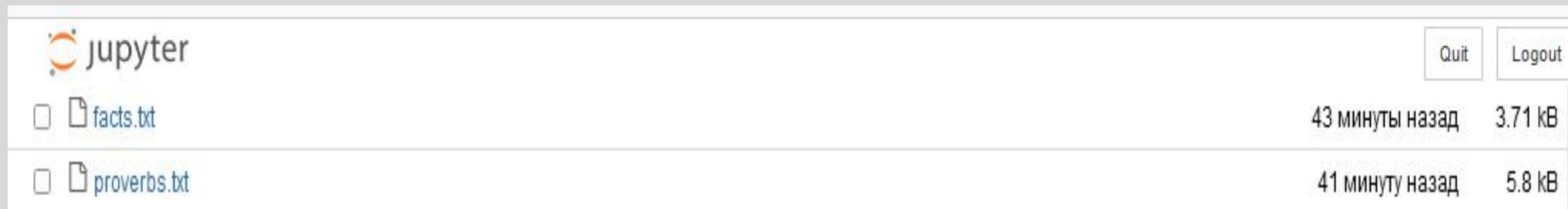
Р

Лекция 2

Разработка чат бота

Чат-бот с двумя виртуальными кнопками

- Во многих Telegram-ботах для выбора каких-то действий используются так называемые **виртуальные кнопки**.
- Реализуем ветвление. Пользователь будет выбирать, что он хочет прочитать (например, интересные факты или поговорки).
- Потребуется создать два файла: `proverbs.txt` и `facts.txt`, которые содержат список поговорок и интересных фактов. Одна строка – одно утверждение.



The screenshot shows the JupyterLab interface. At the top left is the Jupyter logo and the text 'jupyter'. At the top right are 'Quit' and 'Logout' buttons. Below is a file browser showing two files:

File Name	Time	Size
facts.txt	43 минуты назад	3.71 kB
proverbs.txt	41 минуту назад	5.8 kB

Важно! Если использовать для этого бота тот же токен, что и для предыдущего, то, чтобы увидеть кнопки, потребуется перезапуск командой `/start`.

```
In [ ]: import telebot
import random
from telebot import types
# Загружаем список интересных фактов
f = open('facts.txt', 'r', encoding='UTF-8')
facts = f.read().split('\n')
f.close()
# Загружаем список поговорок
f = open('proverbs.txt', 'r', encoding='UTF-8')
thinks = f.read().split('\n')
f.close()
# Создаем бота
bot = telebot.TeleBot('5982403489:AAEH_17FJpis5sTg0r15LSwNQZG0WkFY04A')
# Команда start
@bot.message_handler(commands=["start"])
def start(m, res=False):
    # Добавляем две кнопки
    markup=types.ReplyKeyboardMarkup(resize_keyboard=True)
    item1=types.KeyboardButton("Факт")
    item2=types.KeyboardButton("Поговорка")
    markup.add(item1)
    markup.add(item2)
    bot.send_message(m.chat.id, 'Нажми: \nФакт для получения интересного факта\nПоговорка – для получения мудрой цитаты ',
                      reply_markup=markup)
```



