

Московский Авиационный Институт  
(Научный Исследовательский Институт)

Факультет прикладной математики и физики  
Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу  
«Обработка естественно-языковых текстов»

Студент: Мхитарян С.А.

Преподаватель: Кухтичев А.А.

Группа: М8О-206М

Дата:

Оценка:

Подпись:

Москва, 2021

## ЛР1: Токенизация

### Задание

Реализовать процесс разбиения текстов документов на токены, который потом будет использоваться при индексации. Для этого потребуется выработать правила, по которым текст делится на токены. Необходимо описать их в отчёте, указать достоинства и недостатки

выбранного метода. Привести примеры токенов, которые были выделены неудачно, объяснить, как можно было бы поправить правила, чтобы исправить найденные проблемы.

### Метод решения

1. Изучение способов токенизации текста с использованием разных библиотек python.
2. Установка необходимых библиотек на ПК.
3. Выделение текста статей из скачанного дампа при помощи nltk и собственных наработок.
4. Написание и отладка кода, выполняющего разделение текста на отдельные токены.
5. Сбор и анализ статистических данных.

### Журнал выполнения

№	Действие	Проблема	Решение
1	Установка nltk библиотеки python на ПК	-	
2	Выполнение программы на данных, которые были получены лабораторной работы 2 по информационному поиску	Время - можно оптимизировать с помощью потоков и более быстрого языка программирования	

## Результаты выполнения

Данные	Размер до токенизации	Размер после токенизации	Средняя длина токена	Время
Все статьи в формате json	18 ГБ	6.2ГБ	6.45	1:21:12.494894

## Исходные данные

[illegible]

## Полученный результат

```
1 4 {"id": "4", "revid": "450", "url": "https://ru.wikipedia.org/wiki?curid=4", "title": "Базовая статья", "text": ""}
2 {"id": "7", "revid": "117955298", "url": "https://ru.wikipedia.org/wiki?curid=7", "title": "Литва", "text": "Литва́ официальное название литовская респуб́лика государство расположенное северной части европы площадь км² протяжённость севера юг 280 км запада восток 370 км население составляет человек январь 2021 занимает 137е место мире численности населения 121е территории имеет выход балтийскому морю расположена восточном побережье береговая линия составляет 99 км наименьший показатель среди государств балтии севере граничит латвией югостоке белоруссией югозападе польшей калининградской областью россии площади населению является самым крупным прибалтийским государствомстолица вильнюс официальный язык литовский денежная единица евровосстановление независимости страны провозглашено 11 марта 1990 года 6 сентября 1991 года государственный совет ссср признал независимость литвы литва член оон 1991 обсе 1991 совета европы 1993 вто 2001 европейского союза 2004 нато 2004 озср 2018 входит шенгенскую зону еврозонуэтимологияэтимология слова литва точно известна существует множество версий одна которых получила всеобщего признания корень лит варианты летлют допускают различные толкования балтских славянских других индоевропейских языках например существуют созвучные топонимы территории словакии lytva румынии litua известные xixii веков мнению е поспелова топоним образован древнего названия реки летава Lietava лить русское летаука феодальное княжество землям которого протекала эта река временем заняло ведущее положение название распространено всё государство повести временных лет xii век упоминается этноним литва полностью совпадающий названием местности литва смыслу территория живёт литва формагеографияповерхность равнинная следами древнего оледенения поля луга занимают 57 территории леса кустарники 30 болота 6 внутренние воды 1 высшая точка 29384 м уровнем моря холм аукштояс аукштасис калнас юговосточной части страны 235 км вильнюсакрупнейшие реки неман вилияболее 3 тыс озёр 15 территории крупнейшее друکشяй границе латвии литвы белоруссии площадь 448 км² самое глубокое таурагнас 61 м самое длинное асвея длиной 30 км местечка дубингайклимат переходный морского континентальному средняя температура зимой -5 °с летом 17 °с выпадает 748 мм осадков годполезные ископаемые торф минеральные материалы строительные материалыисториядревнейшая историятерритория современной литвы заселена людьми конца xix тысячелетия н э жители занимались охотой рыболовством использовали лук стрелы кремнёвыми наконечниками скребки обработки кожи удочки сети конце неолита iiiiii тысячелетия н э территорию современной литвы проникли индоевропейские племена занимались земледелием скотоводством охота рыболовство оставались основными занятиями местных жителей вплоть широкого распространения железных орудий труда индоевропейцы заселившие земли устьями вислы западной двины выделились отдельную группу названную учёными балтамитрадиционно считается этническая
```

