# Задание лабораторной работы

Для произвольного предложения или текста решить следующие задачи:

* Токенизация.
* Частеречная разметка.
* Лемматизация.
* Выделение (распознавание) именованных сущностей.
* Разбор предложения.

# Выполнение работы

Исходный текст:

text = 'Во дворе флигель патологоанатома с вывеской над дверью: "Добро пожаловать! - Эй, новенький! Заходи! - закричал он мне, приветливо размахивая топором.'

## Токенизация

!pip install nltk

Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)  
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)  
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.4.2)  
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2024.5.15)  
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.4)

import nltk  
from nltk.tokenize import punkt  
nltk.download('punkt')

[nltk\_data] Downloading package punkt to /root/nltk\_data...  
[nltk\_data] Unzipping tokenizers/punkt.zip.

True

from nltk import tokenize  
dir(tokenize)[:18]

['BlanklineTokenizer',  
 'LegalitySyllableTokenizer',  
 'LineTokenizer',  
 'MWETokenizer',  
 'NLTKWordTokenizer',  
 'PunktSentenceTokenizer',  
 'RegexpTokenizer',  
 'ReppTokenizer',  
 'SExprTokenizer',  
 'SpaceTokenizer',  
 'StanfordSegmenter',  
 'SyllableTokenizer',  
 'TabTokenizer',  
 'TextTilingTokenizer',  
 'ToktokTokenizer',  
 'TreebankWordDetokenizer',  
 'TreebankWordTokenizer',  
 'TweetTokenizer']

nltk\_tk = nltk.WordPunctTokenizer()  
nltk\_tk.tokenize(text)

['Во',  
 'дворе',  
 'флигель',  
 'патологоанатома',  
 'с',  
 'вывеской',  
 'над',  
 'дверью',  
 ':',  
 '"',  
 'Добро',  
 'пожаловать',  
 '!',  
 '-',  
 'Эй',  
 ',',  
 'новенький',  
 '!',  
 'Заходи',  
 '!',  
 '-',  
 'закричал',  
 'он',  
 'мне',  
 ',',  
 'приветливо',  
 'размахивая',  
 'топором',  
 '.']

Токенизация по предложениям:

nltk\_tk\_sents = nltk.tokenize.sent\_tokenize(text)  
print(len(nltk\_tk\_sents))  
nltk\_tk\_sents

4

['Во дворе флигель патологоанатома с вывеской над дверью: "Добро пожаловать!',  
 '- Эй, новенький!',  
 'Заходи!',  
 '- закричал он мне, приветливо размахивая топором.']

!pip install razdel

Collecting razdel  
 Downloading razdel-0.5.0-py3-none-any.whl (21 kB)  
Installing collected packages: razdel  
Successfully installed razdel-0.5.0

from razdel import tokenize, sentenize

n\_tok\_text = list(tokenize(text))  
n\_tok\_text

[Substring(0, 2, 'Во'),  
 Substring(3, 8, 'дворе'),  
 Substring(9, 16, 'флигель'),  
 Substring(17, 32, 'патологоанатома'),  
 Substring(33, 34, 'с'),  
 Substring(35, 43, 'вывеской'),  
 Substring(44, 47, 'над'),  
 Substring(48, 54, 'дверью'),  
 Substring(54, 55, ':'),  
 Substring(56, 57, '"'),  
 Substring(57, 62, 'Добро'),  
 Substring(63, 73, 'пожаловать'),  
 Substring(73, 74, '!'),  
 Substring(75, 76, '-'),  
 Substring(77, 79, 'Эй'),  
 Substring(79, 80, ','),  
 Substring(81, 90, 'новенький'),  
 Substring(90, 91, '!'),  
 Substring(92, 98, 'Заходи'),  
 Substring(98, 99, '!'),  
 Substring(100, 101, '-'),  
 Substring(102, 110, 'закричал'),  
 Substring(111, 113, 'он'),  
 Substring(114, 117, 'мне'),  
 Substring(117, 118, ','),  
 Substring(119, 129, 'приветливо'),  
 Substring(130, 140, 'размахивая'),  
 Substring(141, 148, 'топором'),  
 Substring(148, 149, '.')]

