МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа №__8__
по дисциплине «Методы машинного обучения в автоматизированных системах»

Тема: «Предобработка текста»

ИСПОЛНИТЕЛЬ: ___Калюта Н.И.___
ФИО
группа ИУ5-22М _____
подпись
"30"___05___2024 г.

ПРЕПОДАВАТЕЛЬ: ___ _____
ФИО
_____
подпись
"__"_____2024 г.

Москва - 2024

Цель лабораторной работы:

Изучение методов предобработки текстов.

Задание:

Для произвольного предложения или текста решить следующие задачи:

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

Установка библиотек Python, которые используются для обработки естественного языка (NLP)

```
[ ] !pip install nltk
    !pip install razdel
    !pip install navec
    !pip install slovnet
    !pip install natasha
    !pip install setuptools
    !pip install ipymarkup
```

Импорт библиотеки NLTK (Natural Language Toolkit) и инициализирует токенизатор "punkt" для английского языка

```
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
```

Импорт библиотек

```
[ ] from razdel import tokenize, sentenize
    from navec import Navec
    from slovnet import Morph
    from natasha import Doc, Segmenter, NewsEmbedding, NewsMorphTagger, MorphVocab
    from slovnet import NER
    from ipymarkup import show_span_ascii_markup as show_markup
    from natasha import NewsSyntaxParser
```

Исходные данные

```
[ ] text = """Самое большое количество английских вкраплений обнаруживается в критике и публицистике Пушкина и в его письмах, поменьше – в поэзии, еще меньше – в прозе. Что представляют собой эти вкрапления?
    Во-первых, это имена писателей: Byron, Walter Scott, Wordsworth, Southy, Shakespeare, Walpole, Coleridge и др., при этом английская форма имени сохраняется и в русском, и в английском текстах.
    Так же Пушкин пишет и имена литературных героев: Child-Harold, Manfred, Lalla-Rookh. Прямые цитаты служат в основном эпиграфами, в текстах их крайне мало"""
```

Токенизация

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

```
['Самое',
 'большое',
 'количество',
 'английских',
 'вкраплений',
 'обнаруживается',
 'в',
 'критике',
 'и',
 'публицистике',
 'Пушкина',
 'и',
 'в',
 'его',
 'письмах',
 ',',
 'поменьше',
 '–',
 'в',
 'поэзии',
 ',',
 'еще',
 'меньше',
 '–',
 'в',
 'прозе',
 '.',
```

```
'Что',
'представляют',
'собой',
'эти',
'вкрапления',
'?',
'Во',
'-',
'первых',
',',
'это',
'имена',
'писателей',
':',
'Byron',
',',
'Walter',
'Scott',
',',
'Wordsworth',
',',
'Southy',
',',
'Shakespeare',
',',
'Walpole',
',',
'Coleridge',
'и',
'др',
'.',',
'при',
'этом',
'английская',
'форма',
'имени',
'сохраняется',
'и',
'в',
'русском',
',',
'и',
'в',
'английском',
'текстах',
'.',
'Так',
'же',
'Пушкин',
'пишет',
'и',
'имена',
'литературных',
'героев',
':',
'Child',
'-',
'Harold',
',',
'Manfred',
',',
'Lalla',
'-',
'Rookh',
'.',
'Прямые',
'цитаты',
'служат',
'в',
'основном',
'эпиграфами',
',',
'в',
'текстах',
'их',
'крайне',
'мало']
```

Токенизация по предложениям

```
[ ] nltk_tk_sents = nltk.tokenize.sent_tokenize(text)
    print(len(nltk_tk_sents))
    nltk_tk_sents
```

```
5
['Самое большое количество английских вкраплений обнаруживается в критике и публицистике Пушкина и в его письмах, поменьше – в поэзии, еще меньше – в прозе.',
 'Что представляют собой эти вкрапления?',
 'Во-первых, это имена писателей: Byron, Walter Scott, Wordsworth, Southy, Shakespeare, Walpole, Coleridge и др., при этом английская форма имени сохраняется и в русском, и\nв английском текстах.',
 'Так же Пушкин пишет и имена литературных героев: Child-Harold, Manfred, Lalla-Rookh.',
 'Прямые цитаты служат в основном эпиграфами, в текстах их крайне мало']
```