Здесь впиши токен,
полученный от
@botfather

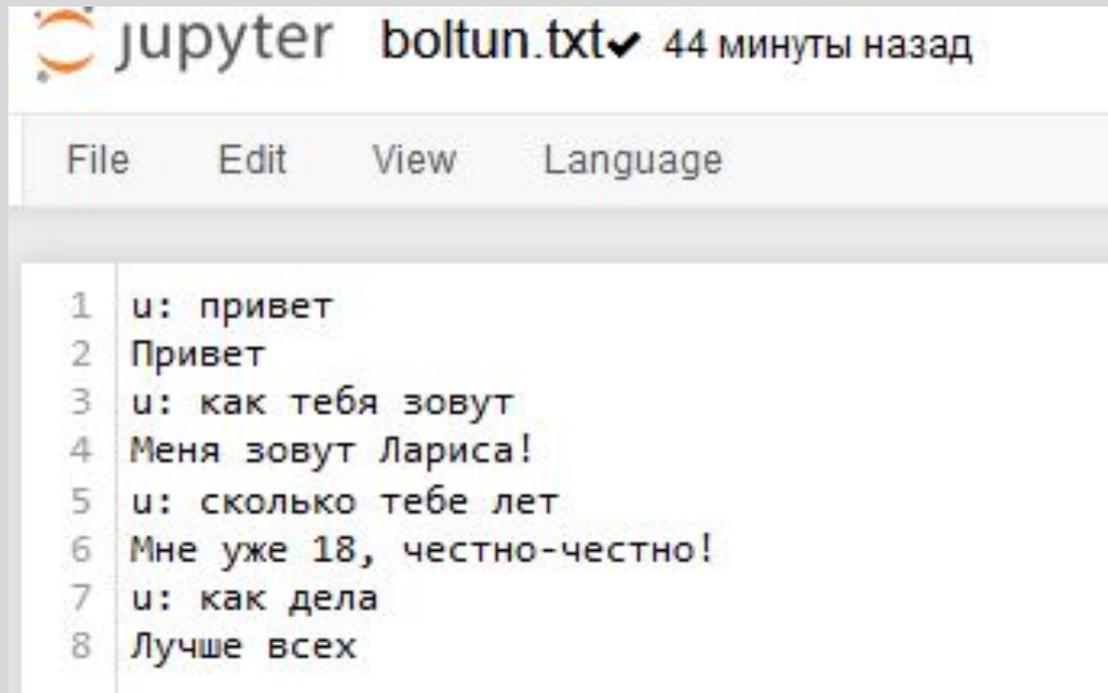
```
# Получение сообщений от юзера
@bot.message_handler(content_types=["text"])
def handle_text(message):
    # Если юзер прислал 1, выдаем ему случайный факт
    if message.text.strip() == 'Факт':
        answer = random.choice(facts)
    # Если юзер прислал 2, выдаем умную мысль
    elif message.text.strip() == 'Поговорка':
        answer = random.choice(thinks)
    # Отсылаем юзеру сообщение в его чат
    bot.send_message(message.chat.id, answer)
# Запускаем бота
bot.polling(none_stop=True, interval=0)
```



10.12.2022

Чат-бот «Лариса»

- Разработаем простейшего чат-бота, который будет болтать с пользователем
- Для этого подготовим файл `boltun.txt`, содержащий строки с вопросами (в начале таких строк поставим метку `u:`) и ответами на них в следующей строке.



The screenshot shows a Jupyter Notebook interface. At the top, the Jupyter logo is followed by the text "jupyter boltun.txt" and a checkmark icon, with "44 минуты назад" (44 minutes ago) to the right. Below this is a menu bar with "File", "Edit", "View", and "Language" options. The main area displays a list of 8 lines of text, each starting with a line number in a light blue column:

```
1 u: привет
2 Привет
3 u: как тебя зовут
4 Меня зовут Лариса!
5 u: сколько тебе лет
6 Мне уже 18, честно-честно!
7 u: как дела
8 Лучше всех
```

- Для поиска похожих вопросов используем модуль `fuzzywuzzy`, который позволяет сравнивать, насколько похожи между собой две строки.
- Данный модуль нужно установить:

```
In [3]: pip install python-Levenshtein
```

```
Collecting python-LevenshteinNote: you may need to restart the kernel to use updated packages.  
  Downloading python_Levenshtein-0.20.8-py3-none-any.whl (9.4 kB)  
Collecting Levenshtein==0.20.8  
  Downloading Levenshtein-0.20.8-cp38-cp38-win_amd64.whl (100 kB)  
Collecting rapidfuzz<3.0.0,>=2.3.0  
  Downloading rapidfuzz-2.13.4-cp38-cp38-win_amd64.whl (1.0 MB)  
Installing collected packages: rapidfuzz, Levenshtein, python-Levenshtein  
Successfully installed Levenshtein-0.20.8 python-Levenshtein-0.20.8 rapidfuzz-2.13.4
```

```
In [4]: pip install fuzzywuzzy
```

```
Collecting fuzzywuzzyNote: you may need to restart the kernel to use updated packages.  
  
  Downloading fuzzywuzzy-0.18.0-py2.py3-none-any.whl (18 kB)  
Installing collected packages: fuzzywuzzy  
Successfully installed fuzzywuzzy-0.18.0
```

