Модуль 4

4.1 Разработка чат бота

Заключительная часть работы - написание чат бота, для этого сначала загрузим нужные нам библиотеки, чтобы чатбот мог понимать человеческую речь и выдавать нам более-менее адекватные ответы.

In [ ]:

```
#Импортируем библиотеки
import re
import nltk
import random
import string
import bs4 as bs
import numpy as np
import urllib.request

import warnings #игнорирование ошибок
warnings.filterwarnings("ignore")
```

Загружаем с помощью request данные с HTML Вики сайта про ДТП

In [ ]:

```
#Загружаем HTML дату
raw_html = urllib.request.urlopen('https://ru.wikipedia.org/wiki/%D0%94%D
0%BE%D1%80%D0%BE%D0%B6%D0%BD%D0%BE-%D1%82%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D
0%BE%D1%80%D1%82%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B8%D1%81%D1%88%D
0%B5%D1%81%D1%82%D0%B2%D0%B8%D0%B5')
raw_html = raw_html.read()

#Выделяем артикли с HTML страницы с помощью BeautifulSoup с артиклем P
article_html = bs.BeautifulSoup(raw_html, 'lxml')
article_paragraphs = article_html.find_all('p')
article_text = ''

#Цикл добавления всех артиклей p в датасет article_text
for para in article_paragraphs:
    article_text += para.text

article_text = article_text.lower()
```

Помещаем все артикли p в переменную article_text, для последующего редактирования полученного материала.

In [ ]:

```
#Редактируем собранный текст
article_text = re.sub(r'\[[0-9]*\]', ' ', article_text)
```

```
article_text = re.sub(r'\s+', ' ', article_text)
```

Делаем анализ предложений и слов с помощью библиотеки nltk

In [ ]:

```
#Обрабатываем натуральный язык
article_sentences = nltk.sent_tokenize(article_text)
article_words = nltk.word_tokenize(article_text)
```

Обрабатываем функциями токен и документы

In [ ]:

```
wnlemmatizer = nltk.stem.WordNetLemmatizer()
#Функция токенезации
def perform_lemmatization(tokens):
    return [wnlemmatizer.lemmatize(token) for token in tokens]

punctuation_removal = dict((ord(punctuation), None) for punctuation in str
ing.punctuation)
#Функция токенезации
def get_processed_text(document):
    return perform_lemmatization(nltk.word_tokenize(document.lower().trans
late(punctuation_removal)))
```

In [ ]:

```
#Заготовки ввода и вывода
greeting_inputs = ("Привет", "Доброе утро", "Добрый день", "Хай", "Хей",
"Прив", "Ты тут?")
greeting_responses = ["Привет!", "Приветствую", "*nods*", "Хееей!", "Приве
т-привет", "Добро пожаловать, слушаю Вас"]

def generate_greeting_response(greeting):
    for token in greeting.split():
        if token.lower() in greeting_inputs:
            return random.choice(greeting_responses)
```

Догружаем библиотеки

In [ ]:

```
#Импортирование библиотек
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

In [ ]:

```
#Берём новую функцию ввода юзера
def generate_response(user_input):
    tennisrobo_response = ''
    article_sentences.append(user_input)

    word_vectorizer = TfidfVectorizer(tokenizer=get_processed_text, stop_w
ords='english')
    all_word_vectors = word_vectorizer.fit_transform(article_sentences)
```

```
    similar_vector_values = cosine_similarity(all_word_vectors[-1], all_wo
rd_vectors)
    similar_sentence_number = similar_vector_values.argsort()[0][-2]

    matched_vector = similar_vector_values.flatten()
    matched_vector.sort()
    vector_matched = matched_vector[-2]

    if vector_matched == 0:
        tennisrobo_response = tennisrobo_response + "Ой-ой, много информац
ии"
        return tennisrobo_response
    else:
        tennisrobo_response = tennisrobo_response + article_sentences[simi
lar_sentence_number]
        return tennisrobo_response
```

In [ ]:

```
word_vectorizer = TfidfVectorizer(tokenizer=get_processed_text, stop_words
='english')
all_word_vectors = word_vectorizer.fit_transform(article_sentences)
```

In [ ]:

```
similar_vector_values = cosine_similarity(all_word_vectors[-1], all_word_v
ectors)
```

In [ ]:

```
similar_sentence_number = similar_vector_values.argsort()[0][-2]
```

In [ ]:

```
continue_dialogue = True
print("Привет, Меня зовут Чатбот Френдинанд. Задавай мне любой вопрос про
 ДТП")
while(continue_dialogue == True):
    human_text = input()
    human_text = human_text.lower()
    if human_text != 'Пока':
        if human_text == 'Спасибо' or human_text == 'Спасибо тебе большое'
or human_text == 'Благодарю':
            continue_dialogue = False
            print("Френдинанд: Обращайся!")
        else:
            if generate_greeting_response(human_text) != None:
                print("Френдинанд: " + generate_greeting_response(human_te
xt))
            else:
                print("Френдинанд: ", end="")
                print(generate_response(human_text))
                article_sentences.remove(human_text)
    else:
        continue_dialogue = False
        print("Френдинанд: Пока, бери себя...")
```

```
Привет, Меня зовут Чатбот Френдинанд. Задавай мне любой вопро
с про ДТП
Привет, сколько людей пострадало в ДТП в 2020 году?
```

Франдинанд: в 2020 году в японии погибло 2839 человек в дтп.