Токенайзер **slovnet**

```
[ ] n_tok_text = list(tokenize(text))
    n_tok_text
```

```
[Substring(0, 5, 'Самое'),
 Substring(6, 13, 'большое'),
 Substring(14, 24, 'количество'),
 Substring(25, 35, 'английских'),
 Substring(36, 46, 'вкраплений'),
 Substring(47, 61, 'обнаруживается'),
 Substring(62, 63, 'в'),
 Substring(64, 71, 'критике'),
 Substring(72, 73, 'и'),
 Substring(74, 86, 'публицистике'),
 Substring(87, 94, 'Пушкина'),
 Substring(95, 96, 'и'),
 Substring(97, 98, 'в'),
 Substring(99, 102, 'его'),
 Substring(103, 110, 'письмах'),
 Substring(110, 111, ','),
 Substring(112, 120, 'поменьше'),
 Substring(121, 122, '—'),
 Substring(123, 124, 'в'),
 Substring(125, 131, 'поэзии'),
 Substring(131, 132, ','),
 Substring(133, 136, 'еще'),
 Substring(137, 143, 'меньше'),
 Substring(144, 145, '—'),
 Substring(146, 147, 'в'),
 Substring(148, 153, 'прозе'),
 Substring(153, 154, '.'),
 Substring(155, 158, 'Что'),
 Substring(159, 171, 'представляют'),
 Substring(172, 177, 'собой'),
 Substring(178, 181, 'эти'),
 Substring(182, 192, 'вкрапления'),
 Substring(192, 193, '?'),
 Substring(194, 203, 'Во-первых'),
 Substring(203, 204, ','),
 Substring(205, 208, 'это'),
 Substring(209, 214, 'имена'),
 Substring(215, 224, 'писателей'),
 Substring(224, 225, ':'),
 Substring(226, 231, 'Byron'),
 Substring(231, 232, ','),
 Substring(233, 239, 'Walter'),
 Substring(240, 245, 'Scott'),
 Substring(245, 246, ','),
 Substring(247, 257, 'Wordsworth'),
 Substring(257, 258, ','),
 Substring(259, 265, 'Southy'),
 Substring(265, 266, ','),
 Substring(267, 278, 'Shakespeare'),
 Substring(278, 279, ','),
 Substring(280, 287, 'Walpole'),
 Substring(287, 288, ','),
 Substring(289, 298, 'Coleridge'),
 Substring(299, 300, 'и'),
 Substring(301, 303, 'др'),
 Substring(303, 304, '.'),
 Substring(304, 305, ','),
 Substring(306, 309, 'при'),
 Substring(310, 314, 'этом'),
 Substring(315, 325, 'английская'),
 Substring(326, 331, 'форма'),
 Substring(332, 337, 'имени'),
 Substring(338, 349, 'сохраняется'),
 Substring(350, 351, 'и'),
 Substring(352, 353, 'в'),
 Substring(354, 361, 'русском'),
 Substring(361, 362, ','),
 Substring(363, 364, 'и'),
 Substring(365, 366, 'в'),
 Substring(367, 377, 'английском'),
 Substring(378, 385, 'текстах'),
 Substring(385, 386, '.'),
 Substring(387, 390, 'Так'),
 Substring(391, 393, 'же'),
 Substring(394, 400, 'Пушкин'),
 Substring(401, 406, 'пишет'),
 Substring(407, 408, 'и'),
 Substring(409, 414, 'имена'),
 Substring(415, 427, 'литературных'),
 Substring(428, 434, 'героев'),
 Substring(434, 435, ':'),
 Substring(436, 448, 'Child-Harold'),
 Substring(448, 449, ','),
 Substring(450, 457, 'Manfred'),
 Substring(457, 458, ','),
 Substring(459, 470, 'Lalla-Rookh'),
 Substring(470, 471, '.'),
 Substring(472, 478, 'Прямые'),
 Substring(479, 485, 'цитаты'),
 Substring(486, 492, 'служат'),
 Substring(493, 494, 'в'),
 Substring(495, 503, 'основном'),
 Substring(504, 514, 'эпиграфами'),
 Substring(514, 515, ','),
 Substring(516, 517, 'в'),
 Substring(518, 525, 'текстах'),
 Substring(526, 528, 'их'),
 Substring(529, 535, 'крайне'),
 Substring(536, 540, 'мало')]
```

```
[ ] list(sentenize(text))
```

```
[Substring(0,
           154,
           'Самое большое количество английских вкраплений обнаруживается в критике и публицистике Пушкина и в его письмах, поменьше – в поэзии, еще меньше – в прозе.'),
 Substring(155, 193, 'Что представляют собой эти вкрапления?'),
 Substring(194,
           386,
           'Во-первых, это имена писателей: Byron, Walter Scott, Wordsworth, Southy, Shakespeare, Walpole, Coleridge и др., при этом английская форма имени сохраняется и в русском, и\nв английском текстах.'),
 Substring(387,
           471,
           'Так же Пушкин пишет и имена литературных героев: Child-Harold, Manfred, Lalla-Rookh.'),
 Substring(472,
           540,
           'Прямые цитаты служат в основном эпиграфами, в текстах их крайне мало')]
```

