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

Лабораторная работа №6
по дисциплине
«Методы машинного обучения»

Выполнил:
студент группы ИУ5-22М
Лю Ченхао

Москва — 2024 г.

Задача токенизации

In 11
```
text='В 1885 году, молодой Константин Варнава, окончив студии в Санкт-Петербурге, оказался перед выбором — возвращаться в родные Пензенские просторы или отправиться в Москву. Семнадцатилетний Константин, буквально глотая знания, остался в столице и занялся репетиторством, чтобы избавить себя от нужды. За эти годы он увлеченно читал, написал статьи для студенческих газет, а газету «Пересмешник» с краткими зарисовками о жизни в Петербурге отправлял родным в Пензу. Тогда же Варнава впервые попробовал себя в драматургии, написав пьесу «Стыд и страсть» и водевиль «Вдова и зять».К 1889 году Варнава завершил свои студии и покинул Петербург, чтобы вернуться в Москву. Там он начал заботиться о своих близких, обеспечивая их скромный доход от литературных публикаций. Варнава дебютировал в печати в мае этого же года: в журнале «Жизнь и искусство» были опубликованы рассказ «Я и мои злоключения» и юмористическая заметка «Что видится во сне писателю».В том же году Варнава был принят на факультет искусств Московского университета им. М.В. Ломоносова. '
```
Executed at 2024.06.06 22:40:09 in less than 1ms

In 12
```
!pip install razdel
from razdel import tokenize, sentenize
n_tok_text = list(tokenize(text))
n_tok_text
```
Executed at 2024.06.06 22:40:13 in 2s 440ms

```
Requirement already satisfied: razdel in d:\python\lib\site-packages (0.5.0)
```

Out 12
```
[Substring(0, 1, 'В'),
 Substring(2, 6, '1885'),
 Substring(7, 11, 'году'),
 Substring(11, 12, ','),
 Substring(13, 20, 'молодой'),
 Substring(21, 31, 'Константин'),
 Substring(32, 39, 'Варнава'),
 Substring(39, 40, ','),
 Substring(41, 48, 'окончив'),
 Substring(49, 55, 'студии'),
 Substring(56, 57, 'в'),
```

```
In 13  1  [_.text for _ in n_tok_text]
          Executed at 2024.06.06 22:40:13 in 10ms
```

```
Out 13    'году',
          ',',
          'молодой',
          'Константин',
          'Варнава',
          ',',
          'окончив',
          'студии',
          'в',
          'Санкт-Петербурге',
          ',',
          'оказался',
```

```
In 14  1  n_sen_text = list(sentenize(text))
       2  n_sen_text
          Executed at 2024.06.06 22:40:13 in 21ms
```

```
Out 14  [Substring(0,
                   168,
                   'В 1885 году, молодой Константин Варнава, окончив студии в Санкт-Петербурге,
                   оказался перед выбором — возвращаться в родные Пензенские просторы или отправиться
                    в Москву.'),
         Substring(169,
                   296,
                   'Семнадцатилетний Константин, буквально глотая знания, остался в столице и занялся
                    репетиторством, чтобы избавить себя от нужды.'),
         Substring(297,
                   460,
```

In 15
```
[_.text for _ in n_sen_text], len([_.text for _ in n_sen_text])
```
Executed at 2024.06.06 22:40:14 in 17ms

Out 15
```
(['В 1885 году, молодой Константин Варнава, окончив студии в Санкт-Петербурге, оказался перед
 выбором — возвращаться в родные Пензенские просторы или отправиться в Москву.',
  'Семнадцатилетний Константин, буквально глотая знания, остался в столице и занялся
  репетиторством, чтобы избавить себя от нужды.',
  'За эти годы он увлеченно читал, написал статьи для студенческих газет, а газету
  «Пересмешник» с краткими зарисовками о жизни в Петербурге отправлял родным в Пензу.',
  'Тогда же Варнава впервые попробовал себя в драматургии, написав пьесу «Стыд и страсть» и
  водевиль «Вдова и зять».К 1889 году Варнава завершил свои студии и покинул Петербург, чтобы
   вернуться в Москву.',
  'Там он начал заботиться о своих близких, обеспечивая их скромный доход от литературных
```