[\_.text for \_ in n\_tok\_text]

['Во',  
 'дворе',  
 'флигель',  
 'патологоанатома',  
 'с',  
 'вывеской',  
 'над',  
 'дверью',  
 ':',  
 '"',  
 'Добро',  
 'пожаловать',  
 '!',  
 '-',  
 'Эй',  
 ',',  
 'новенький',  
 '!',  
 'Заходи',  
 '!',  
 '-',  
 'закричал',  
 'он',  
 'мне',  
 ',',  
 'приветливо',  
 'размахивая',  
 'топором',  
 '.']

n\_sen\_text = list(sentenize(text))  
n\_sen\_text

[Substring(0,  
 74,  
 'Во дворе флигель патологоанатома с вывеской над дверью: "Добро пожаловать!'),  
 Substring(75, 91, '- Эй, новенький!'),  
 Substring(92,  
 149,  
 'Заходи! - закричал он мне, приветливо размахивая топором.')]

[\_.text for \_ in n\_sen\_text], len([\_.text for \_ in n\_sen\_text])

(['Во дворе флигель патологоанатома с вывеской над дверью: "Добро пожаловать!',  
 '- Эй, новенький!',  
 'Заходи! - закричал он мне, приветливо размахивая топором.'],  
 3)

Токенизация для последующей обработки:

def n\_sentenize(text):  
 n\_sen\_chunk = []  
 for sent in sentenize(text):  
 tokens = [\_.text for \_ in tokenize(sent.text)]  
 n\_sen\_chunk.append(tokens)  
 return n\_sen\_chunk

n\_sen\_chunk = n\_sentenize(text)  
n\_sen\_chunk

[['Во',  
 'дворе',  
 'флигель',  
 'патологоанатома',  
 'с',  
 'вывеской',  
 'над',  
 'дверью',  
 ':',  
 '"',  
 'Добро',  
 'пожаловать',  
 '!'],  
 ['-', 'Эй', ',', 'новенький', '!'],  
 ['Заходи',  
 '!',  
 '-',  
 'закричал',  
 'он',  
 'мне',  
 ',',  
 'приветливо',  
 'размахивая',  
 'топором',  
 '.']]

## Частеричная разметка

!pip install navec  
!pip install slovnet

Collecting navec  
 Downloading navec-0.10.0-py3-none-any.whl (23 kB)  
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from navec) (1.25.2)  
Installing collected packages: navec  
Successfully installed navec-0.10.0  
Collecting slovnet  
 Downloading slovnet-0.6.0-py3-none-any.whl (46 kB)  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 46.7/46.7 kB 1.7 MB/s eta 0:00:00  
ent already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from slovnet) (1.25.2)  
Requirement already satisfied: razdel in /usr/local/lib/python3.10/dist-packages (from slovnet) (0.5.0)  
Requirement already satisfied: navec in /usr/local/lib/python3.10/dist-packages (from slovnet) (0.10.0)  
Installing collected packages: slovnet  
Successfully installed slovnet-0.6.0

from navec import Navec  
from slovnet import Morph

# Файл необходимо скачать по ссылке https://github.com/natasha/navec#downloads  
navec = Navec.load('navec\_news\_v1\_1B\_250K\_300d\_100q.tar')

# Файл необходимо скачать по ссылке https://github.com/natasha/slovnet#downloads  
n\_morph = Morph.load('slovnet\_morph\_news\_v1.tar', batch\_size=4)

morph\_res = n\_morph.navec(navec)

def print\_pos(markup):  
 for token in markup.tokens:  
 print('{} - {}'.format(token.text, token.tag))

n\_text\_markup = list(\_ for \_ in n\_morph.map(n\_sen\_chunk))  
[print\_pos(x) for x in n\_text\_markup]