Создадим набор токенов для каждого предложения отдельно для дальнейшей работы:

```
[ ] n_sen_chunk = []
    for sent in sentenize(text):
        tokens = [_.text for _ in tokenize(sent.text)]
        n_sen_chunk.append(tokens)
```

```
n_sen_chunk
```

```
[['Самое',
  'большое',
  'количество',
  'английских',
  'вкраплений',
  'обнаруживается',
  'в',
  'критике',
  'и',
  'публицистике',
  'Пушкина',
  'и',
  'в',
  'его',
  'письмах',
  ',',
  'поменьше',
  '–',
  'в',
  'поэзии',
  ',',
  'еще',
  'меньше',
  '–',
  'в',
  'прозе',
  '.'],
 ['Что', 'представляют', 'собой', 'эти', 'вкрапления', '?'],
 ['Во-первых',
  ',',
  'это',
  'имена',
  'писателей',
  ':',
  '.',
  'Byron',
  ',',
  'Walter',
  'Scott',
  ',',
  'Wordsworth',
  ',',
  'Southy',
  ',',
  'Shakespeare',
  ',',
  'Walpole',
  ',',
  'Coleridge',
  'и',
  'др',
  '.',
  '.',
  ',',
  'при',
  'этом',
  'английская',
  'форма',
  'имени',
  'сохраняется',
  'и',
  'в',
  'русском',
  ',',
  'и',
  'в',
  'английском',
  'текстах',
  '.'],
 ['Так',
  'же',
  'Пушкин',
  'пишет',
  'и',
  'имена',
  'литературных',
  'героев',
  ':',
  'Child-Harold',
  ',',
  'Manfred',
  ',',
  'Lalla-Rookh',
  '.'],
 ['Прямые',
  'цитаты',
  'служат',
  'в',
  'основном',
  'эпиграфами',
  ',',
  'в',
  'текстах',
  'их',
  'крайне',
  'мало']]
```

Частеричная разметка

```
[ ] !wget https://storage.yandexcloud.net/natasha-navec/packs/navec_news_v1_1B_250K_300d_100q.tar
    !wget https://storage.yandexcloud.net/natasha-slovnet/packs/slovnet_morph_news_v1.tar
```

```
[ ] navec = Navec.load('navec_news_v1_1B_250K_300d_100q.tar')
    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]
```