In 16
```
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
```
Executed at 2024.06.06 22:40:16 in 24ms

In 17
```
n_sen_chunk = n_sentenize(text)
n_sen_chunk
```
Executed at 2024.06.06 22:40:17 in 114ms

Out 17
```
[['В',
  '1885',
  'году',
  ',',
  'молодой',
  'Константин',
  'Варнава',
  ',',
  'окончив',
  'студии',
  'в',
```

Частеречная разметка

In 13
```
!pip install navec
!pip install slovnet
from navec import Navec
from slovnet import Morph
```

```
Requirement already satisfied: navec in /usr/local/lib/python3.7/dist-packages (0.10.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from navec)
(1.19.5)
Collecting slovnet
  Downloading https://files.pythonhosted
  .org/packages/a9/3b/f1ef495be8990004959dd0510c95f688d1b07529f6a862bc56a405770b26/slovnet-0
  .5.0-py3-none-any.whl (49kB)
     |████████████████████████████████| 51kB 1.6MB/s
Requirement already satisfied: navec in /usr/local/lib/python3.7/dist-packages (from slovnet)
 (0.10.0)
Requirement already satisfied: razdel in /usr/local/lib/python3.7/dist-packages (from
```

In 24
```
morph_res = n_morph.navec(navec)
```

In 25
```
def print_pos(markup):
    for token in markup.tokens:
        print('{} - {}'.format(token.text, token.tag))
```

In 28
```
n_text_markup = list(_ for _ in n_morph.map(n_sen_chunk))
[print_pos(x) for x in n_text_markup]
```

```
7 - ADJ
июня - NOUN|Animacy=Inan|Case=Gen|Gender=Masc|Number=Sing
1889 - ADJ
года - NOUN|Animacy=Inan|Case=Gen|Gender=Masc|Number=Sing
. - PUNCT
Он - PRON|Case=Nom|Gender=Masc|Number=Sing|Person=3
поступил - VERB|Aspect=Perf|Gender=Masc|Mood=Ind|Number=Sing|Tense=Past|VerbForm=Fin|Voice
=Act
в - ADP
Киевский - ADJ|Animacy=Inan|Case=Acc|Degree=Pos|Gender=Masc|Number=Sing
политехнический - ADJ|Animacy=Inan|Case=Acc|Degree=Pos|Gender=Masc|Number=Sing
институт - NOUN|Animacy=Inan|Case=Acc|Gender=Masc|Number=Sing
```

Out 28
```
[None, None, None]
```

Лемматизация

Лемматизация

In 31
```
!pip install natasha
from natasha import Doc, Segmenter, NewsEmbedding, NewsMorphTagger, MorphVocab
```

```
Requirement already satisfied: razdel>=0.5.0 in /usr/local/lib/python3.7/dist-packages (from
natasha) (0.5.0)
Collecting yargy>=0.14.0
  Downloading https://files.pythonhosted
  .org/packages/d3/46/bc1a17200a55f4b0608f39ac64f1840fd4a52f9eeea462d9afecbf71246b/yargy-0.15
  .0-py3-none-any.whl (41kB)
     |████████████████████████████████| 51kB 5.5MB/s
Collecting pymorphy2
  Downloading https://files.pythonhosted
  .org/packages/07/57/b2ff2fae3376d4f3c697b9886b64a54b476e1a332c67eee9f88e7f1ae8c9/pymorphy2
  -0.9.1-py3-none-any.whl (55kB)
     |████████████████████████████████| 61kB 7.0MB/s
Collecting ipymarkup>=0.8.0
  Downloading https://files.pythonhosted
  .org/packages/bf/9b/bf54c98d50735a4a7c84c71e92c5361730c878ebfe903d2c2d196ef66055/ipymarkup
  -0.9.0-py3-none-any.whl
```

