осковский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа №5

по дисциплине «Методы машинного обучения»

на тему«Предобработка текста»

Выполнил: Ван Пэй

Группа：ИУ5-22М

Москва — 2021 г.

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.

```
with open('../input/hp-rus/HP1RUS.TXT', 'r', encoding='cp1251') as f:
    data = f.read()
```

```
data[:500]
```

```
'  \n  Джоан Кэтлин Роулинг \n  Гарри Поттер и философский камень \n  \n  Глава первая \n Мальчик, который выжил \n  \n  Мистер и миссис Дарсли, из дома номер четыре по Бирючинному проезду, могли бы \nс гордостью сказать, что они, слава богу, совершенно нормальные люди. Они были \nбы последними, от кого вы могли бы ожидать участия в чем-нибудь странном и \nтаинственном, потому что они совершенно не одобряли подобной чепухи. \n  Мистер Дарсли работал директором фирмы Граннингз, которая выпускала'
```

```
data = data.replace('\n', '')
```

```
data[:100]
```

```
'  Джоан Кэтлин Роулинг  Гарри Поттер и философский камень  Глава первая  Мальчик, кот'
```

- **Токенизация

```
from nltk.tokenize import sent_tokenize
sentences = sent_tokenize(data)
```

```
[['t'], 'ttt']
```

```
[['t'], 'ttt']
```

```
type(sentences)
```

```
list
```

```
sentences[:10]
```

```
['     Джоан Кэтлин Роулинг   Гарри Поттер и
философский камень     Глава первая   Маль
чик, который выжил     Мистер и миссис Дарс
ли, из дома номер четыре по Бирючинному про
езду, могли бы с гордостью сказать, что они,
слава богу, совершенно нормальные люди.',
'Они были бы последними, от кого вы могли б
ы ожидать участия в чем-нибудь странном и т
аинственном, потому что они совершенно не
одобряли подобной чепухи.',
'Мистер Дарсли работал директором фирмы Г
раннингз, которая выпускала сверла.',
'Он был крупный, крепкий мужчина с очень ко
роткой шеей и очень большими усами.',
'Миссис Дарсли была худой и светловолосой,
зато ее шеи с лихвой хватило бы на двоих.',
'Этим подарком природы она усердно пользо
валась, большую часть времени шпионя за со
седями через садовые изгороди.',
'У Дарсли рос сын Дадли  и они были уверены
```

```
from string import punctuation
punctuation
```

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

```
# sentences[2].lower()
```

```
table = str.maketrans("","", punctuation)
simple_preprocess = [s.lower().translate(table) for s in sentences]
```

```
sentences[10]
```

```
'Дарсли содрогались от мысли, что скажут
седи, если увидят Поттеров.'
```

```
simple_preprocess[10]
```

```
'д а р с л и  с о д р о г а л и с ь  о т  м ы с л и  ч т о  с к а ж у т  с
с е д и  е с л и  у в и д я т  п о т т е р о в'
```

- **Токенизация

```
from nltk.tokenize import word_tokenize
tokenized_words = [word_tokenize(s) for s in simple_preprocess]
```

```
tokenized_words[2:5]
```

```
 'Ч В О N Х',]]
 'Н Я',
 'Q Р',
 'Х В Я Т N Ч О',
 'Ч N Х В О Ц',
 'С',
 'П Б N',
 'Б Б',
 'З Я Т О',
 'С В Б Т Ч О В О Ч О С О Ц',
 'N',
 'Х Л Ч О Ц',
 'Q Р Ч Я',
 'Ч Я Ь С Ч N',
[' И N С С N С',
 'Л С Я И N',]
 'Q О Ч Р П N И N',
 'О Й Б Н Р',
 'N',
 'П Б Б Ц',
 'К О Ь О Т К О Ц',
 'О Й Б Н Р',
 'С',
 'И Л Ж Й N Н Я',
 'К Ь Б Ш К N Ц',
 'К Ь Л Ш Н Р Ц',
 'Q Р Ч',
[' О Н',
 'С В Б Ь Ч Я',]
 'В Р Ш Л С К Я Ч Я',
 'К О Т О Ь Я Я',
 'L Ь Я Н Н N Н L З',
 'Ф N Ь И Р',
 'Ч N Ь Б К Т О Ь О И',
 'Ь Я Q О Т Я Ч',
 'Ч Я Ь С Ч N',
[[' И N С Т Б Ь',
```

```
[word for sent in tokenized_words for word in sent][:5]
```

```
['д ж о а н', 'к э т л и н', 'р о у л и н г', 'г а р р и', 'п о т т е р']
```

```
','.join([word for sent in tokenized_words for word in sent][:5])
```

```
'д ж о а н,к э т л и н,р о у л и н г,г а р р и,п о т т е р'
```

