ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
NHẬP MÔN CÔNG NGHỆ THÔNG TIN

----------o0o----------

BÁO CÁO ĐỒ ÁN MÔN HỌC

SEARCH ENGINE

GVHD: **VÕ HOÀI VIỆT**
NHÓM: 02
SVTH: Vương Gia Huy 20120014
Huỳnh Thiết Gia 20120070
LỚP: 20CTT1TN

Tp. Hồ Chí Minh, tháng 7 năm 2021

Mục lục

1. GIỚI THIỆU

1.1. Nội dung đồ án

Rút trích đơn giản nội dung chính của văn bản tiếng Việt. Tìm kiếm những văn bản có nội dung tương ứng với từ khoá do người dùng nhập vào, xếp hạng kết quả tìm kiếm theo mức độ liên quan đến từ khoá từ cao đến thấp

1.2. Yêu cầu đồ án

Các sinh viên trong lớp cùng tập hợp dữ liệu thống nhất bao gồm các văn bản tiếng Việt. Các tập tin văn bản được đặt trong cùng một thư mục với tên tập tin tương ứng với tựa đề và phần mở rộng .txt. Dựa trên các văn bản nguồn, sinh viên tự tạo ra tập tin siêu dữ liệu (metadata) ở dạng nhị phân hoặc văn bản: thông tin về các văn bản đã tiền xử lý và nội dung chính của từng văn bản theo cấu trúc tự định nghĩa. Khi thêm/xóa tập tin văn bản, chương trình tự động cập nhật

dữ liệu của tập tin siêu dữ liệu.

1.3. Thông tin nhóm

Họ và tên	MSSV	Công việc
Vương Gia Huy	**20120014**	**-Thiết kế thuật toán xây dựng meta data** **-Viết báo cáo**

Huỳnh Thiết Gia	**20120070**	**-Thiết kế thuật toán tìm kiếm** **-Tổ chức xây dựng** **-Kiểm thử**

2. XÂY DỰNG TẬP TIN SIÊU DỮ LIỆU (META DATA)

2.1. Truy xuất văn bản

Vấn đề kiểu dữ liệu lưu trữ

Kiểu string của C++ mặc định sử dụng kiểu ký tự (character type) là char, chỉ chứa được 8 bits tương đương 256 ký tự cho bộ mã ASCII, để lưu được tiếng việt phải chuyển qua kiểu wide string dùng kiểu ký tự wchar.

Vấn đề mã hóa văn bản

File txt được lưu với những kiểu mã hóa khác nhau, phổ biến nhất là ASCII, UTF8, UTF8-BOM (chứa 3 ký tự 0xEF, 0xBB, 0xBF ở đầu), UCS2-LE BOM, UCS2-BE BOM. Bộ dữ liệu thầy đưa từ năm 2006, xài kiểu mã hóa UCS2-LE BOM khác kiểu phổ biến nhất internet là UTF8, nên phải thiết kế hàm đọc văn bản cho các trường hợp trên.

Đọc văn bản

Đọc văn bản không có gì khác biệt so với đọc bình thường, dùng ifstream đọc từ file truyền vào stringstream, sử dụng codecvt_utf8_utf16 để chuyển đổi các kiểu mã hóa sang string UTF8 rồi chuyển hóa hết sang UTF16Le.

Dọn dẹp văn bản

Văn bản đọc xong phải xóa hết khoảng trắng thừa và các ký tự đặc biệt (“!@#$….”), duyệt từ đầu tới cuối kiểm tra từng ký tự trong chuỗi (và ký tự trước đó để phát hiện hai khoảng trắng liên tiếp) . Dùng hashtable tự định nghĩa để lưu mảng ký tự giúp truy xuất nhanh.

2.2. Loại bỏ stop words

Lưu danh sách stopwords vào hashtable để truy xuất nhanh, sử dụng wistringstream để nhanh chóng kiểm tra chuỗi theo từng từ, lưu lại 3 từ trước đó để kiểm tra stop words dài ("biết chừng nào").

2.3. Tạo danh sách tokens

Sử dụng mô hình Bag-of-words để lưu lại danh sách tất cả các từ khóa trong văn bản đã loại bỏ stop words, chỉ lấy những từ khóa xuất hiện hơn 2 lần (Giảm thời gian tìm kiếm nhưng vẫn giữ được chất lượng, đã thử nghiệm với bộ dữ liệu) do những từ khóa chính trong văn bản thường xuất hiện nhiều lần.

2.4. Tổ chức dữ liệu metadata

Lưu các tokens ở dạng pair <wstring, int> với ý nghĩa: <từ khóa, số lần xuất hiện> vào mảng, sau đó lưu tiếp mảng đó ở dạng pair <wstring, pair <wstring, int>> với ý nghĩa <tên file, danh sách token>, sắp xếp lại mảng theo thứ tự tên file để truy xuất nhanh.

2.5. Đọc ghi tập tin siêu dữ liệu (meta data)

Sử dụng wofstream để lưu file với định dạng "[tên file],0\n" + "[từ khóa đầu],[số lần xuất hiện]\n" + "[từ khóa hai],[số lần xuất hiện]\n" + …

Khi đọc file chỉ cần đọc theo dòng rồi chia theo dấu phẩy, nếu phía sau dấu phải là số 0 thì ta biết là đã qua tên file và danh sách token khác.

3. TÌM KIẾM VĂN BẢN

3.1. Tải tập tin siêu dữ liệu

Khi mới khởi động chương trình ta đọc tập tin ngay từ đầu, với cách đọc như trên.

3.2. Nhận và xử lý từ khoá

Đọc từ khóa từ bàn phím người dùng, tách từ khóa như cách ta tách stop words: lưu lại 3 từ trước đó để tìm kiếm từ khóa dài.

3.3. Thuật toán tìm kiếm

Duyệt hết tất cả tokens, tìm kiếm các từ khóa bên trong và tính tổng số lần xuất hiện nếu phát hiện trùng, theo tỉ lệ độ dài từ khóa (từ khóa trùng càng dài thì càng được tôn trọng).

3.4. Giao diện

Không biết dùng Qt nên nhóm sử dụng giao diện nhập bằng bàn phím.

4. NHẬN XÉT

4.1. Thời gian đọc file

Khoảng 14 phút để đọc hết thư mục new train.

4.2. Thời gian trả kết quả

Rất nhanh cho tập dữ liệu train, dưới 3 giây, hoàn toàn phù hợp cho người dùng bình thường.

4.3. Sử dụng bộ nhớ

Chương trình dùng 78MB lúc khởi tạo, đọc 1000 files dùng thêm 2MB.

5. ĐÁNH GIÁ THỰC HIỆN

5.1. Mức độ hoàn thành đồ án

Bảng 2. Đánh giá mức độ hoàn thành đồ án

Yêu cầu	Mức độ hoàn thành
Các chức năng cơ bản	100%
Tối ưu đọc file	80%
Tính năng tìm kiếm	100%
Lưu, mở metadata	100%
TỔNG	**90%**

5.2. Đánh giá thành viên

Bảng 3. Đánh giá thành viên trong nhóm

Họ và tên	Khối lượng công việc	Mức độ hoàn thành
Vương Gia Huy	49.99%	100%
Huỳnh Thiết Gia	50.01%	100%

---[HẾT]---