In 32
```
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 35
```
n_doc = n_lemmatize(text)
{_.text: _.lemma for _ in n_doc.tokens}
```

Out 35
```
{'.': '.',
 '1889': '1889',
 '1907': '1907',
 '1909-1912': '1909-1912',
 '7': '7',
 'В': 'в',
 'Киевский': 'киевский',
 'Он': 'он',
 'Сикорский': 'сикорский',
 'в': 'в',
 'вертолёта': 'вертолет',
```

Выделение (распознавание) именованных сущностей, named-entity recognition (NER)

```
In 39  1  ner_res = ner.navec(navec)

In 41  1  markup_ner = ner(text)
       2  markup_ner

Out 41    SpanMarkup(
              text='Сикорский родился 7 июня 1889 года. Он поступил в Киевский политехнический институт
               в 1907 году. В 1909-1912 годах студент Сикорский спроектировал и построил два
              вертолёта',
              spans=[Span(
                   start=50,
                   stop=83,
                   type='ORG'
               ), Span(
                   start=123,
                   stop=132,

In 42  1  show_markup(markup_ner.text, markup_ner.spans)

          Сикорский родился 7 июня 1889 года. Он поступил в Киевский
                                                             ORG————
          политехнический институт в 1907 году. В 1909-1912 годах студент
          ————————————————————————

          Сикорский спроектировал и построил два вертолёта
          PER————
```

Разбор предложения

```
In 43  1  from natasha import NewsSyntaxParser
```

```
In 44  1  emb = NewsEmbedding()
       2  syntax_parser = NewsSyntaxParser(emb)
```

```
In 45  1  n_doc.parse_syntax(syntax_parser)
       2  n_doc.sents[0].syntax.print()
```

```
          ┌─► Сикорский nsubj
       ┌──┌─└ родился
       │  │ ┌─► 7         obl
       │  │ │   └─ июня   flat
       │  │ ├─► 1889      amod
       │  └─└─► года      nmod
       └──────► .         punct
```

```
In 46  1  n_doc.sents[1].syntax.print()
```

```
                ┌─► Он                nsubj
          ┌─┌─┌─└ поступил
          │ │ │ ┌─► в                 case
          │ │ │ │ ┌─► Киевский        amod
          │ │ │ │ │ ┌─► политехнический amod
          │ │ └─└─└─└ институт        obl
          │ │       ┌─► в             case
          │ │       │ ┌─► 1907        amod
          │ └──────►└─└ году          obl
          └──────────► .              punct
```

```
In 47  1  n_doc.sents[2].syntax.print()
```

```
             ┌─► В             case
             │ ┌─► 1909-1912   amod
          ┌─└─└ годах         obl
          │  ┌─► студент       nsubj
          │  │ ┌─► Сикорский   nsubj
          └──└─└ спроектировал
          │    ┌─► и           cc
          ┌──└─└ построил      conj
          │    ┌─► два         nummod:gov
          └────└ вертолёта     obj
```

Список литературы

[1] Гапанюк Ю. Е. Лабораторная работа «Подготовка обучающей и тестовой выборки,кросс-валидация и подбор гиперпараметров на примере метода ближайших соседей»[Электронный ресурс] // GitHub.. — Режим доступа: https://github.com/ugapanyuk/ml_course/wiki/LAB_KNN

[2] Team The IPython Development. IPython 7.3.0 Documentation [Electronic resource] //Read the Docs. — Access mode: https://ipython.readthedocs.io/en/stable/

[3] Waskom M. seaborn 0.9.0 documentation [Electronic resource] // PyData.Access mode: https://seaborn.pydata.org/

[4] pandas 0.24.1 documentation [Electronic resource] // PyData. — Access mode:http://pandas.pydata.org/pandas-docs/stable/