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

text = ','.join([word for sent in tokenized_words for word in sent])

wordcloud = WordCloud(height=1000, width=2000, background_color='white', colorma
plt.figure(figsize=(20,20))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```


```
from nltk.corpus import stopwords

import nltk
nltk.download("stopwords")

russian_stopwords = stopwords.words("russian")
print(russian_stopwords)
```

```
[nltk_data] Downloading package stopwords to /usr/share/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
['и', 'в', 'во', 'не', 'что', 'он', 'на', 'я', 'с', 'со', 'как', 'а', 'то', 'все', 'она', 'так', 'его', 'но', 'да', 'ты', 'к', 'у', 'же', 'вы', 'за', 'бы', 'по', 'только', 'ее', 'мне', 'было', 'вот', 'от', 'меня', 'еще', 'нет', 'о', 'из', 'ему', 'теперь', 'когда', 'даже', 'ну', 'вдруг', 'ли', 'если', 'уже', 'или', 'ни', 'быть', 'был', 'него', 'до', 'вас', 'нибудь', 'опять', 'уж', 'вам', 'ведь', 'там', 'потом', 'себя', 'ничего', 'ей', 'может', 'они', 'тут', 'где', 'есть', 'надо', 'ней', 'для', 'мы', 'тебя', 'их', 'чем', 'была', 'сам', 'чтоб', 'без', 'будто', 'чего', 'раз', 'тоже', 'себе', 'под', 'будет', 'ж', 'тогда', 'кто', 'этот', 'того', 'потому', 'этого', 'какой', 'совсем', 'ним', 'здесь', 'этом', 'один', 'почти', 'мой', 'тем', 'чтобы', 'нее', 'сейчас', 'были', 'куда', 'зачем', 'всех', 'никогда', 'можно', 'при', 'наконец', 'два', 'об', 'другой', 'хоть', 'после', 'над', 'больше', 'тот', 'через', 'эти', 'нас', 'про', 'всего', 'них', 'какая', 'много', 'разве', 'три', 'эту', 'моя', 'впрочем', 'хорошо', 'свою', 'этой', 'перед', 'иногда', 'лучше', 'чуть', 'том', 'нельзя', 'такой', 'им', 'более', 'всегда', 'конечно', 'всю', 'между']
```

```
russian_stopwords.append('стоп-слово')
```

Отфильтруем все стопслова, которые найдем

```
removed_stopwords = [word for sent in tokenized_words for word in sent if word n
```

```
text = ','.join(removed_stopwords)

wordcloud = WordCloud(height=1000, width=2000, background_color='white', colorma
plt.figure( figsize=(20, 20))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

Стемминг

Лемматизация

```
!pip install pymorphy2
```

```
Collecting pymorphy2
  Downloading pymorphy2-0.9.1-py3-none-any.whl (55 kB)
     |████████████████████████████████| 55 kB
286 kB/s eta 0:00:011
Collecting dawg-python>=0.7.1
  Downloading DAWG_Python-0.7.2-py2.py3-none-any.whl (11 kB)
Requirement already satisfied: docopt>=0.6 in /opt/conda/lib/python3.7/site-pa
ckages (from pymorphy2) (0.6.2)
Collecting pymorphy2-dicts-ru<3.0,>=2.4
  Downloading pymorphy2_dicts_ru-2.4.417127.4579844-py2.py3-none-any.whl (8.2
MB)
     |████████████████████████████████| 8.2 MB
3.6 MB/s eta 0:00:01
Installing collected packages: dawg-python, pymorphy2-dicts-ru, pymorphy2
Successfully installed dawg-python-0.7.2 pymorphy2-0.9.1 pymorphy2-dicts-ru-2.
4.417127.4579844
WARNING: You are using pip version 20.1.1; however, version 20.2.4 is availabl
e.
You should consider upgrading via the '/opt/conda/bin/python3.7 -m pip install
--upgrade pip' command.
```

```
import pymorphy2
morph = pymorphy2.MorphAnalyzer()
```

```
removed_stopwords[2766]
```

```
'д о р о г о й'
```

```
morph.parse(removed_stopwords[2766])[0].normal_form
```

```
'д о р о г а'
```

```
lemmatized = [morph.parse(w)[0].normal_form for w in removed_stopwords]
```

```
text = ','.join(lemmatized)

wordcloud = WordCloud(height=1000, width=2000, background_color='white', colorma
plt.figure( figsize=(20, 20))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