### Положительные стороны:

- можно быстро и удобно реализовать на языке Python
- библиотека с русскими словами уже готова, не нужно писать «велосипеды»

### Отрицательные стороны:

- использования языка python (можно выбрать более оптимальный язык для работы с текстами)
- не все «лишние знаки» были подчищены

## Исходный код

```
1 import ssl
2 import json
3 import logging
4 import os.path
5 import threading
6 from os import listdir
7 from pathlib import Path
8 from datetime import datetime
9
10 import nltk
11 from nltk.corpus import stopwords
12 from nltk.tokenize import word_tokenize
13
14
15 # Avoid an error with the ssl certificate
16 try:
17     _create_unverified_https_context = ssl._create_unverified_context
18 except AttributeError:
19     pass
20 else:
21     ssl._create_default_https_context = _create_unverified_https_context
22
23 nltk.download('stopwords')
24 nltk.download('punkt')
25 russian_stopwords = stopwords.words('russian')
```

```

27 name = os.path.splitext(os.path.basename(__file__))[0]
28 home_path = '/Users/samvel/projects/MAI-IR'
29 wikipedia_data_path = os.path.join(home_path, 'IR', 'laboratory_work_1', 'wikipedia')
30 transformed_data_path = os.path.join(home_path, 'NLPT', 'laboratory_work_1', 'transformed_data')
31
32 bad_chars = {
33     '\n', '\t', '\\', '/', '"', '\'', '.', ',',
34     '!', '$', '#', '@', '%', '&', '-', '=', '+',
35     '?', '~', '|', '^', '_', '±', '§', '¶', '»', '«', '€',
36     '°c', '„', '“', '”', '…', 'SHY'
37 }
38
39
40 def transform_data(file_path, logger):
41     logger.info(f'threadId={threading.current_thread().ident}, transform_file={file_path}')
42     with open(file_path) as file:
43         lines = file.readlines()
44
45         data = ''
46         for line in lines:
47             json_data = json.loads(line)
48             transformed_data = json_data['text'].lower()
49
50             for bad_char in bad_chars:
51                 transformed_data = transformed_data.replace(bad_char, '')
52
53             tokens = word_tokenize(transformed_data, language="russian")
54             json_data['text'] = " ".join(token for token in tokens if token not in russian_stopwords)
55             data += f'{json.dumps(json_data, ensure_ascii=False)}\n'
56
57         folder_name = os.path.basename(os.path.dirname(file_path))
58         file_name = os.path.splitext(os.path.basename(file_path))[0]
59
60         article_path = os.path.join(transformed_data_path, folder_name, file_name)
61         with open(article_path, 'w', encoding='utf8') as new_file:
62             new_file.write(data)
63
64
65 if __name__ == '__main__':
66     fmt = '%Y-%m-%d'
67     logger = logging.getLogger(name)
68     ch = logging.StreamHandler()
69     formatter = logging.Formatter(fmt='%(asctime)s %(levelname)s %(name)s: %(message)s', datefmt='%y/%m/%d %H:%M:%S')
70     ch.setFormatter(formatter)
71     logger.addHandler(ch)
72     logger.setLevel(logging.DEBUG)
73
74     articles_paths = []
75     for folder in sorted(listdir(wikipedia_data_path)):
76         if not folder.startswith('.'):
77             Path(os.path.join(transformed_data_path, folder)).mkdir(parents=True, exist_ok=True)
78             articles_paths.append(os.path.join(wikipedia_data_path, folder))
79
80     start = datetime.now()
81
82     for articles_path in articles_paths:
83         articles_path_folders = sorted(listdir(articles_path))
84         articles = [os.path.join(articles_path, folder) for folder in articles_path_folders]
85
86         threads = []
87         for number, article in enumerate(articles):
88             thread = threading.Thread(target=transform_data, name=f'thread_{number}', args=(article, logger))
89             thread.start()
90             threads.append(thread)
91
92         for thread in threads:
93             thread.join()
94
95     end = datetime.now()
96     logger.info(end - start)

```

## Выводы

При выполнении лабораторной работы по обработке естественно-языковых текстов были токенизированы тексты из статей русской Википедии. Хочется отметить, что в этом помогла такая полезная python-библиотека, как nltk, в которой есть русские стоп слова. Также использование библиотек threading на Python помогло увеличить скорость токенизации примерно на 10%.