Практическое занятие №2.3.3. Библиотека natasha. Сегментация на токены и предложения, морфологический и синтаксический анализ, лемматизация, извлечение именованных сущностей

Уважаемые слушатели! Проделайте все шаги самостоятельно и прикрепите файл или ссылку на Colab.

Для выполнения задания вам необходимо скачать текст приговора (Приговор.txt).

Извлечение имен и фамилий из текстов на русском языке с помощью библиотеки natasha

Перед началом работы запустите программный код, который позволит загрузить ваш файл с текстом приговора.

```
# Загрузить любой файл с компьютера в google.colab
from google.colab import files
uploaded = files.upload()

for fn in uploaded.keys():
    print('User uploaded file "{name}" with length {length} bytes'.format(
      name=fn, length=len(uploaded[fn])))
```

Экспортируйте библиотеку и пропишите все необходимые строки для дальнейшей обработки текста.

```
!pip install natasha
import natasha as nt

segmenter = nt.Segmenter()
morph_vocab = nt.MorphVocab()

emb = nt.NewsEmbedding()
morph_tagger = nt.NewsMorphTagger(emb)
syntax_parser = nt.NewsSyntaxParser(emb)
ner_tagger = nt.NewsNERTagger(emb)

names_extractor = nt.NamesExtractor(morph_vocab)
dates_extractor = nt.DatesExtractor(morph_vocab)
money_extractor = nt.MoneyExtractor(morph_vocab)
addr_extractor = nt.AddrExtractor(morph_vocab)

file = open('Приговор.txt')
text = file.read()
print(text)
```

Передайте текст библиотеке. Назовите этот объект переменной doc.

```
doc = nt.Doc(text)
doc.segment(segmenter)
doc.tag_morph(morph_tagger)
doc.parse_syntax(syntax_parser)
doc.tag_ner(ner_tagger)
```

1. Сегментация: проверьте, как текст бьется на токены. Распечатайте первые пять.

```
doc.segment(segmenter)
display(doc.tokens[:5])
```

2. Разбейте текст на предложения. Выведите первые пять.

```
display(doc.sents[:5])
#(print – выводит в строчку; display – в столбец)
```

3. Сделайте морфологически разбор слов. Посмотрите на разбор первых пяти слов.

```
doc.tag_morph(morph_tagger)
display(doc.tokens[:5])
```

Библиотека дает полный разбор. У каждого слова указана часть речи, а также, например, в каком числе, роде и т.д. это слово употребляется.
4. Нормализуйте слова и словосочетания – приведите их к правильной форме.

```
for span in doc.spans:
    span.normalize(morph_vocab)
```

```
{_.text: _.normal for _ in doc.spans if _.text != _.normal}
```

5. Приведите каждое слово к начальной форме, то есть лемматизируйте его, используя команду lemmatize.

```
for token in doc.tokens:
    token.lemmatize(morph_vocab)
```

```
{_.text: _.lemma for _ in doc.tokens}
```

6. Извлеките даты из приговора:

```
matches = dates_extractor(text)
facts = [i.fact.as_json for i in matches]
facts
```

7. Приведите даты к более удобной форме с помощью f-строк:

```
for f in facts:
    print(f"{f.get('day')}.{f.get('month')}.{f.get('year')}")
```

8. Извлеките имена в разных написаниях из текста приговора (это фамилия и инициалы).

```
for span in doc.spans:
    if span.type == nt.PER:
        span.extract_fact(names_extractor)
names_dict = {_.normal: _.fact.as_dict for _ in doc.spans
if _.fact}
names_dict
```

9. С помощью команды `keys()` получите все ключи словаря и сделайте из него список.

```
list(names_dict.keys())
```