Распознавание именованных сущностей

```
!pip install --user natasha navec
```

```
B 23.6 MB/s eta 0:00:01     |████████████████████████████████
█████     | 29.9 MB 10.0 MB/s eta 0:00:01
Collecting navec
  Downloading navec-0.10.0-py3-none-any.whl (23 kB)
Requirement already satisfied: pymorphy2 in /opt/conda/lib/python3.7/site-pack
ages (from natasha) (0.9.1)
Collecting slovnet>=0.3.0
  Downloading slovnet-0.5.0-py3-none-any.whl (49 kB)
     |████████████████████████████████████████| 49 kB
3.3 MB/s  eta 0:00:01
Collecting razdel>=0.5.0
  Downloading razdel-0.5.0-py3-none-any.whl (21 kB)
Collecting ipymarkup>=0.8.0
  Downloading ipymarkup-0.9.0-py3-none-any.whl (14 kB)
Collecting yargy>=0.14.0
  Downloading yargy-0.15.0-py3-none-any.whl (41 kB)
     |████████████████████████████████████████| 41 kB
67 kB/s s eta 0:00:01
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages
(from navec) (1.18.5)
Requirement already satisfied: pymorphy2-dicts-ru<3.0,>=2.4 in /opt/conda/lib/
python3.7/site-packages (from pymorphy2->natasha) (2.4.417127.4579844)
Requirement already satisfied: docopt>=0.6 in /opt/conda/lib/python3.7/site-pa
ckages (from pymorphy2->natasha) (0.6.2)
Requirement already satisfied: dawg-python>=0.7.1 in /opt/conda/lib/python3.7/
site-packages (from pymorphy2->natasha) (0.7.2)
Requirement already satisfied: intervaltree>=3 in /opt/conda/lib/python3.7/sit
e-packages (from ipymarkup>=0.8.0->natasha) (3.0.2)
Requirement already satisfied: sortedcontainers<3.0,>=2.0 in /opt/conda/lib/py
thon3.7/site-packages (from intervaltree>=3->ipymarkup>=0.8.0->natasha) (2.1.
0)
```

```
!wget https://storage.yandexcloud.net/natasha-navec/packs/navec_news_v1_1B_250K_
!wget https://storage.yandexcloud.net/natasha-slovnet/packs/slovnet_ner_news_v1.
```

```
--2020-11-11 09:51:41--  https://storage.yandexcloud.net/natasha-navec/packs/navec_news_v1_1B_250K_300d_100q.tar
Resolving storage.yandexcloud.net (storage.yandexcloud.net)... 213.180.193.243, 2a02:6b8::1d9
Connecting to storage.yandexcloud.net (storage.yandexcloud.net)|213.180.193.243|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26634240 (25M) [application/x-tar]
Saving to: ‘navec_news_v1_1B_250K_300d_100q.tar’

navec_news_v1_1B_25 100%[===================>]  25.40M  32.6MB/s    in 0.8s

2020-11-11 09:51:42 (32.6 MB/s) - ‘navec_news_v1_1B_250K_300d_100q.tar’ saved [26634240/26634240]

--2020-11-11 09:51:43--  https://storage.yandexcloud.net/natasha-slovnet/packs/slovnet_ner_news_v1.tar
Resolving storage.yandexcloud.net (storage.yandexcloud.net)... 213.180.193.243, 2a02:6b8::1d9
Connecting to storage.yandexcloud.net (storage.yandexcloud.net)|213.180.193.243|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2385920 (2.3M) [application/octet-stream]
Saving to: ‘slovnet_ner_news_v1.tar’

slovnet_ner_news_v1 100%[===================>]   2.28M  6.85MB/s    in 0.3s

2020-11-11 09:51:44 (6.85 MB/s) - ‘slovnet_ner_news_v1.tar’ saved [2385920/2385920]
```

```
from navec import Navec
from slovnet import NER

navec = Navec.load('./navec_news_v1_1B_250K_300d_100q.tar')
ner = NER.load('./slovnet_ner_news_v1.tar')
ner.navec(navec)

markup = ner(data)
```

```
persons = [data[s.start:s.stop] for s in markup.spans if s.type=='PER']
locations = [data[s.start:s.stop] for s in markup.spans if s.type=='LOC']
organizations = [data[s.start:s.stop] for s in markup.spans if s.type=='ORG']
```

```
# persons
```

```
f = {}
f['key1'] = 'ppppp'
f['key2'] = 5
f['key3'] = [4, 5, 7]
```

```
locations[:10]
```

```
['Бирючинному проезду',
 'Бирючинного проезда',
 'Кенте',
 'Йоркшире',
 'Данди',
 'Великобританией',
 'Бирючинный проезд',
 'Бирючинного проезда',
 'Бирючинном проезде',
 'Кенте']
```

```
def cnt_objects(objects_list: list) -> dict:
    cnt = {}

    for o in objects_list:
        if o not in cnt.keys():
            cnt[o] = 1
        else:
            cnt[o] +=1
    return cnt
```

```
sorted(cnt_objects(locations).items(), key=lambda x: x[1], reverse=True)[:5]
```

```
[('Хогвартсе', 23),
 ('Хогвартс', 11),
 ('Гринготтс', 11),
 ('Лондон', 7),
 ('Слайзерин', 7)]
```

```
sorted(cnt_objects(persons).items(), key=lambda x: x[1], reverse=True)[:5]
```

```
[('Гарри', 1207),
 ('Рон', 322),
 ('Хагрид', 238),
 ('Гермиона', 162),
 ('Дадли', 138)]
```

Список литературы

[1] Гапанюк Ю. Е. Лабораторная работа «Обработка признаков (часть2)» [Электронный ресурс] https://github.com/ugapanyuk/ml_course_2021/wiki/LAB_MMO__FEATURES