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
    - filmplace
      - animation.jsonlines (22 Мб)
      - documentary.jsonlines (13 Мб)
      - film.jsonlines (295 Мб)
      - music.jsonlines (3 Мб)
      - tv.jsonlines (41 Мб)
    - kinopoisk
      - content.jsonlines (188 Мб)
      - films1.jsonlines (1087 Мб)
      - films2.jsonlines (790 Мб)
    - shikimori
      - comments.jsonlines (419 Мб)
      - critiques.jsonlines (26 Мб)
      - summaries.jsonlines (150 Мб)
- 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
    - animation_tokens.txt
    - comments_tokens.txt
    - content_tokens.txt
    - critiques_tokens.txt
    - documentary_tokens.txt
    - film_tokens.txt
    - films1_tokens.txt
    - films2_tokens.txt
    - music_tokens.txt
    - summaries_tokens.txt
    - tv_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)
`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)
`bool is_letter(wchar_t c)`	Проверка, является ли символ буквой
`bool is_number(wchar_t c)`	Проверка, является ли символ цифрой
`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

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 документов" -o "D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Токены"

[INFO] Processing 1 / 11 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус документов\filmplace\animation.jsonlines

[INFO] Processing 2 / 11 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус документов\filmplace\documentary.jsonlines

[INFO] Reallocate memory for buffer
```

```
[INFO] Processing 3 / 11 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус документов\filmplace\film.jsonlines

[INFO] Reallocate memory for buffer

[INFO] Processing 4 / 11 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус документов\filmplace\music.jsonlines

[INFO] Processing 5 / 11 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус документов\filmplace\tv.jsonlines

[INFO] Processing 6 / 11 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус документов\kinopoisk\content.jsonlines

[INFO] Processing 7 / 11 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус документов\kinopoisk\films1.jsonlines

[INFO] Reallocate memory for buffer

[INFO] Reallocate memory for buffer

[INFO] Reallocate memory for buffer

[INFO] Reallocate memory for buffer

[INFO] Processing 8 / 11 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус документов\kinopoisk\films2.jsonlines

[INFO] Reallocate memory for buffer

[INFO] Processing 9 / 11 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус документов\shikimori\comments.jsonlines

[INFO] Processing 10 / 11 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус документов\shikimori\critiques.jsonlines

[INFO] Processing 11 / 11 : D:/Мои документы/Лабы и рефераты/5 курс 1 семестр/Обработка текстов на естественном языке/Корпус документов\shikimori\summaries.jsonlines

[INFO] Total tokens = 258783688, avg_token = 5,43
```

Total time = 196,8 sec, total size = 3,264 Gb

Speed = 58,893 ms / Kb

3. Выводы

После завершения обработки получились следующие цифры

Общее количество токенов	258’783’688
Средняя длина токена в символах	5,43
Общее время выполнения	196,8 sec
Общий объём обработанных файлов	3,264 Gb
Среднее время обработки КБайта исходного текста документа	58,893 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 слов в тексте. «Неасимптотическое» ускорение можно получить распараллелив программу, например с использованием *OpenMP*, т.к. обработка каждой строки является независимой операцией (автор не стал этого делать, чтобы не «смазывать» время выполнения программы).

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 и разбивать текст на токены.