Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



**Лабораторная работа №5**

**По курсу «Методы машинного обучения»**

**ИСПОЛНИТЕЛЬ:**

Лю Цзыцзянь

Группа ИУ5И-21М

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПРОВЕРИЛ:**

Гапанюк Ю.Е.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оглавление

[Лабораторная работа №5 3](#_Toc200476128)

[Токенизация. NLTK 4](#_Toc200476129)

[Частеречная разметка. Natasha 6](#_Toc200476130)

[Лемматизация. Natasha 9](#_Toc200476131)

[Выделение (распознавание) именованных сущностей. Natasha 11](#_Toc200476132)

[Разбор предложения. Natasha 13](#_Toc200476133)

# Лабораторная работа №5

**Предобработка текста.**

**Цель лабораторной работы:** изучение методов предобработки текстов.

**Задание:**

Для произвольного предложения или текста решите следующие задачи:

1. Токенизация.
2. Частеречная разметка.
3. Лемматизация.
4. Выделение (распознавание) именованных сущностей.
5. Разбор предложения.

!pip install numpy pandas scikit-surprise sklearn seaborn matplotlib spacy nltk navec slovnet natasha

# Токенизация. NLTK

In [ ]:

import nltk

nltk.download('punkt')

text1 = 'Предобработка данных в XML файле.'

text2 = 'Меня зовут Бонд. Джеймс Бонд'

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

[nltk\_data] Package punkt is already up-to-date!

In [ ]:

from nltk import tokenize

dir(tokenize)[:18]

Out[ ]:

['BlanklineTokenizer',

'LineTokenizer',

'MWETokenizer',

'PunktSentenceTokenizer',

'RegexpTokenizer',

'ReppTokenizer',

'SExprTokenizer',

'SpaceTokenizer',

'StanfordSegmenter',

'TabTokenizer',

'TextTilingTokenizer',

'ToktokTokenizer',

'TreebankWordTokenizer',

'TweetTokenizer',

'WhitespaceTokenizer',

'WordPunctTokenizer',

'\_\_builtins\_\_',

'\_\_cached\_\_']

In [ ]:

nltk\_tk\_1 = nltk.WordPunctTokenizer()

nltk\_word = nltk\_tk\_1.tokenize(text1)

print(nltk\_word)

['Предобработка', 'данных', 'в', 'XML', 'файле', '.']

In [ ]:

*# Токенизация по предложениям*

nltk\_tk\_sents = nltk.tokenize.sent\_tokenize(text1)

print(len(nltk\_tk\_sents))

nltk\_tk\_sents

1

Out[ ]:

['Предобработка данных в XML файле.']

# Частеречная разметка. Natasha

In [ ]:

from navec import Navec

from slovnet import Morph

In [ ]:

from google.colab import drive

drive.mount('/content/gdrive')

Mounted at /content/gdrive

In [ ]:

navec = Navec.load('/content/gdrive/My Drive/MMO/navec\_news\_v1\_1B\_250K\_300d\_100q.tar')

n\_morph = Morph.load('/content/gdrive/My Drive/MMO/slovnet\_morph\_news\_v1.tar', batch\_size=4)

In [ ]:

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

In [ ]:

def print\_pos(markup):

for token in markup.tokens:

print('{} - {}'.format(token.text, token.tag))

In [ ]:

n\_text1\_markup = list(\_ for \_ in n\_morph.map(nltk\_tk\_sents))

[print\_pos(x) for x in n\_text1\_markup]

П - PROPN|Animacy=Anim|Case=Nom|Gender=Masc|Number=Sing

р - NOUN

е - X|Foreign=Yes

д - NOUN

о - X|Foreign=Yes

б - NOUN|Animacy=Inan|Case=Loc|Gender=Masc|Number=Sing

р - X|Foreign=Yes

а - CCONJ

б - PROPN

о - NOUN|Animacy=Inan|Case=Gen|Gender=Fem|Number=Sing

т - PRON|Animacy=Inan|Case=Loc|Gender=Neut|Number=Sing

к - ADP

а - X|Foreign=Yes

- PUNCT

д - NOUN|Animacy=Inan|Case=Gen|Gender=Masc|Number=Sing

а - CCONJ

н - X|Foreign=Yes

н - X|Foreign=Yes

ы - X|Foreign=Yes

х - X|Foreign=Yes

- PUNCT

в - X|Foreign=Yes

- PUNCT

X - X|Foreign=Yes

M - PROPN|Foreign=Yes

L - X|Foreign=Yes

- PUNCT

ф - X|Foreign=Yes

а - CCONJ

й - ADJ|Case=Nom|Degree=Pos|Gender=Masc|Number=Sing

л - X|Foreign=Yes

е - NOUN|Animacy=Inan|Case=Gen|Gender=Masc|Number=Sing

. - PUNCT

# Лемматизация. Natasha

In [ ]:

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

In [ ]:

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 [ ]:

n\_doc1 = n\_lemmatize(text1)

{\_.text: \_.lemma for \_ in n\_doc1.tokens}

Out[ ]:

{'.': '.',

'XML': 'xml',

'Предобработка': 'предобработка',

'в': 'в',

'данных': 'данные',

'файле': 'файл'}

In [ ]:

n\_doc2 = n\_lemmatize(text2)

{\_.text: \_.lemma for \_ in n\_doc2.tokens}

Out[ ]:

{'.': '.', 'Бонд': 'бонд', 'Джеймс': 'джеймс', 'Меня': 'я', 'зовут': 'звать'}

# Выделение (распознавание) именованных сущностей. Natasha

In [ ]:

from slovnet import NER

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

In [ ]:

ner = NER.load('/content/gdrive/My Drive/MMO/slovnet\_ner\_news\_v1.tar')

In [ ]:

ner\_res = ner.navec(navec)

In [ ]:

markup\_ner2 = ner(text2)

In [ ]:

markup\_ner2

Out[ ]:

SpanMarkup(

text='Меня зовут Бонд. Джеймс Бонд',

spans=[Span(

start=11,

stop=15,

type='PER'

), Span(

start=17,

stop=28,

type='PER'

)]

)

In [ ]:

show\_markup(markup\_ner2.text, markup\_ner2.spans)

Меня зовут Бонд. Джеймс Бонд

PER─ PER────────

# Разбор предложения. Natasha

In [ ]:

from natasha import NewsSyntaxParser

In [ ]:

emb = NewsEmbedding()

syntax\_parser = NewsSyntaxParser(emb)

In [ ]:

n\_doc1.parse\_syntax(syntax\_parser)

n\_doc1.sents[0].syntax.print()

┌► Предобработка amod

└─ данных

┌► в case

└─ XML

файле

.

In [ ]:

n\_doc2.parse\_syntax(syntax\_parser)

n\_doc2.sents[0].syntax.print()

┌► Меня obj

┌─┌─└─ зовут

│ └──► Бонд xcomp

└────► . punct