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

```
Самое - ADJ|Animacy=Inan|Case=Acc|Degree=Pos|Gender=Neut|Number=Sing
большое - ADJ|Case=Nom|Degree=Pos|Gender=Neut|Number=Sing
количество - NOUN|Animacy=Inan|Case=Nom|Gender=Neut|Number=Sing
английских - ADJ|Case=Gen|Degree=Pos|Number=Plur
вкраплений - NOUN|Animacy=Inan|Case=Gen|Gender=Masc|Number=Plur
обнаруживается - VERB|Aspect=Imp|Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|Voice=Mid
в - ADP
критике - NOUN|Animacy=Inan|Case=Loc|Gender=Fem|Number=Sing
и - CCONJ
публицистике - NOUN|Animacy=Inan|Case=Loc|Gender=Fem|Number=Sing
Пушкина - PROPN|Animacy=Anim|Case=Gen|Gender=Masc|Number=Sing
и - CCONJ
в - ADP
его - DET
письмах - NOUN|Animacy=Inan|Case=Loc|Gender=Neut|Number=Plur
, - PUNCT
поменьше - ADV|Degree=Cmp
— - PUNCT
в - ADP
поэзии - NOUN|Animacy=Inan|Case=Loc|Gender=Fem|Number=Sing
, - PUNCT
еще - ADV|Degree=Pos
меньше - NUM|Degree=Cmp
— - PUNCT
в - ADP
прозе - NOUN|Animacy=Inan|Case=Loc|Gender=Fem|Number=Sing
. - PUNCT
Что - PRON|Animacy=Inan|Case=Acc|Gender=Neut|Number=Sing
представляют - VERB|Aspect=Imp|Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act
собой - PRON|Case=Ins
эти - DET|Animacy=Inan|Case=Acc|Number=Plur
вкрапления - NOUN|Animacy=Inan|Case=Acc|Gender=Masc|Number=Plur
? - PUNCT
Во-первых - ADV|Degree=Pos
, - PUNCT
это - PRON|Animacy=Inan|Case=Nom|Gender=Neut|Number=Sing
имена - NOUN|Animacy=Inan|Case=Nom|Gender=Neut|Number=Plur
писателей - NOUN|Animacy=Anim|Case=Gen|Gender=Masc|Number=Plur
: - PUNCT
Byron - X|Foreign=Yes
, - PUNCT
Walter - X|Foreign=Yes
Scott - X|Foreign=Yes
, - PUNCT
Wordsworth - PROPN|Foreign=Yes
, - PUNCT
Southy - PROPN|Foreign=Yes
, - PUNCT
Shakespeare - PROPN|Foreign=Yes
, - PUNCT
Walpole - PROPN|Foreign=Yes
, - PUNCT
Coleridge - PROPN|Foreign=Yes
и - CCONJ
др - NOUN|Animacy=Inan|Case=Nom|Gender=Neut|Number=Plur
. - PUNCT
, - PUNCT
при - ADP
этом - PRON|Animacy=Inan|Case=Loc|Gender=Neut|Number=Sing
английская - ADJ|Case=Nom|Degree=Pos|Gender=Fem|Number=Sing
форма - NOUN|Animacy=Inan|Case=Nom|Gender=Fem|Number=Sing
имени - NOUN|Animacy=Inan|Case=Gen|Gender=Neut|Number=Sing
сохраняется - VERB|Aspect=Imp|Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|Voice=Mid
и - PART
в - ADP
русском - ADJ|Case=Loc|Degree=Pos|Gender=Masc|Number=Sing
, - PUNCT
и - CCONJ
в - ADP
английском - ADJ|Case=Loc|Degree=Pos|Gender=Masc|Number=Sing
текстах - NOUN|Animacy=Inan|Case=Loc|Gender=Masc|Number=Plur
```

```
. - PUNCT
Так - ADV|Degree=Pos
же - PART
Пушкин - PROPN|Animacy=Anim|Case=Nom|Gender=Masc|Number=Sing
пишет - VERB|Aspect=Imp|Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act
и - PART
имена - NOUN|Animacy=Inan|Case=Nom|Gender=Neut|Number=Plur
литературных - ADJ|Case=Gen|Degree=Pos|Number=Plur
героев - NOUN|Animacy=Anim|Case=Gen|Gender=Masc|Number=Plur
: - PUNCT
Child-Harold - X|Foreign=Yes
, - PUNCT
Manfred - PROPN|Foreign=Yes
, - PUNCT
Lalla-Rookh - X|Foreign=Yes
. - PUNCT
Прямые - ADJ|Case=Nom|Degree=Pos|Number=Plur
цитаты - NOUN|Animacy=Inan|Case=Nom|Gender=Fem|Number=Plur
служат - VERB|Aspect=Imp|Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act
в - ADP
основном - ADJ|Case=Loc|Degree=Pos|Gender=Neut|Number=Sing
эпиграфами - NOUN|Animacy=Inan|Case=Nom|Gender=Masc|Number=Sing
, - PUNCT
в - ADP
текстах - NOUN|Animacy=Inan|Case=Loc|Gender=Masc|Number=Plur
их - PRON|Case=Gen|Number=Plur|Person=3
крайне - ADV|Degree=Pos
мало - ADV|Degree=Pos
[None, None, None, None, None]
```

