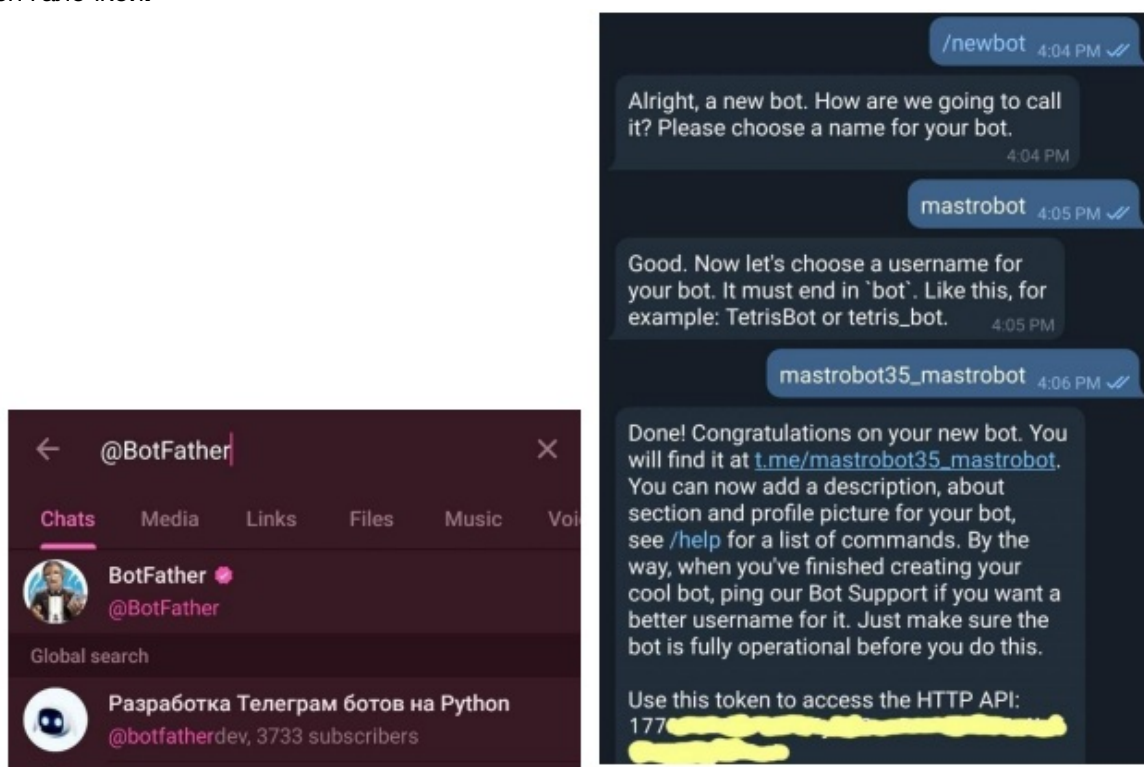


Практическое задание 2_3_5

Создать поискового чат-бота для мессенджера **Telegram**, используя методы **NLP**. Для этого выполните следующие шаги:

Шаг 1. Зарегистрируйте своего бота на канале **@BotFather** и получите ссылку и **API** token вашего чат-бота.

Зайдите в приложение **telegram**. В поисковой строке (знак лупы в верхнем правом углу) напишите **@BotFather**. В найденных результатах найдите ботарегистратора и начните общение с ним. Обратите внимание, что нужный нам отмечен галочкой!



Шаг 2. Найдите научную статью на определенную тему, изучите и подготовьте её для помещения в корпус. Для этого удалить различные спецзнаки, ссылки на другие источники и т.д. Сохраните их в разные файлы формата **.txt**. Для более быстрой работы объем корпуса должен составлять не более **200-250** слов.

