Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу «Обработка текстов на естественном языке»

Студент: А. Т. Бахарев
Преподаватель: А. А. Кухтичев
Группа: М8О-108М
Дата:
Оценка:
Подпись:

Москва, 2021

Лабораторная работа №1 «Токенизация»

Нужно 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. В результатах выполнения работы нужно указать следующие статистические данные:

- Количество токенов.
- Среднюю длину токена

Кроме того, нужно привести время выполнения программы, указать зависимость времени от объёма входных данных. Указать скорость токенизации в расчёте на килобайт входного текста. Является ли эта скорость оптимальной? Как её можно ускорить?

1 Описание

Требуется разработать программу токенизатор, которая будет разбивать текст на токены для дальнейшей обработки. В качестве разделителя довольно эффективно использовать знаки препинания, однако некоторые токены таким образом будут утрачены (например сокращения).

Так же я добавил и другие разделяющие символы, которые были выявлены в процессе анализа исходного текста. Благодаря им, качество токенизации существенно улучшилось. Эти и другие символы можно увидеть в исходном коде.

После получения токена из текста необходимо проверить, что данный токен не является числом и оно не содержится в стоп-словах. После всех проверок токен будет добавлен в файл.

13202 документа токенизируются за 12 минут. Время работы программы можно сократить, если уменьшить количество чтений с диска. Так же, нужно уменьшить количество записей на диск.

2 Исходный код

```
1
2  #include <iostream>
3  #include <regex>
4  #include <boost/filesystem.hpp>
5  #include <boost/range/iterator_range.hpp>
6  #include <fstream>
7  #include <algorithm>
8
9  #include <utils.hpp>
10 #include <common.h>
11 #include "stop_words.h"
12 #include <vector>
13
14 std::map<std::string, uint64_t> frequency;
15
16 namespace fs = boost::filesystem;
17
18 bool is_token_correct(const std::string& low_str)
19 {
20   if (stop_words.find(low_str) != stop_words.end())  stop words
21     return false;
22
23   if (utils::is_digit(low_str))
24     return false;
25
26   return true;
27 }
28
29 void tokenize_text(const fs::path& input, const fs::path& output)
30 {
31   std::fstream input_file(input.string(), std::ios::in);
32   std::fstream output_file(output.string(), std::ios::out);
33
34   std::string curr_str;
35   const std::regex re(R"([\s|,|.|;|:|(|)|<|>|=|+|\/|\[|\]|\'|\%|\-|\*|\?|\!|\@|\"]+)")
         ;
36
37   while (std::getline(input_file, curr_str))
38   {
39     std::sregex_token_iterator it{curr_str.begin(), curr_str.end(), re, -1 };
40     std::vector<std::string> tokenized{ it, {}};
41      Additional check to remove empty strings and one-char strings
42     tokenized.erase(std::remove_if(tokenized.begin(), tokenized.end(),[](std::string
          const& s) {
43       return s.size() == 0 || s.size() == 1;}),
44     tokenized.end());
45
```

```
46      for (auto& token : tokenized)
47      {
48        std::transform(token.begin(), token.end(), token.begin(), [](unsigned char c) {
              return std::tolower(c); });
49
50        if (is_token_correct(token))
51        {
52          output_file << token << "\n";
53        }
54      }
55    }
56
57    return;
58  }
59
60  void make_tokenization(const fs::path& source, const fs::path& destination)
61  {
62    unsigned long long filename_count = 0;
63    for (auto& file : boost::make_iterator_range(fs::directory_iterator(source), {}))
64    {
65      tokenize_text(file, destination / (std::to_string(filename_count) + ".txt"));
66      ++filename_count;
67    }
68
69    return;
70  }
71
72  int main()
73  {
74    fs::path work_folder = LR"(C:\Users\Alexey\Desktop\IS\2020-03-13)";
75    fs::path parsed_data_folder_path = work_folder / parsed_data_folder;
76    fs::path tokenized_folder_path = work_folder / tokenized_data_folder;
77
78    utils::recreate_dir_safely(tokenized_folder_path);
79
80    make_tokenization(parsed_data_folder_path, tokenized_folder_path);
81
82    return 0;
83  }
```

3 Выводы

Выполнив первую лабораторную работу по курсу «Обработка текстов на естественном языке», я познакомился с проблемами и задачами подготовки обработанного текста для индексации. Я изучил и реализовал простейший токенизатор и словарь стоп-слов, который показал себя достаточно эффективным для постройки индекса.

Список литературы

[1] Маннинг, Рагхаван, Шютце *Введение в информационный поиск* — Издательский дом «Вильямс», 2011. Перевод с английского: доктор физ.-мат. наук Д. А. Клюшина — 528 с. (ISBN 978-5-8459-1623-4 (рус.))

[2] Список использованных источников оформлять нужно по ГОСТ Р 7.05-2008