Московский авиационный институт

(национальный исследовательский университет)

Институт информационные технологии и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу

«Обработка текстов на естественном языке»

Студент: Е.М. Стифеев
Преподаватель: А.А. Кухтичев
Группа: М8О-109М-21
Дата: 21.10.21
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

Разбитие документов на токены было решено производить следующим образом:

- Задать путь до директории с документами и начать рекурсивно обходить файлы.
- Открыть очередной текстовый файл, принадлежащий корпусу, в текстовом режиме на чтение. Напомню, что один документ хранится в *jsonlines*-файле (кодировка *UTF-8*) следующим образом:
  - 1 строка 1 документ {….}
  - 2 строка 2 документ {….}
  - *n* строка *n* документ {….}

  Дерево корпуса:

  - Корпус документов
    - films1.txt (94 Мб)
    - films2.txt (96 Мб)
    - films3.txt (184 Мб)
    - films4.txt (219 Мб)
    - films5.txt (322 Мб)
    - films6.txt (711 Мб)
    - films7.txt (823 Мб)
    - films8.txt (226 Мб)
    - films9.txt (67 Мб)
    - films10.txt (75 Мб)
    - films11.txt (99 Мб)
    - films12.txt (78 Мб)
    - films13.txt (41 Мб)

  В каждом файле films*.txt (кроме последнего) содержится по 15000 документов.

- Получить из очередной строки строку с токенами по правилам:
  - Удалить все пробельные символы;
  - Оставить слова и цифры;
  - Оставить дефисы, в случае конструкции вида *-*, где * – буква или цифра.

- Записать строку с токенами, разделёнными пробелами, в файл с названием doc_tokens.txt, где doc – оригинальное название файла (все файлы имеют одинаковое название), в текстовом режиме (для наглядности). Также в ту же строку записано количество содержащихся в ней токенов и её длина в символах.
- В итоге получим следующее дерево с токенами:
    - Токены
        - films1_tokens.txt
        - films2_tokens.txt
        - films3_tokens.txt
        - films4_tokens.txt
        - films5_tokens.txt
        - films6_tokens.txt
        - films7_tokens.txt
        - films8_tokens.txt
        - films9_tokens.txt
        - films10_tokens.txt
        - films11_tokens.txt
        - films12_tokens.txt
        - films13_tokens.txt

2. Исходный код

Инструментарий

На ОС Windows 10 для работы с кодировкой UTF-8 и файловой системой предусмотрены такие инструменты, как:

Инструмент	Назначение
`wchar_t`	Тип данных для работы с *декодированным* символом UTF-8
`wstring`	Класс для хранения строк из `wchar_t`
`filesystem`	Пространство имён с функциями для работы с файловой системой
`path`	Класс для работы с путями из `filesystem`
`_wfopen`	Открытие файлов на чтение/запись в кодировке UTF-8
`fgetws`	Чтение декодированной последовательности символов UTF-8 из файла в виде строки `wchar_t`
`fwprintf, fputws, fputwc, …`	Расширение стандартных функций

Исходный код доступен в проекте VS 2019 и состоит из одного файла main.cpp

Структура main.cpp

Сигнатура	Назначение
`#define ERROR_HANDLE(call, message, ...)`	Враппер для экстренного закрытия программы с очисткой памяти после возможно некорретного вызова call
`#define WARNING_HANDLE(call, message, ...)`	Враппер для пропуска определенных инструкций после возможно некоррекного вызова call
`#define INFO_HANDLE(message, ...)`	Враппер для логинга в процессе выполнения
`#define BUF_SIZE 50000`	Начальный размер буфера для чтения одного документа (предполагается, что один документ может не поместиться в него,

	поэтому предусмотрен механизм реаллокации)
`#define OMP_NUM_THREADS 4`	Количество потоков
`set<std::wstring> EXTENSIONS = { L".json", L".jsonlines", L".txt", L".xml" };`	Множество расширений файлов, подлежащих токенизации (программа будет корректно работать с любыми текстовыми файлами из этого списка)
`int get_tokens(const wchar_t *str, wchar_t *tokens, int *tokens_size)`	Преобразование строки слов `str` в строку токенов `tokens`, `tokens_size` – размер получившийся строки `tokens` в символах. Функция возвращает количество токенов.
`int wmain(int argc, wchar_t *argv[])`	Главная точка входа в программу

Замечу, что я распараллелил работу с файлами через библиотеку OpenMP для ускорения обработки.

