NLTK

1. Рассмотрим инструмент NLTK для разбиения текста на предложения. Будем использовать программу `token_.py` с лекции:

```
import nltk
nf ='test_tokrus.txt'   #= input('имя файла  ')
f=open(nf,"r")
sentences = nltk.sent_tokenize(f.read()  , language="russian")
# для английского текста указывать язык не надо (по умолчанию), для других - обязательно
i=1
for sentence in sentences:
    print(i, '  ',sentence)
    print()
    i+=1
```

Рассмотрим работу программы ***для английского языка***.
Так как данный инструмент изначально создавался именно для английского языка, корректность его работы на этом языке высокий:

> *«It's very beautiful!» exclaimed Olga.*
> *His assistant, Mr. Johnson, replied: "I will make sure everything is ready".*
> *Let's see what happens next.*

Однако в некоторых случаях могут возникать проблемы, например, при сокращениях и прямой речи:

> *The screw, the lever, the wedge, the pulley, **etc.** are called simple machines.*
> *She asked: "Do you feel comfortable here**?**" But I didn't know what to say.*

Теперь рассмотрим работу программы ***для русского языка***.
В большинстве случаев, даже трудных, удается правильно разбить текст на предложения (данный ниже текст можно скопировать с фамилиями авторов в квадратных скобках – на результат работы программы это не повлияет):

> [Лермонтов] *«Вот княгиня Литовская, — сказал Грушницкий, — и с нею дочь её Мери, как она её называет на английский манер».*
> [Маршак] *Одна пушкинская строка: «Тяжелёшенько вздохнула» — говорит больше, чем могли бы сказать целые страницы прозы или стихов.*
> [Серж Пьетро] *С 25 млн. лет назад по настоящее время наиболее активно проходили эрозионные процессы в регионах с среднегодовым количеством осадков от 500 мм до 250 мм – в лесостепях, степях и полупустынях.*

Однако при наличии в предложении прямой речи или его окончания на слово “я” возникают трудности. Рассмотрим, например, пару отрывков из произведения “Война и мир” Л.Н. Толстого:

> *Послышался один голос: «Смирно!» Потом, как петухи на заре, повторились голоса в разных концах. И все затихло.*
> *И кто прежде приехал к армии? Император Александр, а не я.*
> *Хотя ему нечего делать при армии.*

При этом если в последнем примере изменить слово “я” на Николай, программа верно поделит текст на предложения:

> *И кто прежде приехал к армии? Император Александр, а не **Николай**. Хотя ему нечего делать при армии.*

Также данный инструмент хорошо справляется и с текстом ***на испанском языке***, например:

> *El Dr. Pérez, un científico reconocido, dijo: "Necesitamos preparar todo el equipo para entonces". Su asistente, el Sr. Rodríguez, respondió: "Me aseguraré de que todo esté listo".*

2. Теперь рассмотрим инструмент NLTK для токенизации. Будем использовать программу `token_word.py` с лекции:

```
1  import nltk
2  sent = input('предложение  ')
3
4  words = nltk.word_tokenize(sent)
5  print(words)
6  print()
```

Рассмотрим работу программы ***для английского языка***.
Как и с разбиением предложений, уровень корректности инструмента на данном языке высокий:

> *When I’m not drawing or reading, I like to listen to music. I have a lots of favorite songs from pop to rock. Music always puts me in a good mood and makes me feel happy and positive.*

Тем не менее, возникают сложные случаи, связанные с современными сокращениями слов:

*Hey dude! Happy bday! Wyd? I **wanna** see ya.*
***C'mon**, don't be so captious! She is a loyal friend.*

Рассмотрим работу программы ***для русского языка***.
Большинство простых предложений удается верно разбить на токены:

> *Будучи глубоким мыслителем, Толстой также написал ряд философских и религиозных работ, в которых отражаются его взгляды на общество, мораль и духовность. Он стал пионером движения за мир и ненасилие, влияя на таких видных фигур, как Ганди.*

Однако могут возникнуть трудности, например, при сокращении инициалов и написании дат:

> ***Л.Н. Толстой** призывает при недовольстве человеком "осуждать его поступки, а его любить".*
> *Сергей Александрович Есенин родился **3 октября 1895**.*

Natasha

1. Рассмотрим инструмент Natasha, специализирующийся на работе с русским языком, для разбиения текста на предложения, используя предложенную на лекции программу и выводя результат на экран:

```
from razdel import sentenize
text='Привет!'      #здесь будет наш текст
print(list(sentenize(text)))
```

Возьмем такие же тексты, какие брали для работы с NLTK:

> *«Вот княгиня Литовская, — сказал Грушницкий, — и с нею дочь её Мери, как она её называет на английский манер».*
> *Одна пушкинская строка: «Тяжелёшенько вздохнула» — говорит больше, чем могли бы сказать целые страницы прозы или стихов.*
> *С 25 млн. лет назад по настоящее время наиболее активно проходили эрозионные процессы в регионах с среднегодовым количеством осадков от 500 мм до 250 мм – в лесостепях, степях и полупустынях.*

> *Послышался один голос: «Смирно!» Потом, как петухи на заре, повторились голоса в разных концах. И все затихло.*

И кто прежде приехал к армии? Император Александр, а не я. Хотя ему нечего делать при армии.

При использовании Natasha все ошибки, которые были в NLTK, исправлены, и тексты верно разбиваются на предложения.

2. Теперь рассмотрим работу данного инструмента для токенизации, используя программу с лекции и выводя результат на экран:

```
from razdel import tokenize
tokens = list(tokenize('Привет!')) #здесь наше предложение
print(tokens)
```

Будем брать такие же предложения, какие брали для работы с NLTK:

Будучи глубоким мыслителем, Толстой также написал ряд философских и религиозных работ, в которых отражаются его взгляды на общество, мораль и духовность. Он стал пионером движения за мир и ненасилие, влияя на таких видных фигур, как Ганди.

***Л.Н. Толстой** призывает при недовольстве человеком "осуждать его поступки, а его любить".*
*Сергей Александрович Есенин родился **3 октября 1895**.*

Однако так же, как и в случае с NLTK, возникают ошибки, связанные с токенизацией ФИО и дат.
Тем не менее, если записать дату в формате ДД.ММ.ГГГГ, программа отработает верно:

*Я родился **01.06.2003**.*

Заключение

NLTK лучше подходит для работы с английским языком, так как изначально исследования для него производились именно на этом языке. Однако этот инструмент может испытывать трудности с прямой речью, сокращениями (в том числе связанными со сленгом) и не только, например, неправильно фиксируя окончание предложения из-за слова “я” на русском языке. **Natasha** же лучше подходит для работы с русским языком, потому что специализируется на этом языке и обеспечивает бóльшую точность на нём. Однако она ограничена в поддержке других языков и так же может иметь трудности с сокращениями слов.