Лемматизация

```
[ ] 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}
```

```
{'Самое': 'самый',
 'большое': 'большой',
 'количество': 'количество',
 'английских': 'английский',
 'вкраплений': 'вкрапление',
 'обнаруживается': 'обнаруживаться',
 'в': 'в',
 'критике': 'критика',
 'и': 'и',
 'публицистике': 'публицистика',
 'Пушкина': 'пушкин',
 'его': 'его',
 'письмах': 'письмо',
 ',': ',',
 'поменьше': 'маленький',
 '—': '—',
 'поэзии': 'поэзия',
 'еще': 'еще',
 'меньше': 'меньше',
 'прозе': 'проза',
 '.': '.',
 'Что': 'что',
 'представляют': 'представлять',
 'собой': 'себя',
 'эти': 'этот',
 'вкрапления': 'вкрапление',
 '?': '?',
 'Во-первых': 'во-первых',
 'это': 'это',
 'имена': 'имя',
 'писателей': 'писатель',
 ':': ':',
 'Byron': 'byron',
 'Walter': 'walter',
 'Scott': 'scott',
 'Wordsworth': 'wordsworth',
 'Southy': 'southy',
 'Shakespeare': 'shakespeare',
 'Walpole': 'walpole',
 'Coleridge': 'coleridge',
 'др': 'др',
 'при': 'при',
 'этом': 'это',
```

```
    'английская': 'английский',
    'форма': 'форма',
    'имени': 'имя',
    'сохраняется': 'сохраняться',
    'русском': 'русский',
    'английском': 'английский',
    'текстах': 'текст',
    'Так': 'так',
    'же': 'же',
    'Пушкин': 'пушкин',
    'пишет': 'писать',
    'литературных': 'литературный',
    'героев': 'герой',
    'Child-Harold': 'child-harold',
    'Manfred': 'manfred',
    'Lalla-Rookh': 'lalla-rookh',
    'Прямые': 'прямой',
    'цитаты': 'цитата',
    'служат': 'служить',
    'основном': 'основный',
    'эпиграфами': 'эпиграф',
    'их': 'они',
    'крайне': 'крайне',
    'мало': 'мало'}
```

Выделение (распознавание) именованных сущностей

Загрузка файла с обученной моделью для распознавания именованных сущностей (NER) из хранилища Яндекс Cloud

```
[ ] !wget https://storage.yandexcloud.net/natasha-slovnet/packs/slovnet_ner_news_v1.tar
```

Распознавание именованных сущностей (NER) в тексте text с использованием загруженной модели.

```
[ ] ner = NER.load('slovnet_ner_news_v1.tar')
    ner_res = ner.navec(navec)
    markup_ner = ner(text)
```

Выделение именованных сущностей

```
markup_ner
```

```
SpanMarkup(
    text='Самое большое количество английских вкраплений обнаруживается в критике и публицистике Пушкина и в его письмах, поменьше — в поэзии, еще меньше — в прозе. Что представляет собой эти вкрапления? Во-первых, это имена писателей: Byron, Walter Scott, Wordsworth, Southy, Shakespeare, Walpole, Coleridge и др., при этом английская форма имени сохраняется и в русском, и\nв английском текстах. Так же Пушкин пишет и имена литературных героев: Child-Harold, Manfred, Lalla-Rookh. Прямые цитаты служат в основном эпиграфами, в текстах их крайне мало',
    spans=[Span(
        start=87,
        stop=94,
        type='PER'
    ),
    Span(
        start=226,
        stop=231,
        type='ORG'
    ),
    Span(
        start=233,
        stop=245,
        type='ORG'
    ),
    Span(
        start=247,
        stop=257,
        type='ORG'
    ),
    Span(
        start=259,
        stop=265,
        type='ORG'
    ),
    Span(
        start=267,
        stop=278,
        type='ORG'
    ),
    Span(
        start=280,
        stop=287,
        type='ORG'
    ),
    Span(
        start=289,
        stop=298,
        type='ORG'
    ),
    Span(
        start=394,
        stop=400,
        type='PER'
    ),
    Span(
        start=436,
        stop=448,
        type='ORG'
    ),
    Span(
        start=450,
        stop=457,
        type='ORG'
    ),
    Span(
        start=459,
        stop=470,
        type='ORG'
    )]
)
```