In [12]:

```
import os
```

```
text1 = '''Екатерина II (Екатерина Алексеевна; Екатерина Великая; урождённая София Август  
а Фредерика Ангальт-Цербстская, в православии Екатерина Алексеевна; 21 апреля [2 мая] 172  
9, Штеттин, Пруссия – 6 ноября 1796, Зимний дворец, Санкт-Петербург, Российская империя)  
– императрица и Самодержица Всероссийская (1762–1796). Политик просвещённого абсолютизма.  
Дочь князя Ангальт-Цербстского, Екатерина взошла на престол в результате дворцового перев  
орота против своего мужа – Петра III, вскоре погибшего при невыясненных обстоятельствах (   
возможно, он был убит). Она взошла на престол, следуя прецеденту, созданному Екатериной I  
, сменившей своего мужа Петра Великого в 1725 году.  
Екатерининская эпоха ознаменовалась максимальным закрепощением крестьян и всесторонним ра  
сширением привилегий дворянства.  
При Екатерине Великой границы Российской империи были значительно раздвинуты на запад (ра  
зделы Речи Посполитой) и на юг (присоединение Новороссии, Крыма, отчасти Кавказа).  
Система государственного управления при Екатерине Второй впервые со времени Петра I была  
реформирована. Реформы Екатерины II подготовили трансформацию русского государства и обще  
ства в первой четверти XIX века и стали необходимым условием для реформ 1860-х годов.  
В культурном отношении Россия окончательно вошла в число великих европейских держав, чему  
немало способствовала сама императрица, увлекавшаяся литературной деятельностью, собиравш  
ая шедевры живописи и состоявшая в переписке с французскими просветителями. В целом полит  
ика Екатерины II и её реформы вписываются в русло просвещённого абсолютизма XVIII века.
```

```
'''  
text2 = '''София Фредерика Августа Ангальт-Цербстская родилась 21 апреля (2 мая) 1729 год  
а в немецком городе Штеттин — столице Померании (ныне — Щецин, Польша), в доме № 791 на Д  
омштрассе.
```

Отец Кристиан Август Ангальт-Цербстский происходил из цербст-дорнбургской линии Ангальтск
ого дома и состоял на службе у прусского короля, был полковым командиром, комендантом, за
тем губернатором города Штеттина, где будущая императрица и появилась на свет, баллотиров
ался в курляндские герцоги, но неудачно, службу закончил прусским фельдмаршалом. Мать — И
оганна Елизавета, из Готторпского владетельного дома, четвёртая дочь князя Гольштейн-Готт
орпского, после смерти отца воспитывалась при дворе своего дяди, владетельного князя Брау
ншвейга. Приходилась двоюродной тёткой будущему Петру III. Родословная Иоганны Елизаветы
восходит к Кристиану I, королю Дании, Норвегии и Швеции, первому герцогу Шлезвиг-Гольштей
нскому и основателю династии Ольденбургов.

Дядя по материнской линии Адольф-Фридрих был в 1743 году избран в наследники шведского пр
естола, на который он вступил в 1751 году под именем Адольфа-Фредрика. Другой дядя, Карл
Эйтинский, по замыслу Екатерины I, должен был стать мужем её дочери Елизаветы, однако уме
р от оспы в преддверии свадебных торжеств в Санкт-Петербурге.

В семье герцога Цербстского Екатерина получила домашнее образование. Обучалась английском
у, французскому и итальянскому языкам, танцам, музыке, основам истории, географии, богосл
овия. Она росла резвой, любознательной, шаловливой девчонкой, любила шегольнуть своей отв
агой перед мальчишками, с которыми запросто играла на штеттинских улицах. Родители были н
едовольны «мальчишеским» поведением дочери, но их устраивало, что Фредерика заботилась о
младшей сестре Августе. Мать называла её в детстве Фике или Фикхен.

В 1743 году российская императрица Елизавета Петровна, подбирая невесту для своего наслед
ника — великого князя Петра Фёдоровича (будущего русского императора Петра III), вспомнил
а о том, что на смертном одре мать завещала ей стать женой голштинского принца, родного б
рата Иоганны Елизаветы. Елизавета Петровна так пояснила свой выбор: «за лучшее я сочла вз
ять принцессу протестантской веры, и при том из дома, хоть и знатного, но небольшого... Поэ
тому всех пригоднее принцесса Цербская, тем более что она уже в родстве с Голштинским дом
ом». Ранее Елизавета энергично поддержала избрание на шведский престол её дяди, любекског
о епископа Адольфа Фридриха Голштинского, и обменялась портретами с её матерью.

