Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»

Лабораторная работа №5
по дисциплине «Методы машинного обучения»
«Предобработка текста»

ИСПОЛНИТЕЛЬ:

Цветкова Алена
Группа ИУ5-21М

_____

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

_____

Москва, 2021

Цель лабораторной работы: изучение методов предобработки текстов.

In [21]:

```
text = 'На краю дороги стоял дуб. С огромными, неуклюже, несимметрично рас
топыренными корявыми руками и пальцами, он старым, сердитым и презрительны
м уродом стоял между улыбающимися березами. Только он один не хотел подчин
иться обаянию весны и не хотел видеть ни весны, ни солнца. Уже было начало
июня, когда князь Андрей Болконский, возвращаясь домой, въехал опять в ту
 березовую рощу, в которой этот старый, корявый дуб так странно и памятно
 поразил его.'
text
```

Out[21]:

```
'На краю дороги стоял дуб. С огромными, неуклюже, несимметрич
но растопыренными корявыми руками и пальцами, он старым, серд
итым и презрительным уродом стоял между улыбающимися березам
и. Только он один не хотел подчиниться обаянию весны и не хот
ел видеть ни весны, ни солнца. Уже было начало июня, когда кн
язь Андрей Болконский, возвращаясь домой, въехал опять в ту б
ерезовую рощу, в которой этот старый, корявый дуб так странно
и памятно поразил его.'
```

Токенизация

Токенизация — это процесс разбиения текста на текстовые единицы

In [2]:

```
import nltk
from nltk import tokenize
```

In [3]:

```
#nltk.download('punkt')
```

In [4]:

```
nltk_tk = nltk.WordPunctTokenizer()
nltk_tk.tokenize(text)
```

Out[4]:

```
['На',
 'краю',
 'дороги',
 'стоял',
 'дуб',
 '.',
 'С',
 'огромными',
 ',',
 'неуклюже',
 '.',
```

```
 ' ',
 'несимметрично',
 'растопыренными',
 'корявыми',
 'руками',
 'и',
 'пальцами',
 ',',
 'он',
 'старым',
 ',',
 'сердитым',
 'и',
 'презрительным',
 'уродом',
 'стоял',
 'между',
 'улыбающимися',
 'березами',
 '.',
 'Только',
 'он',
 'один',
 'не',
 'хотел',
 'подчиниться',
 'обаянию',
 'весны',
 'и',
 'не',
 'хотел',
 'видеть',
 'ни',
 'весны',
 ',',
 'ни',
 'солнца',
 '.']
```

In [5]:

```
nltk_tk_sents = nltk.tokenize.sent_tokenize(text)
nltk_tk_sents
```

Out[5]:

```
['На краю дороги стоял дуб.',
 'С огромными, неуклюже, несимметрично растопыренными корявым
и руками и пальцами, он старым, сердитым и презрительным урод
ом стоял между улыбающимися березами.',
 'Только он один не хотел подчиниться обаянию весны и не хоте
л видеть ни весны, ни солнца.']
```

In [6]:

```
#pip install spacy
```

In [7]:

```
# !python3 -m spacy download ru_core_news_sm
```

In [8]:

```
from spacy.lang.ru import Russian
import spacy
nlp = spacy.load('ru_core_news_sm')
spacy_text = nlp(text)
spacy_text
```

Out[8]:

```
На краю дороги стоял дуб. С огромными, неуклюже, несимметричн
о растопыренными корявыми руками и пальцами, он старым, серди
тым и презрительным уродом стоял между улыбающимися березами.
Только он один не хотел подчиниться обаянию весны и не хотел
видеть ни весны, ни солнца.
```

In [9]:

```
for t in spacy_text:
    print(t)
```

```
На
краю
дороги
стоял
дуб
.
С
огромными
,
неуклюже
,
несимметрично
растопыренными
корявыми
руками
и
пальцами
,
он
старым
,
сердитым
и
презрительным
уродом
стоял
между
улыбающимися
березами
.
Только
он
один
не
хотел
подчиниться
обаянию
весны
и
не
```

```
хотел
видеть
ни
весны

,
ни
солнца
.
```

Частеречная разметка

In [10]:

```
for token in spacy_text:
    print('{} - {} - {}'.format(token.text, token.pos_, token.dep_))
```

```
На - ADP - case
краю - NOUN - obl
дороги - NOUN - nmod
стоял - VERB - ROOT
дуб - NOUN - nsubj
. - PUNCT - punct
С - ADP - case
огромными - ADJ - amod
, - PUNCT - punct
неуклюже - NOUN - conj
, - PUNCT - punct
несимметрично - ADV - advmod
растопыренными - VERB - acl
корявыми - ADJ - amod
руками - NOUN - obl
и - CCONJ - cc
пальцами - NOUN - conj
, - PUNCT - punct
он - PRON - nsubj
старым - ADJ - conj
, - PUNCT - punct
сердитым - ADJ - conj
и - CCONJ - cc
презрительным - ADJ - conj
уродом - NOUN - obl
стоял - VERB - ROOT
между - ADP - case
улыбающимися - VERB - amod
березами - NOUN - obl
. - PUNCT - punct
Только - PART - advmod
он - PRON - nsubj
один - DET - det
не - PART - advmod
хотел - VERB - ROOT
подчиниться - VERB - xcomp
обаянию - NOUN - iobj
весны - NOUN - nmod
и - CCONJ - cc
не - PART - advmod
хотел - VERB - conj
видеть - VERB - xcomp
```

```
ни - CCONJ - cc
весны - NOUN - obj
, - PUNCT - punct
ни - CCONJ - cc

солнца - NOUN - conj
. - PUNCT - punct
```