Визуализация результаты работы алгоритма распознавания именованных сущностей

```
show_markup(markup_ner.text, markup_ner.spans)
```

```
Самое большое количество английских вкраплений обнаруживается в
критике и публицистике Пушкина и в его письмах, поменьше – в поэзии,
                      PER————
еще меньше – в прозе. Что представляют собой эти вкрапления? Во-
первых, это имена писателей: Byron, Walter Scott, Wordsworth, Southy,
                              ORG——  ORG—————  ORG————  ORG——
Shakespeare, Walpole, Coleridge и др., при этом английская форма имени
ORG————  ORG——  ORG————
 сохраняется и в русском, и
в английском текстах. Так же Пушкин пишет и имена литературных героев:
                              PER——
 Child-Harold, Manfred, Lalla-Rookh. Прямые цитаты служат в основном
 ORG————  ORG——  ORG————
эпиграфами, в текстах их крайне мало
```

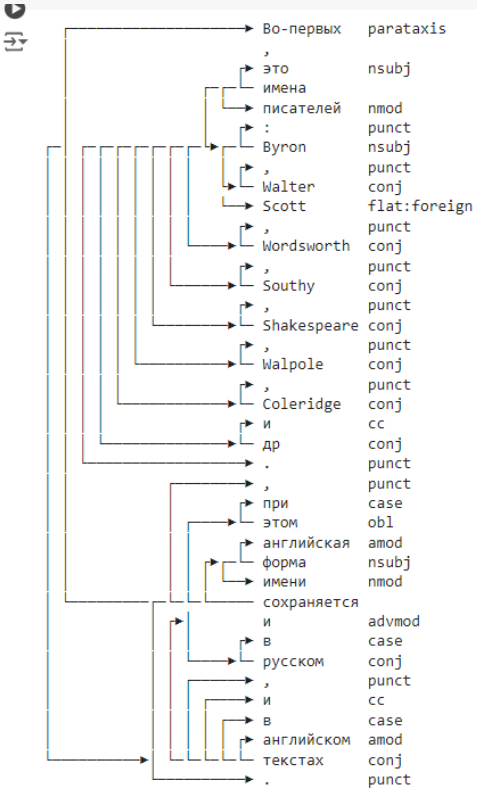
Разбор предложения

Создаём объект NewsEmbedding, который представляет собой модель для получения векторных представлений слов (эмбеддингов), а затем используем эту модель для инициализации синтаксического анализатора NewsSyntaxParser

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

Синтаксический анализ текста, представленного в объекте n_doc, и затем вывод синтаксического деревоъа для третьего предложения этого текста

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



Вывод:

В ходе выполнения лабораторной работы познакомился с основами обработки естественного языка (NLP) на Python с использованием библиотек NLTK, Razdel, Navec, Slovnet и Natasha.

Я научился выполнять такие операции, как:

- Токенизация и сегментация текста: Разбиение текста на предложения и слова.
- Морфологический анализ: Определение части речи, рода, числа и других

грамматических характеристик слов.

- Лемматизация: Приведение слов к их основной форме.
- Разметка именованных сущностей (NER): Выделение именованных сущностей в тексте, таких как имена людей, организаций и мест.
- Синтаксический анализ: Анализ грамматической структуры предложений.

Я также узнал о различных моделях и ресурсах для NLP, таких как Navec и Slovnet, и о возможностях их использования в различных задачах.

Эти знания помогут мне в будущем реализовывать более сложные NLP-проекты.