Запуск

Исполняемый файл, скомпилированный под ОС Windows 10 лежит в папке \ЛР1\Токенизация\Release\Токенизация.exe.

Запуск:

```
$ ./Токенизация.exe –i path2corpusdir –o path2tokensdir
```

В моём случае программа отработала следующим образом:

```
$ ./Токенизация.exe -i "D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус" -o "D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Токены"

[INFO] Thread 0 processing 1/13 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус\films13.txt

[INFO] Thread 1 processing 2/13 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус\films9.txt

[INFO] Thread 3 processing 4/13 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус\films12.txt
```

[INFO] Thread 2 processing 3/13 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус\films10.txt

[INFO] Reallocate memory for buffer in thread 0

[INFO] Reallocate memory for buffer in thread 3

[INFO] Reallocate memory for buffer in thread 1

[INFO] Thread 3 processing 8/13 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус\films3.txt

[INFO] Reallocate memory for buffer in thread 1

[INFO] Reallocate memory for buffer in thread 2

[INFO] Reallocate memory for buffer in thread 2

[INFO] Thread 2 processing 7/13 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус\films11.txt

[INFO] Thread 1 processing 6/13 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус\films2.txt

[INFO] Thread 2 processing 11/13 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус\films5.txt

[INFO] Reallocate memory for buffer in thread 2

[INFO] Reallocate memory for buffer in thread 2

[INFO] Reallocate memory for buffer in thread 2

[INFO] Reallocate memory for buffer in thread 2

[INFO] Thread 0 processing 5/13 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус\films1.txt

[INFO] Thread 3 processing 12/13 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус\films6.txt

[INFO] Thread 1 processing 10/13 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус\films8.txt

[INFO] Reallocate memory for buffer in thread 1

[INFO] Reallocate memory for buffer in thread 3

[INFO] Reallocate memory for buffer in thread 3

[INFO] Reallocate memory for buffer in thread 1

```
[INFO] Reallocate memory for buffer in thread 3

[INFO] Reallocate memory for buffer in thread 3

[INFO] Thread 0 processing 9/13 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус\films4.txt

[INFO] Reallocate memory for buffer in thread 3

[INFO] Thread 0 processing 13/13 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус\films7.txt

[INFO] Reallocate memory for buffer in thread 0

[INFO] Reallocate memory for buffer in thread 0

[INFO] Reallocate memory for buffer in thread 0

[INFO] Reallocate memory for buffer in thread 0

[INFO] Reallocate memory for buffer in thread 0

[INFO] Reallocate memory for buffer in thread 3

[INFO] Reallocate memory for buffer in thread 0

[INFO] Total tokens = 259167384, avg_token = 5,43

Total time = 51,3 sec, total size = 2,899 Gb

Speed = 17,268 ms / Kb
```

3. Выводы

После завершения обработки получились следующие цифры

Общее количество токенов	259’167’384
Средняя длина токена в символах	5,43
Общее время выполнения	51,3 sec
Общий объём обработанных файлов	2,899 Gb
Среднее время обработки КБайта исходного текста документа	17,268 ms / Kb

Ссылка на корпус токенов: https://cloud.mail.ru/public/EjXn/3BkexibzN.

Зависимость времени выполнения от объёмы входных данных является линейной по общему количеству символов во всех документах. Скорость выполнения по асимптотике является оптимальной, т.к. является минимальной для обработки всех символов в тексте. Ускорение можно получить, если поменять жёсткий диск на SSD, т.к. всё упирается именно в скорость чтения/записи.

Примеры неудачно вычисленных токенов:

Токен	Причина неудачного выбора
115	Ничего не значащее число, однако пользователь может попробовать искать значащее число
с	Предлог, возможно, не несущий важной информации
что	Союз
её	Притяжательное местоимение
зоаноидов	Возможно, слово, написанное с ошибкой

Возможные улучшения: машиннообучаемая или иного рода система по распознаванию ошибок в тексте, притяжательных местоимений и прочего.

В ходе выполнения лабораторной работы я научился обрабатывать текстовые файлы в UTF-8 кодировке, разбивать текст на токены с помощью C++.