До границы Иоганна Елизавета с дочерью Софией путешествовали инкогнито как графиня Рейнбу
ш с дочерью. Из Берлина они выехали 31 декабря 1743 года и в конце января 1744 года перес
екли русскую границу, где их встретили оружейным салютом и одарили собольими шубами. Пятн
адцатилетняя принцесса с матерью проследовала в Россию через Ригу, где возле дома, в кото
ром они остановились, нёс почётный караул поручик барон фон Мюнхгаузен. В Москву Иоганна
Елизавета и София приехали 9 февраля и успели ко дню рождения великого князя (10 февраля)
. Впервые София увидела своего будущего мужа, Петра Фёдоровича, в Эйтинском замке в 1739
году.

Сразу после приезда в Россию она стала изучать русский язык, историю, православие, русски
е традиции, так как стремилась наиболее полно ознакомиться с Россией, которую воспринимал
а как новую родину. Среди её учителей выделяют известного проповедника Симона Тодорского
(учитель православия), автора первой русской грамматики Василия Ададунова (учитель русско
го языка) и балетмейстера Ланге (учитель танцев).

Стремясь как можно быстрее выучить русский язык, будущая императрица занималась по ночам,
сидя у открытого окна на морозном воздухе. Вскоре она заболела воспалением лёгких, и сост
ояние её было столь тяжёлым, что её мать предложила привести лютеранского пастора. София,
однако, отказалась и послала за Симоном Тодорским. Это обстоятельство прибавило ей популя
рности при русском дворе. 28 июня (9 июля) 1744 года София Фредерика Августа перешла из л
ютеранства в православие и получила имя Екатерины Алексеевны (то же имя и отчество, что и
у матери Елизаветы — Екатерины I), а на следующий день была обручена с будущим императоро
м.

Появление Софии с матерью в Санкт-Петербурге сопровождалось политической интригой, в кото
рой была замешана её мать, княгиня Цербстская. Она была поклонницей короля Пруссии Фридри
ха II, и последний решил использовать её пребывание при русском императорском дворе для у
становления своего влияния на внешнюю политику России. Для этого планировалось, посредств
ом интриг и влияния на императрицу Елизавету Петровну, удалить от дел канцлера Бестужева,
проводившего антипрусскую политику, и заменить его другим вельможей, симпатизировавшим Пр
уссии. Однако Бестужеву удалось перехватить письма княгини Цербстской Фридриху II и предъ
явить их Елизавете Петровне. После того, как последняя узнала о «некрасивой роли прусског
о шпиона», которую играла при её дворе мать, то немедленно изменила к ней своё отношение
и подвергла опале. Однако внешне это не повлияло на положение самой Софии, официально не
принимавшей участия в этой интриге.

```
'''  
if not os.path.exists('newcorpus'):  
    os.mkdir('newcorpus')  
else:  
    pass  
  
f1 = open(os.path.join('newcorpus', 'Kate II.txt'), 'a')  
f1.write(text1)
```

```
f1.close()

f2 = open(os.path.join('newcorpus', 'Kate II Born.txt'), 'a')
f2.write(text2)
f2.close()
```

Шаг 3. Подключите все необходимые функции и библиотеки.

In [13]:

```
import string
import nltk
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from nltk.corpus.reader.plaintext import PlaintextCorpusReader
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

Out[13]:

True

In [18]:

```
try:
    import natasha as nt
    from aiogram import Bot, Dispatcher, executor, types
    import nest_asyncio
    nest_asyncio.apply()
except:
    !pip install natasha
    !pip install aiogram
    !pip install nest_asyncio

    import natasha as nt
    from aiogram import Bot, Dispatcher, executor, types
    import nest_asyncio
    nest_asyncio.apply()
```

Шаг 4. Создайте вспомогательные функции нормализации текста – **Normalize**, которая в качестве своего параметра принимает текст из корпуса, и формирование ответа на запрос – **Response**, которая принимает вопрос от пользователя и формирует ответ на него ссылаясь на информацию из своего корпуса.

In []:

```
#-----Вспомогательные функции-----
#Функция нормализации текста
def Normalize(text):
    #Инициализируем вспомогательные объекты библиотеки natasha
    segmenter = nt.Segmenter()
    morph_vocab = nt.MorphVocab()
    emb = nt.NewsEmbedding()
    morph_tagger = nt.NewsMorphTagger(emb)
    ner_tagger = nt.NewsNERTagger(emb)

    #Убираем знаки пунктуации из текста
    word_token = text.translate(str.maketrans("", "", string.punctuation)).replace("-", " ")

    #Преобразуем очищенный текст в объект Doc и
    doc = nt.Doc(word_token)
    doc.segment(segmenter)
    doc.tag_morph(morph_tagger)
    doc.tag_ner(ner_tagger)

    #Приводим каждое слово к его изначальной форме
    for token in doc.tokens:
        token.lemmatize(morph_vocab)
```

```

resDict = {_.text: _.lemma for _ in doc.tokens}

#Возвращаем результат в виде списка
return [resDict[i] for i in resDict]

def Response(user_response):
    user_response = user_response.lower()
    robo_response = '#Будущий ответ нашего бота'
    sent_tokens.append(user_response) #Временно добавим запрос пользователя в наш корпус.
    TfidfVec = TfidfVectorizer(tokenizer = Normalize) #Вызовем векторизатор TF-IDF
    tfidf = TfidfVec.fit_transform(sent_tokens) #Создадим вектора
    vals = cosine_similarity(tfidf[-1], tfidf) #Через метод косинусного сходства найдем предложение с наилучшим результатом
    idx=vals.argsort()[0][-2] #Запомним индексы этого предложения
    flat = vals.flatten() #сглаживаем полученное косинусное сходство
    flat.sort()
    req_tfidf = flat[-2]
    sent_tokens.remove(user_response)
    if(req_tfidf==0): #Если сглаженное значение будет равно 0, то ответ не был найден
        robo_response=robo_response+"Извините, я не нашел ответа..."
        return robo_response
    else:
        robo_response = robo_response+sent_tokens[idx]
        return robo_response

```

Шаг 5. В основном коде программы (вне функций) сформируйте корпус из подготовленных заранее **txt** файлов (сохраните их рядом с основным **py** файлом в отдельной папке **newcorpus**) и разбейте весь текст на предложения. Сформируйте два списка, которые будут хранить возможные приветствия и прощания пользователя.

In [16]:

```

newcorpus = PlaintextCorpusReader('newcorpus/', r'.*\.txt')

data = newcorpus.raw(newcorpus.fileids())
sent_tokens = nltk.sent_tokenize(data)

welcome_input = ["привет", "ку", "прив", "добрый день", "доброго времени суток", "здравствуй", "приветствую"]
goodbye_input = ["пока", "стоп", "выход", "конец", "до свидания"]

```

Шаг 6. Инициализируйте бота, создав экземпляр класса **Bot** из библиотеки **aiogram** и создайте для него диспетчер – **Dispatcher** (пример есть в лекции). Напишите две асинхронные функции, используя декоратор. Одна из них будет функционировать по ключевому слову **«start»** и будет показывать готовность бота к работе. Вторая функция считывает все остальные сообщения пользователя и определяет содержат ли они слова приветствия и прощания. Если да, то бот должен написать в ответ «Привет!» и «Буду ждать вас!» соответственно. В ином случае функция начинает поиск по корпусу. В конце вызвать команду для запуска бота.

In [17]:

```

bot = Bot(token = 'ВАШ ТОКЕН')
dp = Dispatcher(bot)

@dp.message_handler(commands=['start'])
async def hi_func(message: types.Message):
    await message.answer("Привет!\nНапиши мне что-нибудь!")

@dp.message_handler()
async def search_func(message: types.Message):
    if (message.text).lower() in welcome_input:
        await message.answer('Привет!')
    elif (message.text).lower() in goodbye_input:
        await message.answer('Буду ждать вас!')
    else:
        await message.answer('Дайте подумать...')
        await message.answer(Response(message.text))

```

```
if __name__ == '__main__':  
    executor.start_polling(dp, skip_updates=True)
```

Updates were skipped successfully.
Goodbye!