Лемматизация

Лемматиза́ция — процесс приведения словоформы к лемме — её нормальной (словарной) форме

In [11]:

```
for token in spacy_text:
      print(token, token.lemma, token.lemma_)
```

```
На 16191904166009283104 на
краю 980890529103078125 край
дороги 10315905627005774972 дорога
стоял 6001902071050492373 стоять
дуб 10563177250858603705 дуб
. 12646065887601541794 .
С 5863529159893111856 с
огромными 8315696299342348519 огромный
, 2593208677638477497 ,
неуклюже 13992719679393433945 неуклюже
, 2593208677638477497 ,
несимметрично 6761246629650065449 несимметрично
растопыренными 4993664938926901233 растопырить
корявыми 15253620947780143884 корявый
руками 18107734691848073173 рука
и 15015917632809974589 и
пальцами 15414162451173240934 палец
, 2593208677638477497 ,
он 7004339974413567607 он
старым 4368933178171963056 старый
, 2593208677638477497 ,
сердитым 7487154488081755156 сердитый
и 15015917632809974589 и
презрительным 6044509452139401437 презрительный
уродом 27125673880894423 67 урод
стоял 6001902071050492373 стоять
между 9001874684165121135 между
улыбающимися 44577887938410679 41 улыбаться
березами 2465664272251865655 берёза
. 12646065887601541794 .
Только 18551102991121890 69 только
он 7004339974413567607 он
один 5834873107654807359 один
не 5319710824202933802 не
хотел 14604981939338786408 хотеть
подчиниться 88597375 4097758102 подчиниться
обаянию 44843931923973176 83 обаяние
весны 7658501378292251113 весна
и 15015917632809974589 и
не 5319710824202933802 не
```

```
хотел 14604981939338786408 хотеть
видеть 11385572989387288387 видеть
ни 10089292569908228859 ни
весны 7658501378292251113 весна

, 2593208677638477497 ,
ни 10089292569908228859 ни
солнца 14046915539579070395 солнце
. 12646065887601541794 .
```

Выделение (распознавание) именованных сущностей

In [27]:

```
text2 = 'Далее действие переносится в Москву, и мы сразу попадаем не прост
о в другой город, а в другой мир. Это прежде всего мир дома Ростовых, откр
ытый для каждого, с его особой любовной атмосферой, с тем высшим накалом р
адости жизни, радушия, гостеприимства, которые всегда будут характерны для
этого дома и которые в полной мере проявятся и тогда, когда Ростовы приеду
т в Петербург.'
```

In [28]:

```
spacy_text2 = nlp(text2)
spacy_text2
```

Out[28]:

```
Далее действие переносится в Москву, и мы сразу попадаем не п
росто в другой город, а в другой мир. Это прежде всего мир до
ма Ростовых, открытый для каждого, с его особой любовной атмо
сферой, с тем высшим накалом радости жизни, радушия, гостепри
имства, которые всегда будут характерны для этого дома и кото
рые в полной мере проявятся и тогда, когда Ростовы приедут в
Петербург.
```

In [30]:

```
for ent in spacy_text2.ents:
    print(ent.text, ent.label_)
```

```
Москву LOC
Ростовых PER
Ростовы PER
Петербург LOC
```

In [31]:

```
print(spacy.explain("LOC"))
```

```
Non-GPE locations, mountain ranges, bodies of water
```

In [32]:

```
print(spacy.explain("PER"))
```

```
Named person or family.
```

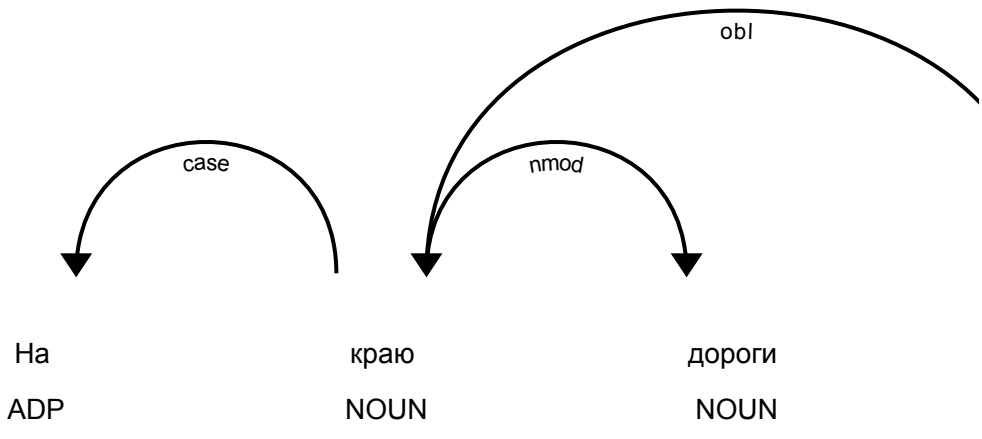
Разбор предложения

In [33]:

```
from spacy import displacy
```

In [34]:

```
displacy.render(spacy_text, style='dep', jupyter=True)
```



In [38]:

```
#pip install natasha
```

In [43]:

```
from natasha import NewsSyntaxParser, NewsEmbedding, Doc, Segmenter, NewsM
```

```
orphTagger, MorphVocab
```

