# 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. Написание и отладка кода, выполняющего разделение текста на отдельные токены.
- 3. Сбор и анализ статистических данных.

В качестве инструмента по токенизации текста была выбрана библиотека nltk, которая обладает словарем стоп-слов русского языка. Дополнительно текст был очищен он мусорных знаков: знаков пунктуцации, знаков операций и других аscii символов, которые не представляют буквы русского алфавита.

## Результаты выполнения

Файлы	Размер до токенизации	Размер после токенизации	Средняя длина токена	Время
все статьи в формате json	11Gb	9.9Gb	6.33	1:29:40

#### Исходные данные



## Результат



Как видно из результата токенизации, не все токены были выделены удачно, к примеру '°с', 'iii', '→', знаки ударения все еще присутствуют. Добиться лучших результатов можно более тщательной обработкой текста.

Из достоинств выбранного метода токенизации можно выделить:

• использование готового словаря стоп-слов русского языка

Из недостатков:

• невысокая скорость обработки

## Исходный код

```
import os
import json
from pathlib import Path
from trash import trash
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
nltk.download("stopwords")
russian_stopwords = stopwords.words("russian")
def replacer(text, dic):
    for i, j in dic.items():
       text = text.replace(i, j)
    return text
def preprocess_text(text):
    text = text.lower()
    text = replacer(text, trash)
    tokens = word_tokenize(text, language="russian")
    tokens = [token for token in tokens if token not in russian_stopwords]
    text = " ".join(tokens)
    text = " ".join(text.split())
    return text
if __name__ == '__main__':
    raw_dir_path = "/Users/denis/MAI/IR/lab3/wikipedia_articles_one_by_one"
    raw_dir_list = [(raw_dir_path + "/" + i) for i in sorted(
        os.listdir(raw_dir_path),
        key=lambda x: int(x.replace(".json", "").replace("wiki_article_", ""))
    new_path = "/Users/denis/MAI/IR/lab3/wikipedia_articles_tokenized"
    for filepath in raw_dir_list:
        with open(filepath) as file:
            data = json.load(file)
            data["text"] = preprocess_text(data["text"])
            new_name = f"wiki_article_{data['id']}.json"
            with open(new_path + "/" + new_name, "w", encoding='utf8') as new_file:
                json.dump(data, new_file, ensure_ascii=False)
```

#### Выводы

В процессе выполнения данной лабораторной работы была выполнена токенизация текстов статей википедии с использованием библиотеки nltk на Python. Было интересно поискать готовые средства для предобработки текста и создания токенов. Наличие встроенного и постоянно обновляемого словаря стопслов в библиотеке nltk было приятным открытием.