Подробнее про теорию работы можно прочитать по ссылке:
<https://habr.com/ru/post/491448/>

- Ниже приведен исходный код бота.
- После его запуска иницилируем диалог с ботом словом «Привет».
- Естественно, это не искусственный интеллект, и набор его ответов ограничен фразами из файла boltun.txt. Однако, чем больше входной файл, тем больше переписка с ботом будет напоминать осмысленный диалог.

```
In [*]: import telebot
import os
from fuzzywuzzy import fuzz
# Создаем бота, пишем свой токен
bot = telebot.TeleBot('5620310966:AAHInbQJejTZWi1vjDo20auWIGY8eQdIzg0')
# Загружаем список фраз и ответов в массив
mas=[]
if os.path.exists('boltun.txt'):
    f=open('boltun.txt', 'r', encoding='UTF-8')
    for x in f:
        if(len(x.strip()) > 2):
            mas.append(x.strip().lower())
    f.close()
```



Здесь впиши токен,
полученный от
@botfather

```
# С помощью fuzzywuzzy вычисляем наиболее похожую фразу и выдаем в качестве ответа следующий элемент списка
def answer(text):
    try:
        text=text.lower().strip()
        if os.path.exists('boltun.txt'):
            a = 0
            n = 0
            nn = 0
            for q in mas:
                if('u: ' in q):
                    # С помощью fuzzywuzzy получаем, насколько похожи две строки
                    aa=(fuzz.token_sort_ratio(q.replace('u: ', ''), text))
                    if(aa > a and aa!= a):
                        a = aa
                        nn = n
                        n = n + 1
            s = mas[nn + 1]
            return s
        else:
            return 'Ошибка'
    except:
        return 'Ошибка'
```

```
# Команда «Старт»
@bot.message_handler(commands=["start"])
def start(m, res=False):
    bot.send_message(m.chat.id, 'Давай поболтаем! Напиши мне Привет )')
# Получение сообщений от юзера
@bot.message_handler(content_types=["text"])
def handle_text(message):
    # Запись ответа
    s=answer(message.text)
    # Отправка ответа
    bot.send_message(message.chat.id, s)
# Запускаем бота
bot.polling(none_stop=True, interval=0)
```



Бот, ведущий Telegram-канал с анекдотами

- Разработаем бота, который работает полностью в автоматическом режиме: по таймеру.
- На вход бот будет получать список анекдотов, которые затем будет отправлять в канал через заданный интервал времени.
- Потребуется создать свой канал в Telegram, добавить в подписчики канала нашего бота и назначить его администратором канала с правом публиковать сообщения.
- Файл с анекдотами должен лежать в папке data рядом со скриптом бота

```
import telebot
import time

# Токен, который выдает @botfather
bot = telebot.TeleBot('Здесь твой токен, полученный от @botfather')
# Адрес телеграм-канала, начинается с @
CHANNEL_NAME = '@адрес_твоего_канала'

# Загружаем список шуток
f = open('data/fun.txt', 'r', encoding='UTF-8')
jokes = f.read().split('\n')
f.close()

# Пока не закончатся шутки, посылаем их в канал
for joke in jokes:
    bot.send_message(CHANNEL_NAME, joke)
    # Делаем паузу в один час
    time.sleep(3600)

bot.send_message(CHANNEL_NAME, "Анекдоты закончились :-(")
```



Пример работы бота, ведущего канал с анекдотами

Итоги

- Разработаны и запущены пять простых ботов, на примере которых научились принимать и отправлять сообщения, реализовывать ветвление, делать кнопки и понимать неточные запросы.