In [44]:

```
def n_lemmatize(text):
    emb = NewsEmbedding()
    morph_tagger = NewsMorphTagger(emb)
    segmenter = Segmenter()
    morph_vocab = MorphVocab()
    doc = Doc(text)
    doc.segment(segmenter)
    doc.tag_morph(morph_tagger)
    for token in doc.tokens:
        token.lemmatize(morph_vocab)
    return doc
```

In [45]:

```
n_doc1 = n_lemmatize(text)
{_.text: _.lemma for _ in n_doc1.tokens}
```

Out[45]:

```
{'На': 'на',
 'краю': 'край',
 'дороги': 'дорога',
 'стоял': 'стоять',
 'дуб': 'дуб',
 '.': '.',
 'С': 'с',
 'огромными': 'огромный',
 ',': ',',
 'неуклюже': 'неуклюже',
 'несимметрично': 'несимметрично',
 'растопыренными': 'растопырить',
 'корявыми': 'корявый',
 'руками': 'рука',
 'и': 'и',
 'пальцами': 'палец',
 'он': 'он',
 'старым': 'старый',
 'сердитым': 'сердитый',
 'презрительным': 'презрительный',
 'уродом': 'урод',
 'между': 'между',
 'улыбающимися': 'улыбаться',
 'березами': 'береза',
 'Только': 'только',
 'один': 'один',
 'не': 'не',
 'хотел': 'хотеть',
 'подчиниться': 'подчиниться',
 'обаянию': 'обаяние',
 'весны': 'весна',
 'видеть': 'видеть',
 'ни': 'ни',
 'солнца': 'солнце',
 'Уже': 'уже',
 'было': 'быть',
 'начало': 'начало',
 'июня': 'июнь',
```

```
 'когда': 'когда',
 'князь': 'князь',
 'Андрей': 'андрей',
 'Болконский': 'болконский',

 'возвращаясь': 'возвращаться',
 'домой': 'домой',
 'въехал': 'въехать',
 'опять': 'опять',
 'в': 'в',
 'ту': 'тот',
 'березовую': 'березовый',
 'рощу': 'роща',
 'которой': 'который',
 'этот': 'этот',
 'старый': 'старый',
 'корявый': 'корявый',
 'так': 'так',
 'странно': 'странный',
 'памятно': 'памятный',
 'поразил': 'поразить',
 'его': 'он'}
```

In [41]:

```
emb = NewsEmbedding()
syntax_parser = NewsSyntaxParser(emb)
```

In [46]:

```
n_doc1.parse_syntax(syntax_parser)
n_doc1.sents[0].syntax.print()
```

```
        ┌─► На      case
   ┌─┌─└── краю    obl
   │ └──►  дороги  nmod
┌──└─────┌ стоял
│        └► дуб     nsubj
└──────►  .       punct
```

In [47]:

```
n_doc1.parse_syntax(syntax_parser)
n_doc1.sents[1].syntax.print()
```

```
┌──────────────►  С                case
│  ┌──────────►  ┌─── огромными       amod
│  │             │ ┌► ,               punct
│  │             └►└─ неуклюже        conj
│  │               ┌► ,               punct
│  │               └─ несимметрично   acl
│  │  ┌─┌─┌─┌─┌──►  растопыренными    amod
│  │  │ │ │ │ │ ┌►  корявыми          amod
│  │  │ └►└─└─└─└─  руками            obl
│  │  │     ┌►  и                    cc
│  │  │     └──►  пальцами            conj
│  │  └─────────►  ,                  punct
│  │  ┌──────────►  он                nsubj
│  │  │ ┌►┌─┌─┌──  старым             amod
│  │  │ │ │ │ └►  ,                   punct
│  │  │ │ │ └►└─  сердитым            conj
```

```
                    и               cc
                    презрительным   conj
                    уродом          obl
                    стоял
                    между           case
                    улыбающимися    amod
                    березами        obl
                    .               punct
```

In [48]:

```
n_doc1.parse_syntax(syntax_parser)
n_doc1.sents[2].syntax.print()
```

```
          Только       advmod
          он           nsubj
          один         nsubj
          не           advmod
          хотел
          подчиниться  xcomp
          обаянию      iobj
          весны        obj
          и            cc
          не           advmod
          хотел        conj
          видеть       xcomp
          ни           cc
          весны        iobj
          ,            punct
          ни           cc
          солнца       conj
          .            punct
```

In [ ]:

```

```