Во - ADP  
дворе - NOUN|Animacy=Inan|Case=Loc|Gender=Masc|Number=Sing  
флигель - NOUN|Animacy=Inan|Case=Nom|Gender=Masc|Number=Sing  
патологоанатома - NOUN|Animacy=Anim|Case=Gen|Gender=Masc|Number=Sing  
с - ADP  
вывеской - NOUN|Animacy=Inan|Case=Ins|Gender=Fem|Number=Sing  
над - ADP  
дверью - NOUN|Animacy=Inan|Case=Ins|Gender=Fem|Number=Sing  
: - PUNCT  
" - PUNCT  
Добро - NOUN|Animacy=Inan|Case=Nom|Gender=Neut|Number=Sing  
пожаловать - VERB|Aspect=Perf|VerbForm=Inf|Voice=Act  
! - PUNCT  
- - PUNCT  
Эй - PROPN|Animacy=Anim|Case=Nom|Gender=Masc|Number=Sing  
, - PUNCT  
новенький - ADJ|Case=Nom|Degree=Pos|Gender=Masc|Number=Sing  
! - PUNCT  
Заходи - NOUN|Animacy=Inan|Case=Nom|Gender=Neut|Number=Sing  
! - PUNCT  
- - PUNCT  
закричал - VERB|Aspect=Perf|Gender=Masc|Mood=Ind|Number=Sing|Tense=Past|VerbForm=Fin|Voice=Act  
он - PRON|Case=Nom|Gender=Masc|Number=Sing|Person=3  
мне - PRON|Case=Dat|Number=Sing|Person=1  
, - PUNCT  
приветливо - VERB|Aspect=Imp|Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act  
размахивая - VERB|Aspect=Imp|Tense=Pres|VerbForm=Conv|Voice=Act  
топором - NOUN|Animacy=Inan|Case=Ins|Gender=Masc|Number=Sing  
. - PUNCT

[None, None, None]

## Лемматизация

!pip install natasha  
!pip install setuptools

Collecting natasha  
 Downloading natasha-1.6.0-py3-none-any.whl (34.4 MB)  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 34.4/34.4 MB 15.0 MB/s eta 0:00:00  
orphy2 (from natasha)  
 Downloading pymorphy2-0.9.1-py3-none-any.whl (55 kB)  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 55.5/55.5 kB 6.4 MB/s eta 0:00:00  
ent already satisfied: razdel>=0.5.0 in /usr/local/lib/python3.10/dist-packages (from natasha) (0.5.0)  
Requirement already satisfied: navec>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from natasha) (0.10.0)  
Requirement already satisfied: slovnet>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from natasha) (0.6.0)  
Collecting yargy>=0.16.0 (from natasha)  
 Downloading yargy-0.16.0-py3-none-any.whl (33 kB)  
Collecting ipymarkup>=0.8.0 (from natasha)  
 Downloading ipymarkup-0.9.0-py3-none-any.whl (14 kB)  
Collecting intervaltree>=3 (from ipymarkup>=0.8.0->natasha)  
 Downloading intervaltree-3.1.0.tar.gz (32 kB)  
 Preparing metadata (setup.py) ... ent already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from navec>=0.9.0->natasha) (1.25.2)  
Collecting dawg-python>=0.7.1 (from pymorphy2->natasha)  
 Downloading DAWG\_Python-0.7.2-py2.py3-none-any.whl (11 kB)  
Collecting pymorphy2-dicts-ru<3.0,>=2.4 (from pymorphy2->natasha)  
 Downloading pymorphy2\_dicts\_ru-2.4.417127.4579844-py2.py3-none-any.whl (8.2 MB)  
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.2/8.2 MB 61.5 MB/s eta 0:00:00  
 pymorphy2->natasha)  
 Downloading docopt-0.6.2.tar.gz (25 kB)  
 Preparing metadata (setup.py) ... ent already satisfied: sortedcontainers<3.0,>=2.0 in /usr/local/lib/python3.10/dist-packages (from intervaltree>=3->ipymarkup>=0.8.0->natasha) (2.4.0)  
Building wheels for collected packages: docopt, intervaltree  
 Building wheel for docopt (setup.py) ... e=docopt-0.6.2-py2.py3-none-any.whl size=13706 sha256=54001ed5c25e287f05e169b3ba10266cde66d1bc19c6181d4b12b3fc14485584  
 Stored in directory: /root/.cache/pip/wheels/fc/ab/d4/5da2067ac95b36618c629a5f93f809425700506f72c9732fac  
 Building wheel for intervaltree (setup.py) ... e=intervaltree-3.1.0-py2.py3-none-any.whl size=26096 sha256=8007d7ec0a28f58910619a5edc0c3e088288acd2b3fd6499d64897fe492a007c  
 Stored in directory: /root/.cache/pip/wheels/fa/80/8c/43488a924a046b733b64de3fac99252674c892a4c3801c0a61  
Successfully built docopt intervaltree  
Installing collected packages: pymorphy2-dicts-ru, docopt, dawg-python, pymorphy2, intervaltree, yargy, ipymarkup, natasha  
Successfully installed dawg-python-0.7.2 docopt-0.6.2 intervaltree-3.1.0 ipymarkup-0.9.0 natasha-1.6.0 pymorphy2-0.9.1 pymorphy2-dicts-ru-2.4.417127.4579844 yargy-0.16.0  
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (67.7.2)

from natasha import Doc, Segmenter, NewsEmbedding, NewsMorphTagger, MorphVocab

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

n\_doc = n\_lemmatize(text)  
{\_.text: \_.lemma for \_ in n\_doc.tokens}

{'Во': 'в',  
 'дворе': 'двор',  
 'флигель': 'флигель',  
 'патологоанатома': 'патологоанатом',  
 'с': 'с',  
 'вывеской': 'вывеска',  
 'над': 'над',  
 'дверью': 'дверь',  
 ':': ':',  
 '"': '"',  
 'Добро': 'добро',  
 'пожаловать': 'пожаловать',  
 '!': '!',  
 '-': '-',  
 'Эй': 'эй',  
 ',': ',',  
 'новенький': 'новенький',  
 'Заходи': 'заходить',  
 'закричал': 'закричать',  
 'он': 'он',  
 'мне': 'я',  
 'приветливо': 'приветливо',  
 'размахивая': 'размахивать',  
 'топором': 'топор',  
 '.': '.'}

## Выделение (распознавание) именованных сущностей

!pip install ipymarkup

Requirement already satisfied: ipymarkup in /usr/local/lib/python3.10/dist-packages (0.9.0)  
Requirement already satisfied: intervaltree>=3 in /usr/local/lib/python3.10/dist-packages (from ipymarkup) (3.1.0)  
Requirement already satisfied: sortedcontainers<3.0,>=2.0 in /usr/local/lib/python3.10/dist-packages (from intervaltree>=3->ipymarkup) (2.4.0)

from slovnet import NER  
from ipymarkup import show\_span\_ascii\_markup as show\_markup

# Файл необходимо скачать по ссылке https://github.com/natasha/slovnet#downloads  
ner = NER.load('slovnet\_ner\_news\_v1.tar')  
ner\_res = ner.navec(navec)  
markup\_ner = ner(text)

markup\_ner

SpanMarkup(  
 text='Во дворе флигель патологоанатома с вывеской над дверью: "Добро пожаловать! - Эй, новенький! Заходи! - закричал он мне, приветливо размахивая топором.',  
 spans=[]  
)

show\_markup(markup\_ner.text, markup\_ner.spans)

Во дворе флигель патологоанатома с вывеской над дверью: "Добро   
пожаловать! - Эй, новенький! Заходи! - закричал он мне, приветливо   
размахивая топором.

## Разбор предложения

from natasha import NewsSyntaxParser

emb = NewsEmbedding()  
syntax\_parser = NewsSyntaxParser(emb)

n\_doc.parse\_syntax(syntax\_parser)  
n\_doc.sents[0].syntax.print()

┌► Во case  
┌───┌─┌►└─ дворе nmod  
│ └►└─┌─ флигель nmod  
│ ┌─└► патологоанатома nmod  
│ │ ┌► с case  
│ ┌─└►└─ вывеской nmod  
│ │ ┌► над case  
│ ┌─└──►└─ дверью nmod  
│ │ ┌──► : punct  
│ │ │ ┌► " punct  
│ └──►└─└─ Добро parataxis  
│ └──► пожаловать advmod  
└────────► ! punct